	along which communications traffic should be forwarded and that is not a firewall	

Router is a well-understood word.[53] There is nothing ambiguous about it and there is no evidence that the patent applicant clearly redefined (in the application or prosecution history) "router," an unambiguous term, it should simply be construed as "plain and ordinary meaning."[54]

7. host computer[55]

Plaintiff's proposed construction	**Defendant's proposed construction**	**By:**
Plain and ordinary meaning	a computer that processes received and transmitted data throughthe application layer and that is not a firewall	D

A Host Computer is a well-understood term.[56] There is nothing ambiguous about it and there is no evidence that the patent applicant clearly redefined (in the application or prosecution history) "host computer," an unambiguous term, it should simply be construed as "plain and ordinary meaning."[57]

8. means for limiting the transmission rate from the firewall to the maximum acceptable transmission rate for each class of data packet[58]

Plaintiff's proposed construction	**Defendant's proposed construction**	**By:**
Function: limiting the transmission rate from the firewall to the maximum acceptable transmission rate for each class of data packet Structure: 3:18 to 3:20; 3:42 to 3:49; 4:46 to 4:50; 5:10 to 5:18; Figure 1 provides details	Indefinite	J

[53] *See* Narayanaswamy Decl. at ¶74.
[54] *See Arthrex*, 2016 U.S. Dist. LEXIS 105750, at **106, 108, 127 173, 174 (Payne, Mag.).
[55] Claim 1 of the '190 patent; Claim 1 of the '564 patent; Claims 1, 4, 7, 10, 13, and 16 of the '497 patent.
[56] *See* Narayanaswamy Decl. at ¶76.
[57] *See Arthrex*, 2016 U.S. Dist. LEXIS 105750, at **106, 108, 127 173, 174 (Payne, Mag.).
[58] Claim 1 of the '564 patent.

The '564 patent discloses and claims a reverse firewall packet control system for controlling packet transmission between two networks.[59] From the perspective of one of ordinary skill in the art, the preamble of claim 1 of the '564 patent, [60] followed by the subsequent claim language provide the structure of a firewall comprising hardware and software for each of "means for classifying...", means for associating..." and "means for limiting the transmission rate..." A firewall has a very well-defined meaning in the art and is a device which controls the transmission of data between one or more networks. The data packets are forwarded along communication lines or locations between appropriate parts of the one or more networks. It is very common for a firewall to be between two networks, such as an internal network and an external network.[61]

One of ordinary skill in the art would understand that it is a firewall between two networks that is:

1. classifying data packets received at the firewall;

2. associating a maximum acceptable transmission rate with each class of data packet received at said firewall; and,

3. limiting the transmission rate from the firewall to the maximum acceptable transmission rate for each class of data packet.[62]

The literal language of the claim and the specification support this construction. First, the claim sets forth a firewall as the first element.[63] Second, the "at least one firewall" comprises "hardware and software ... serving to control packet transmission"[64] which one of ordinary skill in the art would know is a firewall with software, as a firewall controls packet transmission. Thus, sufficient structure is provided in the claim that each of "means for classifying...", "means for

[59] *See* Doc. No. 1-2 at 3:31-42; Narayanaswamy Decl. at ¶26.
[60] *See* Doc. No. 1-2 at 6:26-47.
[61] *See* Narayanaswamy Decl. at ¶26.
[62] *See* id.
[63] *See* Doc. No. 1-2 at 6:31-34.
[64] *See* id.

associating..." and "means for limiting..." are not properly considered as means plus function claim terms.

As an alternate position, adequate disclosure of the structure of a firewall is disclosed in the '564 patent's specification. The software on the firewall or router for performing the functions is explained through the algorithms disclosed in Figure 1 which illustrates a packet transmission control system 10 for managing traffic 14 between at least two data networks 18, 22. The firewall 42 includes hardware and software providing a non-redundant connection 46 between the networks 18, 22 and serves to control packet transmission between the networks 18, 22. The algorithm discloses maximum transmission rates (one of 12, 9, 20, or 4) and an associated class of data packet 66.[65]

Figure 1 discloses an algorithm and diagram by flowchart for classifying data packets 38 received at the firewall 42 related to the consumption of at least one resource. The flowchart further shows associating a maximum acceptable transmission rate 62 with each class 66 of data packet 38 received at the firewall 42. The flowchart further shows the maximum transmission rate can be set as a limit on the transmission rate of packets across the firewall. When transmission rates 62 from the firewall 42 are so limited, packet flooding and other over usage type distributed denial of service attacks cannot be effectively launched through the network connection.[66] Moreover, a firewall inherently possesses the capability to inspect and label or mark (i.e., classify) data packets based on their properties.[67] This would include available location information that accurately and reliably provides the firewall with the locations through which attackers are sending flood attacks[68] as would be understood by one of ordinary skill in the art and as is disclosed in the highly related '497 patent.[69] The patents cited in the prior art of this specification, US6154775 and US6304975, both show that firewalls are well-known in the prior art to have the

[65] *See* Doc. No. 1-2 at Figure 1.
[66] *See* Doc. No. 1-2 at 5:10-31.
[67] *See* Narayanaswamy Decl. at ¶s 28-29.
[68] A flood attach is commonly known as Distributed Denial of Service ("DDoS").
[69] *See* Doc. No. 1-3 at 3:64-66 ("which is done via packet marks provided by router....).

ability to inspect and classify packets by their properties. In this invention, such classification would also include inspecting the data packets to discern the location information added to those packets by cooperating routers as disclosed in the highly relevant '497 patent. Using data packets that incorporate marks encoding the locations taken by the packet through a cooperating network of routers was not known in the art.[70]

Lastly, it is common parlance for one of ordinary skill in the art to use a firewall for limiting the transmission rate from the firewall to the maximum acceptable transmission rate for a class of data packet. The patents cited in the prior art of this specification, US6154775 and US6304975, both show that firewalls are well-known in the prior art to have the ability to inspect and classify packets by their properties. In this invention, such classification would also include inspecting the data packets to discern the path information added to those packets by cooperating routers as disclosed in the highly relevant '497 patent. Using data packets that incorporate marks encoding the paths taken by the packet through a cooperating network of routers was not known in the art.[71]

9. means for said firewall to measure the amount of service requested by each packet[72]

Plaintiff's proposed construction	**Defendant's proposed construction**	**By:**
Function: measuring the amount of service requested by each packet Structure: 3:20 - 3:24; 4:17 - 4:23; 4:61 - 4:64; 5:62 - 5:64; Figure 4	Indefinite	J

The flow chart and algorithm for this claim term is disclosed in Figure 4. A firewall 42 further identifies data packets 38 as requests for services 98 of at least one type requiring transmission of data packets 38 from locations 78 within one of the networks 18 to another of the networks 22. The firewall 42 measures the amount of service 36 requested by each identified

[70] *See* Narayanaswamy Decl. at ¶29.
[71] *See* id. at ¶31.
[72] Claim 4 of the '564 patent.

packet 38. Based upon this identification and this measure of amount of service 36, firewall 42 will thus limit the transmission rate for data packets 38 that are requests for services 98 based upon the type of service 98 requested in order to limit usage of each Service 98.[73] Two exemplary services disclosed are "requests for file transfer to B" and "requests for web page retrieval from C."[74] The disclosure of this algorithm is sufficient for structure for one of ordinary skill in the art to practice the invention.[75]

10. whereby, packet flooding and other over usage type distributed denial of service attacks cannot be effectively launched through said non-redundant connection[76]

Plaintiff's proposed construction	Defendant's proposed construction	By:
No construction necessary	Indefinite	D

Construction of long claim terms is disfavored.[77] Additionally, this "whereby" clause does not provide a limitation on the claim and need not be construed.[78] If a whereby clause adds nothing to the claims and simply characterizes the intended result of the executing steps, it need not be construed by the court.[79] Here, no meaningful limitation is added by this "whereby" clause and no construction is necessary.

11. whereby, said firewall will limit the transmission rate for data packets of each class from locations within one of said networks to provide proportionally fair forwarding service to other locations within said network that communicates

[73] *See* Doc. No. 1-2 at 5:52-64.
[74] *See* Doc. No. 1-2 at Figure 4 (also shown in Figures 5 and 6).
[75] *See* Narayanaswamy Decl. at ¶33.
[76] Claim 1 of the '564 patent
[77] *Ibuace, Inc. v. Eloqua Ltd.*, Civ. Act. No. 2:11cv2572012, U.S. Dist. LEXIS 62514, at *15, *16 (E.D. Va. May 2, 2012) (describing a 42-word term as "lengthy" and declining to construe the term); *Honeywell Int'l, Inc. v. Nikon Corp.*, 589 F. Supp. 2d 433, 440–444 (D. Del. 2008) (describing a 24-word term as "rather lengthy," then focusing on 2 words of the claim term, and ultimately declining to construe the 24-word term).
[78] *Tex. Instruments Inc. v. U.S. Int'l Trade Comm'n*, 988 F.2d 1165, 1172 (Fed.Cir.1993) (concluding, in process claims, that the whereby clauses described "the result of arranging the components of the claims in the manner recited in the claims" and thus were not limitations).
[79] *See Minton*, 336 F.3d at 1381.

through said non-redundant connection[80]

Plaintiff's proposed construction	Defendant's proposed construction	By:
No construction necessary	Indefinite	D

Construction of long claim terms is disfavored.[81] Additionally, this "whereby" clause does not provide a limitation on the claim and need not be construed.[82] "If a whereby clause adds nothing to the claims and simply characterizes the intended result of the executing steps, it need not be construed by the court.[83] Here, no meaningful limitation is added by this "whereby" clause and no construction is necessary.

12. means for said firewall to measure the amount of service consumed in order to send each identified response data packet[84]

Plaintiff's proposed construction	Defendant's proposed construction	By:
Function: measuring, for the firewall, the amount of service of each type consumed to send each identified response data packet Structure: 3:24 - 3:27; 4:23 - 4:31; 4:65 - 5:03; 5:65 - 6:13; Figure 5	Indefinite	J

Figure 5 discloses flowchart and algorithm for a firewall for classifying data packets 38 received at the firewall 42 wherein the firewall further includes identifying data packets 38 as requests for services 98 of at least one type requiring transmission of data packets 38 from locations

[80] Claim 2 of the '564 patent

[81] *Ihance, Inc. v. Eloqua Ltd.*, Civ. Act. No. 2:11cv2572012, U.S. Dist. LEXIS 62514, at *15, *16 (E.D. Va. May 2, 2012) (describing a 42-word term as "lengthy" and declining to construe the term); *Honeywell Int'l, Inc. v. Nikon Corp.*, 589 F. Supp. 2d 433, 440–444 (D. Del. 2008) (describing a 24-word term as "rather lengthy," then focusing on 2 words of the claim term, and ultimately declining to construe the 24-word term).

[82] *Tex. Instruments Inc. v. U.S. Int'l Trade Comm'n*, 988 F.2d 1165, 1172 (Fed.Cir.1993) (concluding, in process claims, that the whereby clauses described "the result of arranging the components of the claims in the manner recited in the claims" and thus were not limitations).

[83] *See Minton*, 336 F.3d at 1381.

[84] Claim 5 of the '564 patent

78 within one of the networks 18 to another of the networks 22 and identifying data packets 38 as responses to earlier service requests 98 of at least one type from a location 78 within one of the networks 18 requiring transmission of data packets 38 to another of the networks 22. Firewall 42 measures the amount of service 36 requested by each identified data packet 38. Based upon this identification and this measure of amount of service 36, the firewall 42 will thus limit the transmission rate for data packets 38 that are requests for services 98 based upon the type of service 98 requested in order to limit usage of each service 98.[85]

13. means for storing and recalling past measurements of amounts of service provided for each type of service[86]

Plaintiff's proposed construction	**Defendant's proposed construction**	**By:**
Function: storing and recalling past measurements of amounts of service provided for each type of service Structure: 3:24 - 3:29; 4:32 - 4:39; 5:03 - 5:05; 6:14 - 6:20; Figure 6	Indefinite	J

This is not a term subject to a 112(f) analysis. The claim term is for computer memory for storing and recalling data.[87] Computer memory is a well-known example of a term not subject to 112(f).[88] As an alternate position, Plaintiff identifies structure from the patent.

14. whereby, said firewall will limit the transmission rate for data packets that are requests for each type of service to limit usage of each service over extended periods of time [89]

Plaintiff's proposed construction	**Defendant's proposed construction**	**By:**

[85] *See* Doc. No. 1-2 at 5:65-6:13; *See* Narayanaswamy Decl. at ¶35.
[86] Claim 6 of the '564 patent
[87] *See* Narayanaswamy Decl. at ¶37.
[88] *TecSec, Inc.* 731 F.3d at 1437 (A "system memory" is sufficient structure to perform the "storing data" function. To those skilled in the art, a system memory is a specific structure that stores data.).
[89] Claim 6 of the '564 patent

No construction necessary	Indefinite	D

Construction of long claim terms is disfavored.[90] Additionally, this "whereby" clause does not provide a limitation on the claim and need not be construed.[91] If a whereby clause adds nothing to the claims and simply characterizes the intended result of the executing steps, it need not be construed by the court.[92] Here, no meaningful limitation is added by this "whereby" clause and no construction is necessary.

15. means for classifying data packets received at a/said host computer/router into wanted data packets and unwanted data packets/...by path[93]

Plaintiff's proposed construction	Defendant's proposed construction	By:
Function: classifying data packets received at a host computer into wanted data packets and unwanted data packets/...by path Structure: 3:30 - 3:42; 5:29 - 5:33; 6:12 - 6:25; Figure 1; Figure 4; Figure 7	Indefinite	J

16. means for associating a maximum acceptable processing rate with each class of data packet received at said computer[94]

Plaintiff's proposed construction	Defendant's proposed construction	By:
Function: associating a maximum acceptable processing rate with each class of data packet received at said computer Structure: 3:43 - 3:61; 5:33 - 5:39; 6:26 - 6:46; Figure 2; Figure 5; Figure 8	Indefinite	J

[90] *Ihance, Inc. v. Eloqua Ltd.*, Civ. Act. No. 2:11cv2572012, U.S. Dist. LEXIS 62514, at *15, *16 (E.D. Va. May 2, 2012) (describing a 42-word term as "lengthy" and declining to construe the term); *Honeywell Int'l, Inc. v. Nikon Corp.*, 589 F. Supp. 2d 433, 440–444 (D. Del. 2008) (describing a 24-word term as "rather lengthy," then focusing on 2 words of the claim term, and ultimately declining to construe the 24-word term).

[91] *Tex. Instruments Inc. v. U.S. Int'l Trade Comm'n*, 988 F.2d 1165, 1172 (Fed.Cir.1993) (concluding, in process claims, that the whereby clauses described "the result of arranging the components of the claims in the manner recited in the claims" and thus were not limitations).

[92] *See Minton*, 336 F.3d at 1381.

[93] Claims 1, 4, 13 and 16 of the '497 patent.

[94] Claim 1 of the '497 patent.

Plaintiff groups these two terms together as the disclosure of structure and argument is largely coextensive. However, while Plaintiff contends these terms do not require the 112(f) analysis, Plaintiff's identified function and structure are provided in the boxed chart with argument and explanation following.

These claim terms are not subject to the requirements of 112(f) because from the perspective of one of ordinary skill in the art, the preamble of claim 1 of the '497 patent,[95] followed by the subsequent claim language provide the structure of a router or firewall for each of "means for classifying..." and "means for associating..."[96] Both a router and a firewall have a very well-defined meaning in the art. A firewall is a device which is placed between two networks by the owner of one of those networks to safeguard that network from various network attacks. These capabilities include identifying sets of data packets by certain properties (referred to as classification) and allocating a processing rate to those packets identified by class. In the current invention, this method is extended to include information about the path taken by the packets as they traverse the computer network before they arrive at the firewall. A router is a device which forwards data packets to locations within a computer network. The data packets are forwarded along communication lines or paths between the appropriate parts of the computer network. It is very common for a router and/or a firewall to be part of securing a computer network for inspecting the contents of packets, and/or limiting the rates of transmission of specific sets of packets identified by properties (i.e., classification) including blocking that set of packets entirely.[97]

One of ordinary skill in the art would understand that a router or a firewall provides the following capabilities:

---

[95] *See* Doc. No. 1-3 at 8:9-27.
[96] *See* Narayanaswamy Decl. at ¶40.
[97] *See* id. at ¶40.

1. classifying data packets received at a host computer and

2. associating a maximum acceptable processing rate...

based on the literal language of the claim and the specification.[98] First, the claim sets forth in the preamble that a packet flooding defense system of the invention includes a router.[99] Second, in common parlance to one of ordinary skill in the art, an as provided in the specification, the "at least one firewall" comprises "hardware and software serving to control packet transmission"[100] which one of ordinary skill in the art would know is a firewall that can be configured by software to provide the capabilities described.[101] Thus, it is my opinion that sufficient structure is provided in the claim that each of "means for classifying..." and means for associating..." are not properly considered as means plus function claim terms as one of ordinary skill in the art would appreciate the structure disclosed is a router or a firewall. However, adequate disclosure of the structure of a router and a firewall is also disclosed in the specification.

The claimed invention describes how the forwarding path for each packet is associated with that packet, thereby making it possible for a receiving computer to know that information. The software, or configuration, at the firewall is explained through the algorithms disclosed in Figure 1 by flowchart and diagram where a Class A and Class B (38) of a maximum acceptable processing rate (34) are assigned by the firewall to each data packet (30). The description provides that the computer can use the information, the maximum acceptable processing rate, to allocate the processing rate for each class (38) in a desired way among the places from which packets (30)

[98] *See* Narayanaswamy Decl. at ¶s40-41.
[99] *See* Doc. No. 1-3 at 8:9-11.
[100] *See* Doc. No. 1-3 at 4:47-60.
[101] *See* Narayanaswamy Decl. at ¶s40-41.

are transmitted.[102] Figure 2, Figure 3, Figure 4, Figure 5, and Figure 6 illustrate additional embodiments, or algorithms, in which path information associated with each packet can be combined with other properties to classify the packet and assign it a transmission rate.[103] Figure 7, Figure 8 and Figure 9 illustrate how a firewall protecting a network communicates the rate limit for packets routed to it or routed to it by class functions.[104] Moreover, the marking of data packets by a firewall or router is within the ordinary parlance of one of ordinary skill in the art and as is disclosed in the '497 patent.[105] Prior art discussed in the specification of the '497 patent (US5455865 and US6044402 for example) rely on the well-understood and accepted functionality of routers to maintain flow information between source and destination addresses.[106] Marking of data packets that traverse a router is well-known in the art. Marking of data packets to encode path information is unique to this invention, and anyone familiar with firewalls would understand how to use a firewall capability to encode paths. The specification describes how a firewall and a router can add to packet information about the interface points where the packet entered the firewall or router. Anyone familiar with firewalls or routers would be able to implement this in different ways. Using data packet marks that encode the forwarding path of the path in a cooperating network of routers was not known in the prior art. The importance of the path information is that it is generated by routers in the network, and, therefore, not something that the attacker controls. The packet path information is very reliable and accurate in identifying the directions from which attackers are mounting attacks.[107]

---

[102] *See* Narayanaswamy Decl. at ¶42.
[103] *See* Doc. No. 1-3 at Figures 2, 3, 4, 5, and 6; *See* Narayanaswamy Decl. at ¶42.
[104] *See* Doc. No. 1-3 at Figures 7, 8, and 9; *See* Narayanaswamy Decl. at ¶42.
[105] *See* Doc. No. 1-3 at 3:64-66 ("which is done via packet marks provided by router....); *See* Narayanaswamy Decl. at ¶42.
[106] *See* Narayanaswamy Decl. at ¶41.
[107] *See* id. at ¶42.

The '497 patent described the shortcomings of the prior art at 2:23-29 wherein it provided an issue is the lack of a reliable way to associate incoming packets with those users among whom bandwidth should be fairly allocated. The other is lack of control over what packets arrive. The solution described here to both of these problems requires help from the routers that forward packets to the victim.

The '497 patent provides several technological solutions on overcoming these prior art issues. At 3:62-4:5, the '497 patent teaches:

> ... a path (which is not controlled by the attacker) is used to determine the actual direction of the packet flow towards the victim. Bandwidth is allocated based upon path (which is done via packet marks provided by routers leading up to the victim). In other words this invention uses attacker-independent information about the path a packet takes to allocate forwarding bandwidth in a router. The part that makes this invention completely different from Chang, et al, is that the information has to be attacker-independent (i.e., sender-independent) in order to work as a defense.

Further, at 8:18-24, the '497 patent teaches that a host computer is able to identify the path by the packet mark applied at a router:

> means for said computer to find information for packets it receives regarding the path by which said packets came to said computer via packet marks provided by routers leading to said host computer; said path comprising all routers in said network via which said packets are routed to said computer; and

Further, at 9:01-04, the '497 patent teaches a router is able to identify the path of a packet by packet mark provided by a router:

> means for said router to find information for packets it receives regarding the path by which said packets came to said router via packet marks provided by routers leading to said host computer;

Further, at 9:49-9:53, the '497 patent teaches a host computer is able to determine a path by which data packets arrive at a host computer by packet marks:

> determining a path by which data packets arrive at a host computer via packet marks provided by routers leading to said host computer; said path comprising all routers in said network via which said packets are routed to said computer;

Further, at 10:29-34, the '497 patent teaches a downstream router is able to determine a path by packet marks from an upstream router:

> determining a path by which data packets arrive at said router via packet marks provided by routers leading to said host computer; said path comprising all routers in said network via which said packets are routed to said computer;

Further, at 11:10-16, the '497 patent teaches a system of cooperating routers wherein a host computer is able to identify the path a packet comes in on by packet marks:

> means for determining a path by which data packets arrive at a host computer via packet marks provided by routers leading to said host computer; said path comprising all routers in said network via which said packets are routed to said computer;

Further, at 12:06-10, the '497 patent teaches a system of cooperating routers wherein a router is able to identify the path a packet comes in on by packet marks:

> means for a router to determine a path by which said packets came to said router via packet marks provided by routers leading to said router; said path comprising all routers in said network via which said packets are routed to said computer;

Figures 1, 2, 3, 4, 5, and 6 each illustrate illustrative methods of determining, by packet marking, how the path a packet takes is identified. These Figures, in cooperation with the written description, further provide various algorithms, by flowchart, used by the claimed invention to perform the claimed functions.[108]

Lastly, it is common parlance for one of ordinary skill in the art to use a router or a firewall to (1) mark data packets and for (2) associating a maximum acceptable processing rate with each

[108] *See* Narayanaswamy Decl. at ¶51.

class of data packet. Prior art cited in this patent specification including US5455865 and US 5353353 use similar capabilities to implement other network attack defenses. In this patent specification the same capabilities are utilized to limit the rate of packets with added path information.[109] The '497 patent's specification describes how processing rate limits can be used to allocate the processing rates available for packets of each class in a desired manner.[110]

17. means for said computer to find information for packets it receives regarding the path by which said packets came to said computer via packet marks provided by routers leading to said host computer[111]

Plaintiff's proposed construction	**Defendant's proposed construction**	**By:**
Function: finding information, for said computer, for packets received regarding the path by which said packets came to said computer via packet marks provided by routers leading to said host computer Structure: 3:62 - 4:20; 5:40 - 5:44; 6:46 - 6:51; Figure 3; Figure 6; Figure 9	Indefinite	J

18. means in said computer for using said information to allocate the processing rate available for unwanted data packets to be less than or equal to said maximum acceptable processing rate[112]

Plaintiff's proposed construction	**Defendant's proposed construction**	**By:**
Function: allocating the processing rate available for unwanted data packets to be less than or equal to said maximum acceptable processing rate Structure: 3:30 - 3:42; 5:29 - 5:33; 6:12 - 6:25; Figure 1; Figure 4; Figure 7	Indefinite	J

[109] *See* Narayanaswamy Decl. at ¶s 53, 55.
[110] *See* Doc. No. 1-3 at 3:30-42.
[111] Claim 1 of the '497 patent.
[112] Claim 1 of the '497 patent.

Plaintiff groups these two terms together as the disclosure of structure and argument is largely coextensive. Plaintiff's identified function and structure are provided in the boxed chart with argument and explanation following.

The means for the computer (1) to find information for packets it receives and (2) to use the information to allocate processing rate available for unwanted data packets to be less than or equal to said maximum acceptable processing rate, is disclosed in the algorithm shown in Figure 1. The software, or configuration, at the computer is explained through the algorithms disclosed in Figure 1 by flowchart and diagram where a Class A and Class B (38) of a maximum acceptable processing rate (34) are assigned by the firewall to each data packet (30). The description provides that the computer can use the information, the maximum acceptable processing rate, to allocate the processing rate for each class (38) in a desired way among the places from which packets (30) are transmitted.[113] Figure 2, Figure 3, Figure 4, Figure 5, and Figure 6 illustrate additional embodiments, or algorithms, in which path information associated with each packet can be combined with other properties to classify the packet and assign it a processing rate.[114] Figure 7, Figure 8 and Figure 9 illustrate how a firewall protecting a network communicates the rate limit for packets routed to it or routed to it by class functions.[115]

19. means for said router to find information for packets it receives regarding the path by which said packets came to said router via packet marks provided by routers leading to said host computer[116]

Plaintiff's proposed construction	**Defendant's proposed construction**	**By:**
Function: finding information for packets received at the router	Indefinite	J

[113] *See* Narayanaswamy Decl. at ¶57.
[114] *See* Doc. No. 1-3 at Figures 2, 3, 4, 5, and 6.
[115] *See* Doc. No. 1-3 at Figures 7, 8, and 9.
[116] Claim 4 of the '497 patent.

regarding the path by which said packets came to said router via packet marks provided by routers leading to said host computer Structure: 3:62 - 4:20; 5:40 - 5:44; 6:46 - 6:51; Figure 3; Figure 6; Figure 9		

20. means in said router for said router to use said information to allocate the transmission rate for unwanted data packets to be less than equal to said maximum acceptable transmission rate[117]

Plaintiff's proposed construction	**Defendant's proposed construction**	**By:**
Function: allocating the transmission rate for unwanted data packets to be less than equal to said maximum acceptable transmission rate Structure: 3:62 - 4:20; 5:40 - 5:44; 6:46 - 6:51; Figure 3; Figure 6; Figure 9	Indefinite	J

21. means for determining a path by which data packets arrive at a host computer via packet marks provided by routers leading to said host computer[118]

Plaintiff's proposed construction	**Defendant's proposed construction**	**By:**
Function: determining a path by which data packets arrive at a host computer via packet marks provided by routers leading to said host computer Structure: 3:30 - 3:42; 5:29 - 5:33; 6:12 - 6:25; Figure 1; Figure 4; Figure 7	Indefinite	J

22. means for assigning a maximum acceptable processing rate to each class of data packet[119]

Plaintiff's proposed construction	**Defendant's proposed construction**	**By:**
Function: assigning a maximum acceptable processing rate to each class of data packet Structure: 3:30 - 3:42; 5:29 - 5:33; 6:12 - 6:25; Figure 1;	Indefinite	J

[117] Claim 4 of the '497 patent.
[118] Claim 13 of the '497 patent.
[119] Claim 13 of the '497 patent.

Figure 4; Figure 7		

23. means for allocating a processing rate equal to or less than said maximum acceptable processing rate to said unwanted data packets[120]

Plaintiff's proposed construction	**Defendant's proposed construction**	**By:**
Function: allocating a processing rate that is equal to or less than said maximum acceptable processing rate to said unwanted data packets Structure: 3:30 - 3:42; 5:29 - 5:33; 6:12 - 6:25; Figure 1; Figure 4; Figure 7	Indefinite	J

Plaintiff groups these five terms together as the disclosure of structure and argument is largely coextensive. However, while Plaintiff contends these terms do not require the 112(f) analysis, Plaintiff's identified function and structure are provided in the boxed chart with argument and explanation following.

The means (1) for the router to find information for packets it receives regarding the path ...; (2) in the router for the router to use the information...; (3) determining a path by which data packets arrive at a host computer...; (4) assigning a maximum acceptable processing rate...; and, (5) allocating a processing rate..., are disclosed in the algorithm shown in Figure 3. However, the structure of a router is provided in the claim language and this term is not subject to 112(f). It is common parlance to one of ordinary skill in the art for a router or firewall to perform each of these functions.[121]

Additionally, the software, or configuration, at the router is explained through the algorithms disclosed in Figure 3 by flowchart and diagram where a Class A and Class B (38) of a

[120] Claim 13 of the '497 patent.
[121] *See* Narayanaswamy Decl. at ¶s 61-62, 64-65, 67-68, 70-71, 73-74.

maximum acceptable processing rate (34) are assigned by the firewall to each data packet (30). The description provides that the computer can use the information, the maximum acceptable processing rate, to allocate the processing rate for each class (38) in a desired way among the places from which packets (30) are transmitted.[122] A router 22 is capable of receiving information regarding maximum acceptable transmission rate 70 for each class 38 of data packet 30 being transmitted to the computer 18 and the router 22 is capable of controlling the rate of transmission of each class 38 of data packets 30 to the computer 18.[123] Figure 2, Figure 1, Figure 4, Figure 5, and Figure 6 illustrate additional embodiments, or algorithms, in which path information associated with each packet can be combined with other properties to classify the packet and assign it a processing rate.[124] Figure 7, Figure 8 and Figure 9 illustrate how a firewall protecting a network communicates the rate limit for packets routed to it or routed to it by class functions.[125]

Lastly, it is common parlance for one of ordinary skill in the art to use a router or a firewall to (1) for the router to find information for packets it receives regarding the path ...; (2) in the router for the router to use the information...; (3) determining a path by which data packets arrive at a host computer...; (4) assigning a maximum acceptable processing rate...; and, (5) allocating a processing rate.... Prior art cited in this patent specification including US5455865 and US 5353353 use similar capabilities to implement other network attack defenses. In this patent specification the same capabilities are utilized to limit the rate of packets with added path information.[126] The '497

[122] *See* id.
[123] *See* Doc. No. 1-3 at 6:46-51.
[124] *See* Doc. No. 1-3 at Figures 2, 3, 4, 5, and 6.
[125] *See* Doc. No. 1-3 at Figures 7, 8, and 9.
[126] *See* Narayanaswamy Decl. at ¶s 53, 55.

patent's specification describes how processing rate limits can be used to allocate the processing rates available for packets of each class in a desired manner.[127]

IV. CONCLUSION

Plaintiff respectfully requests the Court adopt its constructions.

Respectfully submitted,

Ramey & Schwaller, LLP



William P. Ramey, III
Texas Bar No. 24027643
wramey@rameyfirm.com
5020 Montrose Blvd., Ste. 800
Houston, Texas 77006
(713)426-3923
(832)900-4941 (fax)

Attorneys for PacSec3, LLC

CERTIFICATE OF SERVICE

The undersigned certifies that the foregoing document was filed electronically in compliance with local rules and the rules of civil procedure. As such, this pleading was served on all counsel who have consented to electronic service on April 12, 2021.

William P. Ramey, III
William P. Ramey, III

[127] *See* Doc. No. 1-3 at 3:30-42.

Exhibit 1015

Declaration of Dr. Stefan Savage

Customer No.: 000027683

Haynes and Boone, LLP
IP Section
2323 Victory Avenue, Suite 700
Dallas, Texas 75219
Telephone [214] 651.5000
Fax [214] 200.0853

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE CENTRAL REEXAMINATION UNIT

UNIFIED PATENTS, LLC

Third Party Requester

Ex Parte Reexamination of

U.S. Patent 7,523,497

DECLARATION OF STEFAN SAVAGE, Ph.D.

I, Stefan Savage, do hereby declare as follows:

1. I have been retained by Unified Patents, LLC ("Unified") to provide testimony regarding the publication and distribution of my article, Savage et al., "Practical Network Support for IP Traceback," SIGCOMM'00: Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 2000, pp. 295-306 (hereinafter "*Savage paper*"). I have been informed by Unified that the *Savage paper* is being submitted in this proceeding as Exhibit 1005. A copy of the *Savage paper* is attached to my declaration as Attachment A.

2. I am being compensated for my services in preparing this Declaration. No part of my compensation is dependent on my opinions or the outcome of this proceeding.

3. I am currently a Professor in the Department of Computer Science and Engineering at the University of California, San Diego (UCSD). At UCSD, I serve on the faculty of the Systems & Networking research group, and the Security research group. I started at UCSD as an Assistant Professor in July of 2000. I was promoted to Associate Professor in July of 2006, and then to Full Professor in July of 2010—a title which I retain today. I also served as the Associate Director of UCSD's Center for Networked Systems starting in July of 2009, and became the Director in the following year. I have continued to serve as Director (now co-

Director) since then. My *curriculum vitae* is attached to my declaration as Attachment B.[1]

4. My research interests focus on networking and network security, including network measurement and traffic analysis, network routing, networking blocking, filtering and rate limiting, and managing network attacks such as denial-of-service attacks. These areas include and subsume the notion of "network data transmission controls." I have been informed that the "Field of the Invention" section of U.S. Patent No. 7,523,497 indicates that the '497 patent "pertains to network data transmission controls."

5. From 1993 to 2021, I have published approximately 150 peer-review articles. My published articles are identified on my *curriculum vitae* (Attachment B) and in the list of publications on my UCSD biography page. The list of publications on my USCD biography page is attached to my declaration as Attachment C.[2]

6. At the time that the *Savage paper* was prepared, published, and distributed in 2000, I was a graduate student at the University of Washington in Seattle, Washington. I earned my Ph.D. in Computer Science and Engineering at

[1] I have been informed that my *curriculum vitae* is also being submitted in this proceeding as Exhibit 1016.
[2] The list of publications on my UCSD biography page is available at http://cseweb.ucsd.edu/~savage/pubs.htm.

the University of Washington in January 2002.

7. I was the primary author of the *Savage paper*. I co-authored the *Savage paper* with David Wetherall, Anna Karlin, and Tom Anderson, all of whom served on the faculty of the Computer Science and Engineering Department at the University of Washington at the time the *Savage paper* was prepared.

8. As discussed in more detail below, I personally presented the *Savage paper* at the ACM SIGCOMM 2000 Conference in Stockholm, Sweden, on September 1, 2000. A printed copy of the *Savage paper* was distributed to all attendees of the ACM SIGCOMM 2000 Conference. Before, during, and after the ACM SIGCOMM 2000 Conference, the *Savage paper* was available to be downloaded without restriction on ACM's website advertising the itinerary and content of the ACM SIGCOMM 2000 Conference.

9. SIGCOMM is the annual conference of the Special Interest Group on Data Communication (SIGCOMM) of the Association for Computing Machinery (ACM). According to the SIGCOMM website (Attachment D),[3] "SIGCOMM is ACM's professional forum for discussing communications and computer networks.

[3] The SIGCOMM website is available at http://www.sigcomm.org/. I have been informed that Attachment D is also being submitted in this proceeding as Exhibit 1023.

SIGCOMM members include scientists, engineers, educators and students. They study all aspects of computer communications and networks: analysis, technical design, engineering, measurement and management. Our members are particularly interested in the systems engineering and architectural questions surrounding computer communication." This description of SIGCOMM is consistent with my understanding of the objectives and membership of SIGCOMM in the year 2000 when I attended and presented at the ACM SIGCOMM 2000 conference. I previously attended and presented at the ACM SIGCOMM 1999 Conference in Cambridge, Massachusetts in September 1999 (*see* Attachment C, p. 11),[4] and I attended the ACM SIGCOMM Conference in Vancouver, British Columbia in September 1998. Thus, I am familiar with the objectives and membership of SIGCOMM as well as the nature of the SIGCOMM conference and the distribution of papers at the SIGCOMM conference. At the time of the ACM SIGCOMM 2000 Conference, SIGCOMM was (and still is) the best-known academic conference in network communication protocols, and is well-known to professors, students, and professionals working in the field of network communication protocols.

[4] I presented the paper Savage et al., "The End-to-end Effects of Internet Path Selection," Proceedings of the ACM SIGCOMM Conference 1999, Cambridge, MA (Sept. 1999), pp. 289-299.

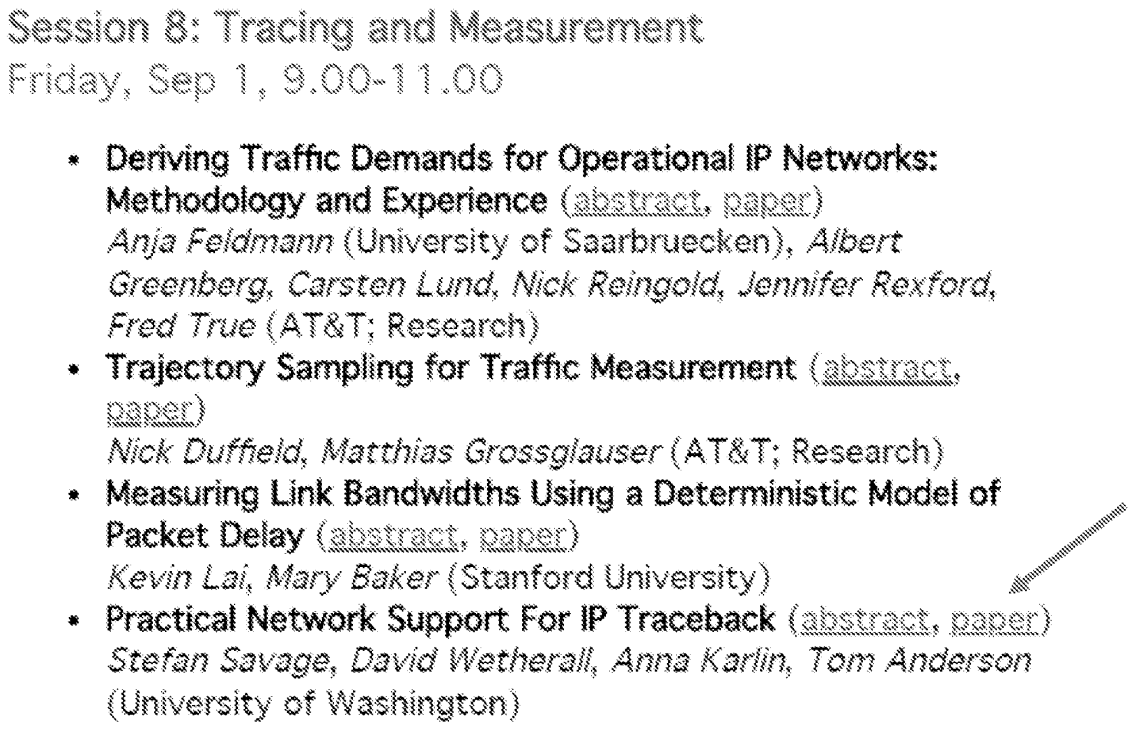
10. I personally presented the *Savage paper* at the ACM SIGCOMM 2000 Conference in Stockholm, Sweden on Friday, September 1, 2000. I presented the *Savage paper* during Session 8 of the ACM SIGCOMM 2000 Conference on Friday, September 1, 2000, as indicated by the ACM SIGCOMM 2000 Conference program, which is attached to my declaration as Attachment E.[5]

11. The *Savage paper* was available to be accessed and downloaded for free and without restriction from the ACM website advertising the itinerary and content of the ACM SIGCOMM 2000 Conference. Attached to my declaration as Attachment F is an Internet Archive version (archived on August 20, 2000) of the ACM website advertising the itinerary and content of the ACM SIGCOMM 2000 Conference.[6] As shown on page 3 of Attachment F, the *Savage paper* was

[5] I have been informed that the ACM SIGCOMM 2000 Conference program is being submitted as Exhibit 1017 in this proceeding. The ACM SIGCOMM 2000 Conference program is available at http://conferences.sigcomm.org/sigcomm/2000/conf/sigcomm2000-final-prog.pdf.

[6] I have been informed that the Internet Archive version of the ACM website advertising the itinerary and content of the ACM SIGCOMM 2000 Conference is being submitted as Exhibit 1018 in this proceeding. Attachment F is available at https://web.archive.org/web/20000820231540/http:/www.acm.org/sigcomm/sigco

available to be downloaded for free, and without restriction, by any conference attendee or other member of the public (including individuals who did not register for the conference), by at least August 20, 2000. The *Savage paper* was available to be downloaded on the ACM SIGCOMM 2000 Conference program (archived by the Internet Archive) by clicking on the "paper" link following the title of the *Savage paper*, as shown in the screenshot of Attachment F (Session 8 on page 3) below.



Session 8: Tracing and Measurement
Friday, Sep 1, 9.00-11.00

- **Deriving Traffic Demands for Operational IP Networks: Methodology and Experience** (abstract, paper)
  Anja Feldmann (University of Saarbruecken), *Albert Greenberg, Carsten Lund, Nick Reingold, Jennifer Rexford, Fred True* (AT&T; Research)
- **Trajectory Sampling for Traffic Measurement** (abstract, paper)
  Nick Duffield, Matthias Grossglauser (AT&T; Research)
- **Measuring Link Bandwidths Using a Deterministic Model of Packet Delay** (abstract, paper)
  Kevin Lai, Mary Baker (Stanford University)
- **Practical Network Support For IP Traceback** (abstract, paper)
  Stefan Savage, David Wetherall, Anna Karlin, Tom Anderson (University of Washington)

Attachment F, p. 3 (annotated excerpt)

mm2000/conf/techprog.htm.

12. Attachment G to my declaration is the version of the *Savage paper* (downloaded on February 26, 2021) that I personally downloaded from the above-identified Internet Archive version (archived on August 20, 2000, i.e., Attachment F) of the ACM website advertising the itinerary and content of the ACM SIGCOMM 2000 Conference.[7] Attachment G is thus the version of the *Savage paper* that was archived on August 20, 2000 by the Internet Archive and would have been available to anyone accessing the ACM conference website on August 20, 2000. Based on my comparison, Attachment G is identical to Attachment A (*Savage paper*).

13. Attachment H to my declaration is a printout of the currently available version (downloaded on February 24, 2021) of ACM's website advertising the itinerary and content of the ACM SIGCOMM 2000 Conference.[8] Other than the

[7] The file that I downloaded by clicking the "paper" link following the title of the *Savage paper*, as shown in the above screenshot of Attachment F, was provided as a gzip-compressed PostScript file (.ps.gz), as was common at that time. I converted that file into a PDF format to allow it to be more easily viewed on modern computers. Attachment G is the PDF version of the file.

[8] Attachment H is available at http://conferences.sigcomm.org/sigcomm/2000/conf/techprog.htm.

Internet Archive banner inserted at the top of each page of Attachment F, Attachments F and H are substantively identical to each other.

14. As shown by the excerpt of Attachment F above (and the corresponding portion of Attachment H, p. 3), the *Savage paper* was identified by title and by author on the ACM website advertising the itinerary and content of the ACM SIGCOMM 2000 Conference. Thus, the *Savage paper* would have been available to anyone performing a title or author search of the ACM website advertising the itinerary and content of the ACM SIGCOMM 2000 Conference. Attachment I provides information on SIGCOMM's objectives in 2000 of providing free access to information presented at SIGCOMM conferences. Attachment I, which is an October 22, 2000 letter to SIGCOMM members from Mr. Gary S. Delp, who was the Secretary – Treasurer of SIGCOMM in 2000, provides that "We have a long tradition of making SIGCOMM information widely accessible. We were one of the first societies to put full newsletter and conference text on-line, and have been adamant about making sure that this access is free to all."

15. When conference attendees checked in at the ACM SIGCOMM 2000 Conference, each conference attendee received a perfect-bound book containing all papers that were to be presented at the conference, including the *Savage paper*. I have kept a copy of the conference book distributed to attendees of the ACM

SIGCOMM 2000 Conference. Attachment J to my declaration is a collection of photographs that I personally took of the conference book distributed to attendees of the ACM SIGCOMM 2000 Conference.[9] The pictures contained in Attachment J are of the conference book that I personally received when I attended the ACM SIGCOMM 2000 Conference. Attachment J includes the cover of the book, the front page, copyright page, individual messages from the conference General Chairs and Program Chairs, the Table of Contents, and the complete pages of the *Savage paper* contained in the book. Thus, each attendee of the ACM SIGCOMM 2000 Conference received a physical copy of the *Savage paper*, by at least September 1, 2000, the last day of the ACM SIGCOMM 2000 Conference.

16. The front cover of the conference book (Attachment J) indicates that the book is titled "Computer Communication Review" of ACM SIGCOMM, and relates to the "Proceedings of ACM SIGCOMM 2000 Conference" in Stockholm, Sweden on August 28, 2000 to September 1, 2000. Attachment J, p. 1. The cover of the conference book indicates that the book is published as Volume 30, Number 4 in October 2000. It is my understanding that ACM SIGCOMM conference books are distributed to all SIGCOMM members. The October 2000 date

[9] I have been informed that Attachment J is being submitted as Exhibit 1021 in this proceeding.

presumably refers to the date on which the conference book would be distributed to SIGCOMM members who did not attend the ACM SIGCOMM 2000 Conference.[10] Like all attendees of the ACM SIGCOMM 2000 Conference, I received the conference book (Attachment J) when I checked in at the ACM SIGCOMM 2000 Conference in Stockholm, Sweden, and thus received this book before October 2000. Attachment J contains the following information.

- PDF page 3: cover page of the conference book, indicating that book is the "Proceedings of ACM SIGCOMM 2000 Conference" in Stockholm, Sweden on August 28 to September 1, 2000;
- Page ii (PDF page 4): copyright 2000 by ACM;
- Page iii (PDF page 5): a "Conference Chairmen's Message" welcoming attendees to Stockholm and the conference ("Welcome to Stockholm, Sweden and the ACM SIGCOMM 2000 Conference!...We hope that you have a pleasant time in Stockholm!");
- Page iv (PDF page 6): "A Message from the Program Committee Chairs" explaining the submission of papers (238 papers were

[10] As indicated on the second page of Attachment J (bottom of page), "Computer Communication Review" is published five times a year, with two issues being published in October.

submitted) and the "full program of 26 high quality papers" among those submitted;

- Pages v-vii (PDF pages 7-9): Table of Contents including the *Savage paper* presented during Session 8 on September 1, 2000; and
- Pages 295-306 (PDF pages 10-21): the *Savage paper*.

17. While I do not recall the exact number, I recall there being approximately 300 to 400 attendees at the ACM SIGCOMM 2000 Conference. Dr. Kevin Almeroth, who is a Professor of Computer Science at the University of California, Santa Barbara, maintains a w§ebsite titled "Networking Conferences Statistics" (Attachment K) that "gather[s] year-over-year statistics for some of the major systems and networking conferences." Attachment K, at p. 1. According to this website, there were 407 attendees at the ACM SIGCOMM 2000 Conference. Attachment K, at p. 39. This number is consistent with my recollection of the number of attendees at the ACM SIGCOMM 2000 Conference.

18. In addition to the number of attendees who attended the ACM SIGCOMM 2000 Conference in person, the ACM SIGCOMM 2000 Conference was also simulcast at the time via (then cutting edge) multicast Internet video streaming. The multicast distribution of the ACM SIGCOMM 2000 Conference is indicated at Attachment L, p. 1 ("August 2000: The technical program will be

multicast on the Internet.").[11] Further, Attachment M, which is a portion of the "Conference Program" website, is directed to the multicast transmission of the ACM SIGCOMM 2000 Conference. Attachment M indicates that "We will multicast video and audio from the technical sessions of the conference live on the Internet. The transmission will start Wednesday August 30 at 9:00 and end Friday September 1 at 13:30." Attachment M, p. 1.[12] Thus, my presentation of the *Savage paper* was also multicast on the Internet, in addition to my in-person, live presentation of the *Savage paper* at the ACM SIGCOMM 2000 Conference on September 1, 2000.

19. In addition to the distribution of the *Savage paper* to all conference attendees, I presented the slides contained in Attachment N to the conference attendees on September 1, 2000.[13] These slides were presented in conjunction with the presentation of the *Savage paper*, which was presented and distributed to conference attendees as described above.

---

[11] Attachment L is available at http://conferences.sigcomm.org/sigcomm/2000/.

[12] Attachment M is available at http://conferences.sigcomm.org/sigcomm/2000/conf/multicast.htm.

[13] The presentation slides contained in Attachment N are available for download at: http://cseweb.ucsd.edu/~savage/papers/Sigcomm00Slides.pdf.

20. The *Savage paper* presented at the ACM SIGCOMM 2000 Conference was one of several papers presented at the conference that were selected to be included in the IEEE/ACM Transactions on Networking (ToN) Journal, which is a joint publication by IEEE (Institute of Electrical and Electronics Engineers) and the ACM. The *Savage paper* was included (i.e., re-published) in the June 2001 Issue (Issue 3) of the IEEE/ACM Transactions on Networking.[14]

21. I have been informed that U.S. Patent No. 7,523,497 ("the '497 patent"), the subject of this proceeding, contains the following information in its "Field of Invention" section:

> The invention pertains to network data transmission controls. More particularly, the invention relates to systems for minimizing the effects of packet flooding attacks directed against computers or routers connected to a network.

I have also been informed that the '497 patent is related to U.S. Patent No. 6,789,190, which was filed on November 16, 2000.

22. At the time of the ACM SIGCOMM 2000 Conference when the *Savage paper* was presented and distributed to conference attendees (at least by

[14] The June 2001 Issue of the IEEE/ACM Transactions on Networking is available at https://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=20107&punumber=90.

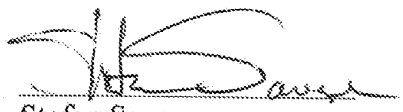
September 1, 2000), I was a person interested and ordinarily skilled in the above-identified subject matter of the '497 patent. As indicated by my *curriculum vitae* (Attachment B), I was employed by five technology companies from 1993 to 2000, including Honeywell Corporation, Hewlett Packard Corporation, Digital Equipment Corporation, Appliant, and Asta Networks. For these companies, I regularly practiced in the area of network data transmission controls. For instance, Asta Networks was a startup, for which I was a co-founder, that was particularly focused on denial-of-service (DoS) attacks. Further, during 1999 and 2000, I co-authored over 10 articles, papers, and presentations in the field of networking and network communications protocols. *See* Attachment B, pp. 11-27. In addition, at the time of the ACM SIGCOMM 2000 Conference, I was a graduate student at the University of Washington in Seattle, Washington studying in the field of Computer Science and Engineering (I earned my Ph.D. in Computer Science and Engineering at the University of Washington in January 2002). The *Savage paper*, of which I was the primary author, is also directed to the same field as the '497 patent. For instance, the Abstract of the *Savage paper* reads, in part: "This paper describes a technique for tracing anonymous packet flooding attacks in the Internet back towards their source. This work is motivated by the increased frequency and sophistication of denial-of-service attacks and by the difficulty in tracing packets with incorrect, or "spoofed", source addresses." Thus, at the time of the ACM

SIGCOMM 2000 Conference when the *Savage paper* was presented and distributed to conference attendees, I was a person interested and ordinarily skilled in the subject matter of the '497 patent.

23. In signing this declaration, I understand that the declaration will be filed as evidence in a proceeding before the Central Reexamination Unit of the U.S. Patent and Trademark Office. If cross-examination is required of me, I will appear for cross-examination during the time allotted for cross-examination.

24. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

4/23/2021
Date


Stefan Savage

ATTACHMENT A
(EXHIBIT 1005)

Practical Network Support for IP Traceback

Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson

Department of Computer Science and Engineering
University of Washington
Seattle, WA, USA

Abstract

This paper describes a technique for tracing anonymous packet flooding attacks in the Internet back towards their source. This work is motivated by the increased frequency and sophistication of denial-of-service attacks and by the difficulty in tracing packets with incorrect, or "spoofed", source addresses. In this paper we describe a general purpose traceback mechanism based on probabilistic packet marking in the network. Our approach allows a victim to identify the network path(s) traversed by attack traffic without requiring interactive operational support from Internet Service Providers (ISPs). Moreover, this traceback can be performed "post-mortem" – after an attack has completed. We present an implementation of this technology that is incrementally deployable, (mostly) backwards compatible and can be efficiently implemented using conventional technology.

1. INTRODUCTION

Denial-of-service attacks consume the resources of a remote host or network, thereby denying or degrading service to legitimate users. Such attacks are among the hardest security problems to address because they are simple to implement, difficult to prevent, and very difficult to trace. In the last several years, Internet denial-of-service attacks have increased in frequency, severity and sophistication. Howard reports that between the years of 1989 and 1995, the number of such attacks reported to the Computer Emergency Response Team (CERT) increased by 50 percent per year [25]. More recently, a 1999 CSI/FBI survey reports that 32 percent of respondents detected denial-of-service attacks directed against their sites [16]. Even more worrying, recent reports indicate that attackers have developed tools to coordinate distributed attacks from many separate sites [14].

Unfortunately, mechanisms for dealing with denial-of-service have not advanced at the same pace. Most work in this area has focused on *tolerating* attacks by mitigating their effects on the victim [38, 2, 26, 29, 9]. This approach can provide an effective stop-gap measure, but does not eliminate the problem nor does it discourage attackers. The other option, and the focus of this paper, is to trace attacks back towards their origin – ideally stopping an attacker at the source.

A perfect solution to this problem is complicated by the potential use of indirection to "launder" the true causal origin of an attack. For example, an attack may consist of packets sent from many different slave machines, themselves under the control of a remote master machine. Such indirection may be achieved either explicitly (by compromising the individual slave hosts directly) or implicitly (by sending false requests to the slaves on behalf of the victim – a so-called *reflector*). More challenging still, the true origin and identity of the attacker can be similarly concealed through chains of false computer accounts, call forwarding, and so forth. Consequently, we regard a complete solution – particularly one able to address the forensic needs of law enforcement – as an open problem.

Instead, we address the more limited operational goal of simply identifying the machines that *directly* generate attack traffic and the network path this traffic subsequently follows. We call this the *traceback problem* and it is motivated by the operational need to control and contain attacks. In this setting, even incomplete or approximate information is valuable because the efficacy of measures such as packet filtering improve as they are applied further from the victim and closer to the source.

However, even for our restricted problem, determining the source generating attack traffic is surprisingly difficult due to the stateless nature of Internet routing. Attackers routinely disguise their location using incorrect, or "spoofed", IP source addresses. As these packets traverse the Internet their true origin is lost and a victim is left with little useful information. While there are several ad hoc traceback techniques in use, they all have significant drawbacks that limit their practical utility in the current Internet.

In this paper we present a new approach to the traceback problem that addresses the needs of both victims and network operators. Our solution is to probabilistically mark packets with partial path information as they arrive at routers. This approach exploits the observation that attacks generally comprise large numbers of packets. While each marked packet represents only a "sample" of the path it has traversed, by combining a modest number of such packets a victim can reconstruct the entire path. This allows victims to locate the approximate source of attack traffic without requiring the assistance of outside network operators. Moreover, this determination can be made even after an attack has completed. Both facets of our solution represent substantial improvements over existing capabilities for dealing with flooding-style denial-of-service attacks.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGCOMM'00, Stockholm, Sweden.
Copyright 2000 ACM 1-58113-224-7/00/0008...$5.00.

A key practical deployment issue with any modification of Internet routers is to ensure that the mechanisms are efficiently implementable, may be incrementally deployed, and are backwards compatible with the existing infrastructure. We describe a traceback algorithm that adds little or no overhead to the router's critical forwarding path and may be incrementally deployed to allow traceback within the subset of routers supporting our scheme. Further, we demonstrate that we can encode the necessary path information in a way that peacefully co-exists with existing routers, host systems and more than 99% of today's traffic.

The rest of this paper is organized as follows: In Section 2, we describe related work concerning IP spoofing and solutions to the traceback problem. Section 3 outlines our basic approach and section 4 characterizes several abstract algorithms for implementing it. In Section 5 we detail a concrete encoding strategy for our algorithm that can be implemented within the current Internet environment. We also present experimental results demonstrating the effectiveness of our solution. In section 6 we discuss the main limitations and weaknesses of our proposal and potential extensions to address some of them. Finally, we summarize our findings in Section 7.

2. RELATED WORK

It has been long understood that the IP protocol permits anonymous attacks. In his 1985 paper on TCP/IP weaknesses, Morris writes:

> "The weakness in this scheme [the Internet Protocol] is that the source host itself fills in the IP source host id, and there is no provision in ... TCP/IP to discover the true origin of a packet." [31]

In addition to denial-of-service attacks, IP spoofing can be used in conjunction with other vulnerabilities to implement anonymous one-way TCP channels and covert port scanning [31, 3, 24, 44].

There have been several efforts to reduce the anonymity afforded by IP spoofing. Table 1 provides a subjective characterization of each of these approaches in terms of management cost, additional network load, overhead on the router, the ability to trace multiple simultaneous attacks, the ability trace attacks after they have completed, and whether they are preventative or reactive. We also characterize our proposed traceback scheme according to the same criteria. In the remainder of this section we describe each previous approach in more detail.

2.1 Ingress filtering

One way to address the problem of anonymous attacks is to eliminate the ability to forge source addresses. One such approach, frequently called *ingress filtering*, is to configure routers to block packets that arrive with illegitimate source addresses [20]. This requires a router with sufficient power to examine the source address of every packet and sufficient knowledge to distinguish between legitimate and illegitimate addresses. Consequently, ingress filtering is most feasible in customer networks or at the border of Internet Service Providers (ISP) where address ownership is relatively unambiguous and traffic load is low. As traffic is aggregated from multiple ISPs into transit networks, there is no longer enough information to unambiguously determine if a packet arriving on a particular interface has a "legal" source address. Moreover, on many deployed router architectures the overhead of ingress filter becomes prohibitive on high-speed links.

The principal problem with ingress filtering is that its effectiveness depends on widespread, if not universal, deployment. Unfortunately, a significant fraction of ISPs, perhaps the majority, do not implement this service – either because they are uninformed or have been discouraged by the administrative burden[1], potential router overhead and complications with existing services that depend on source address spoofing (e.g. some versions of Mobile IP [33] and some hybrid satellite communications architectures). A secondary problem is that even if ingress filtering were universally deployed at the customer-to-ISP level, attackers could still forge addresses from the hundreds or thousands of hosts within a valid customer network [14].

It is clear that wider use of ingress filtering would dramatically improve the Internet's robustness to denial-of-service attacks. At the same time it is prudent to assume that such a system will never be fullproof – and therefore traceback technologies will continue to be important.

2.2 Link testing

Most existing traceback techniques start from the router closest to the victim and interactively test its upstream links until they determine which one is used to carry the attacker's traffic. Ideally, this procedure is repeated recursively on the upstream router until the source is reached. This technique assumes that an attack remains active until the completion of a trace and is therefore inappropriate for attacks that are detected after the fact, attacks that occur intermittently, or attacks that modulate their behavior in response to a traceback (it is prudent to assume the attacker is fully informed). Below we describe two varieties of link testing schemes, *input debugging* and *controlled flooding*.

2.2.1 Input debugging

Many routers include a feature called *input debugging*, that allows an operator to filter particular packets on some egress port and determine which ingress port they arrived on. This capability is used to implement a trace as follows: First, the victim must recognize that it is being attacked and develop an *attack signature* that describes a common feature contained in all the attack packets. The victim communicates this signature to a network operator, frequently via telephone, who then installs a corresponding input debugging filter on the victim's upstream egress port. This filter reveals the associated input port, and hence which upstream router originated the traffic. The process is then repeated recursively on the upstream router, until the originating site is reached or the trace leaves the ISP's border (and hence its administrative control over the routers). In the later case, the upstream ISP must be contacted and the procedure repeats itself. While such tracing is frequently performed manually, several ISPs have developed tools to automatically trace attacks across their own networks [41].

The most obvious problem with the input debugging approach, even with automated tools, is its considerable management overhead. Communicating and coordinating with network operators at multiple ISPs requires the time, attention and commitment of both the victim and the remote personnel – many of whom have no direct economic incentive to provide aid. If the appropriate network

[1] Some modern routers ease the administrative burden of ingress filtering by providing functionality to automatically check source addresses against the destination-based routing tables (e.g. `ip verify unicast reverse-path` on Cisco's IOS). This approach is only valid if the route to and from the customer is symmetric – generally at the border of single-homed stub networks.

	Management overhead	Network overhead	Router overhead	Distributed capability	Post-mortem capability	Preventative/ reactive
Ingress filtering	Moderate	Low	Moderate	N/A	N/A	Preventative
Link testing						
Input debugging	High	Low	High	Good	Poor	Reactive
Controlled flooding	Low	High	Low	Poor	Poor	Reactive
Logging	High	Low	High	Excellent	Excellent	Reactive
ICMP Traceback	Low	Low	Low	Good	Excellent	Reactive
Marking	Low	Low	Low	Good	Excellent	Reactive

Table 1: Qualitative comparison of existing schemes for combating anonymous attacks and the probabilistic marking approach we propose.

operators are not available, if they are unwilling to assist, or if they do not have the appropriate technical skills and capabilities, then a traceback may be slow or impossible to complete [21].

2.2.2 *Controlled flooding*

Burch and Cheswick have developed a link testing traceback technique that does not require any support from network operators [6]. We call this technique *controlled flooding* because it tests links by flooding them with large bursts of traffic and observing how this perturbs traffic from the attacker. Using a pre-generated "map" of Internet topology, the victim coerces selected hosts along the upstream route into iteratively flooding each incoming link on the router closest to the victim. Since router buffers are shared, packets traveling across the loaded link – including any sent by the attacker – have an increased probability of being dropped. By observing changes in the rate of packets received from the attacker, the victim can therefore infer which link they arrived from. As with other link testing schemes, the basic procedure is then applied recursively on the next upstream router until the source is reached.

While the scheme is both ingenious and pragmatic, it has several drawbacks and limitations. Most problematic among these is that controlled flooding is itself a denial-of-service attack – exploiting vulnerabilities in unsuspecting hosts to achieve its ends. This drawback alone makes it unsuitable for routine use. Also, controlled flooding requires the victim to have a good topological map of large sections of the Internet in addition to an associated list of "willing" flooding hosts. As Burch and Cheswick note, controlled flooding is also poorly suited for tracing distributed denial-of-service attacks because the link-testing mechanism is inherently noisy and it can be difficult to discern the set of paths being exploited when multiple upstream links are contributing to the attack. Finally, like all link-testing schemes, controlled flooding is only effective at tracing an on-going attack and cannot be used "post-mortem".

2.3 Logging

An approach suggested in [37] and [41] is to log packets at key routers and then use data mining techniques to determine the path that the packets traversed. This scheme has the useful property that it can trace an attack long after the attack has completed. However, it also has obvious drawbacks, including potentially enormous resource requirements (possibly addressed by sampling) and a large scale inter-provider database integration problem. We are unaware of any commercial organizations using a fully operational traceback approach based on logging[2].

[2]Historically, the T3-NFSNET *did* log network-to-network traffic statistics and these were used on at least one occasion to trace IP spoofing attacks to an upstream provider [43].

2.4 ICMP Traceback

Since the first writing of this paper, a new traceback proposal bas emerged based on the use of explicit router-generated ICMP traceback messages [4]. The principle idea in this scheme is for every router to sample, with low probability (e.g., 1/20,000), one of the packets it is forwarding and copy the contents into a special ICMP traceback message including information about the adjacent routers along the path to the destination. During a flooding-style attack, the victim host can then use these messages to reconstruct a path back to the attacker. This scheme has many benefits compared to previous work and is in many ways similar to the packet marking approach we have taken. However, there are several disadvantages in the current design that complicate its use. Among these: ICMP traffic is increasingly differentiated and may be filtered or rate limited differently from normal traffic, the ICMP Traceback message relies on an input debugging capability (i.e. the ability to associate a packet with the input port and/or MAC address on which it arrived) that is not available in some router architectures, if only some of the routers participate it seems difficult to positively "connect" traceback messages from participating routers separated by a non-participating router, and finally, it requires a key distribution infrastructure to deal with the problem of attackers sending false ICMP Traceback messages. That said, we believe that the scheme is promising and that hybrid approaches combining it with some of the algorithms we propose are likely to be quite effective.

3. OVERVIEW

Burch and Cheswick mention the possibility of tracing flooding attacks by "marking" packets, either probabilistically or deterministically, with the addresses of the routers they traverse [6]. The victim uses the information in the marked packets to trace an attack back to its source. This approach has not been previously explored in any depth, but has many potential advantages. It does not require interactive cooperation with ISPs and therefore avoids the high management overhead of input debugging. Unlike controlled flooding, it does not require significant additional network traffic and can potentially be used to track multiple attacks. Moreover, like logging, packet marking can be used to trace attacks "post-mortem" – long after the attack has stopped. Finally, we have found that marking algorithms can be implemented without incurring any significant overhead on network routers. The remainder of this paper focuses on fully exploring and characterizing this approach.

3.1 Definitions

Figure 1 depicts the network as seen from a victim V. For the purposes of this paper, V may be a single host under attack, or a network border device such as a firewall or intrusion detection system that represents many such hosts. Every potential *attack origin* A_i is a leaf in a tree rooted at V and every router R_i is an internal

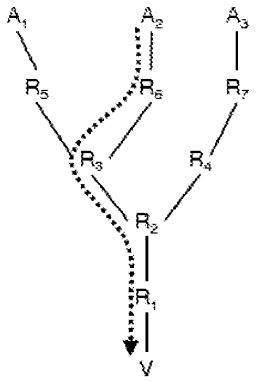


Figure 1: Network as seen from the victim of an attack, V. Routers are represented by R_i, and potential attackers by A_i. The dotted line represents a particular *attack path* between an attacker and the victim.

node along a path between some A_i and V. The *attack path* from A_i is the unique ordered list of routers between A_i and V. For instance, if an attack originates from A_2 then to reach V it must first traverse the path R_6, R_3, R_2, and R_1 – as shown by the dotted line in Figure 1.

The *exact traceback* problem is to determine the attack path and the associated attack origin for each attacker. However, solving this problem is complicated by several practical limitations. The exact attack origin may never be revealed (even MAC source addresses may be spoofed) and a wily attacker may send false signals to "invent" additional routers in the traceback path. We address these issues in section 6, but for now we restrict our discussion to solving a more limited problem. We define the *approximate traceback* problem as finding a candidate attack path for each attacker that contains the true attack path as a suffix. We call this the *valid suffix* of the candidate path. For example, $(R_5, R_6, R_3, R_2, R_1)$ is a valid approximate solution to Figure 1 because it contains the true attack path as a suffix. We say a solution to this problem is *robust* if an attacker cannot prevent the victim from discovering candidate paths containing the valid suffix.

All marking algorithms have two components: a *marking procedure* executed by routers in the network and a *path reconstruction procedure* implemented by the victim. A router "marks" one or more packets by augmenting them with additional information about the path they are traveling. The victim attempts to reconstruct the attack path using only the information in these marked packets. The *convergence time* of an algorithm is the number of packets that the victim must observe to reconstruct the attack path.

3.2 Basic assumptions

The design space of possible marking algorithms is large, and to place our work in context we identify the assumptions that motivate and constrain our design:

- an attacker may generate any packet,
- multiple attackers may conspire,
- attackers may be aware they are being traced,
- packets may be lost or reordered,
- attackers send numerous packets,
- the route between attacker and victim is fairly stable,
- routers are both CPU and memory limited, and
- routers are not widely compromised.

The first four assumptions represent conservative assessments of the abilities of the modern attackers and limitations of the network. Designing a traceback system for the Internet environment is extremely challenging because there is very little that can be trusted. In particular, the attacker's ability to create arbitrary packets significantly constrains potential solutions. When a router receives a packet, it has no way to tell whether that packet has been marked by an upstream router or if the attacker simply has forged this information. In fact, the only invariant that we can depend on is that a packet from the attacker must traverse all of the routers between it and the victim.

The remaining assumptions reflect the basis for our design and deserve additional discussion. First, denial-of-service attacks are only effective so long as they occupy the resources of the victim. Consequently, most attacks are comprised of thousands or millions of packets. Our approach relies on this property because we mark each packet with only a small piece of path state and the victim must observe many such packets to reconstruct the complete path back the the attacker. If many attacks emerge that require only a single packet to disable a host (e.g. ping-of-death [11]), then this assumption may not hold (although we note that even these attacks require multiple packets to *keep* a machine down).

Second, measurement evidence suggests that while Internet routes do change, it is extremely rare for packets to follow many different paths over the short time-scales of a traceback operation (seconds in our system) [32]. This assumption greatly simplifies the role of the victim, since it can therefore limit its consideration to a single primary path for each attacker. If the Internet evolves to allow significant degrees of multi-path routing then this assumption may not hold.

Third, while there have been considerable improvements in router implementation technology, link speeds have also increased dramatically. Consequently, we assert that any viable implementation must have low per-packet overhead and must not require per-flow state. Significantly simpler schemes than ours can be implemented if we assume that routers are not resource constrained.

Finally, since a compromised router can effectively eliminate any information provided by upstream routers, it is effectively indistinguishable from an attacker. In such circumstances, the security violation at the router must be addressed first, before any further traceback is attempted. In normal circumstances, we believe this is an acceptable design point. However, if non-malicious, but information hiding, routing infrastructures become popular, such as described in [22, 35], then this issue may need to be revisited.

4. BASIC MARKING ALGORITHMS

In this section we describe a series of marking algorithms – starting from the most simple and advancing in complexity. Each algorithm attempts to solve the approximate traceback problem in a manner consistent with our assumptions.

Marking procedure at router R:
for each packet w, append R to w

Path reconstruction procedure at victim v:
for any packet w from attacker
extract path $(R_i..R_j)$ from the suffix of w

Figure 2: Node append algorithm.

4.1 Node append

The simplest marking algorithm – conceptually similar to the IP Record Route option [34] – is to append each node's address to the end of the packet as it travels through the network from attacker to victim (see Figure 2). Consequently, every packet received by the victim arrives with a complete ordered list of the routers it traversed – a built-in attack path.

The node append algorithm is both robust and extremely quick to converge (a single packet), however it has several serious limitations. Principal among these is the infeasibly high router overhead incurred by appending data to packets in flight. Moreover, since the length of the path is not known *a priori*, it is impossible to ensure that there is sufficient unused space in the packet for the complete list. This can lead to unnecessary fragmentation and bad interactions with services such as MTU discovery [30]. This problem cannot be solved by reserving "enough" space, as the attacker can completely fill any such space with false, or misleading, path information.

4.2 Node sampling

To reduce both the router overhead and the per-packet space requirement, we can sample the path one node at a time instead of recording the entire path. A single static "node" field is reserved in the packet header – large enough to hold a single router address (i.e. 32 bits for IPv4). Upon receiving a packet, each router chooses to write its address in the node field with some probability p. After enough packets have been sent, the victim will have received at least one sample for every router in the attack path. As stated in section 3, we assume that the attacker sends enough packets and the route is stable enough that this sampling can converge.

Although it might seem impossible to reconstruct an ordered path given only an unordered collection of node samples, it turns out that with a sufficient number of trials, the order can be deduced from the relative number of samples per node. Since routers are arranged serially, the probability that a packet will be marked by a router and then left unmolested by all downstream routers is a strictly decreasing function of the distance to the victim. If we constrain p to be identical at each router, then the probability of receiving a marked packet from a router d hops away is $p(1-p)^{d-1}$. Since this function is monotonic in the distance from the victim, ranking each router by the number of samples it contributes will tend to produce the accurate attack path. The full algorithm is shown in Figure 3.

Putting aside for the moment the difficulty in changing the IP header to add a 32-bit node field, this algorithm is efficient to implement because it only requires the addition of a write and checksum update to the forwarding path. Current high-speed routers already must perform these operations efficiently to update the *time-to-live* field on each hop. Moreover, if $p > 0.5$ then this algorithm is robust against a single attacker because there is no way for an attacker to insert a "false" router into the path's valid suffix by contributing

Marking procedure at router R:
for each packet w
let x be a random number from [0..1)
if $x < p$ then,
write R into w.node

Path reconstruction procedure at victim v:
let $NodeTbl$ be a table of tuples (node,count)
for each packet w from attacker
$z :=$ lookup w.node in $NodeTbl$
if z != NIL then
increment z.count
else
insert tuple (w.node,1) in $NodeTbl$
sort $NodeTbl$ by count
extract path $(R_i..R_j)$ from ordered node fields in $NodeTbl$

Figure 3: Node sampling algorithm.

more samples than a downstream router, nor to reorder valid routers in the path by contributing more samples than the difference between any two downstream routers.

However, there are also two serious limitations. First, inferring the total router order from the distribution of samples is a slow process. Routers far away from the victim contribute relatively few samples (especially since p must be large) and random variability can easily lead to misordering unless a very large number of samples are observed. For instance, if $d = 15$ and $p = 0.51$, the receiver must receive more than 42,000 packets on average before it receives a *single* sample from the furthest router. To guarantee that the order is correct with 95% certainty requires more than seven times that number.

Second, if there are multiple attackers then multiple routers may exist at the same distance – and hence be sampled with the sample probability. Therefore, this technique is not robust against multiple attackers.

4.3 Edge sampling

A straightforward solution to these problems is to explicitly encode *edges* in the attack path rather than simply individual nodes. To do this, we would need to reserve *two* static address-sized fields, *start* and *end*, in each packet to represent the routers at each end of a link, as well as an additional small field to represent the distance of an edge sample from the victim.

When a router decides to mark a packet, it writes its own address into the start field and writes a zero into the distance field. Otherwise, if the distance field is already zero this indicates that the packet was marked by the previous router. In this case, the router writes its own address into the end field – thereby representing the edge between itself and the previous router. Finally, if the router doesn't mark the packet then it always increments the distance field. This somewhat baroque signaling mechanism allows edge sampling to be incrementally deployed – edges are constructed only between participating routers.

The mandatory increment is *critical* to minimize spoofing by an attacker. When the packet arrives at the victim its distance field represents the number of hops traversed since the edge it contains

Marking procedure at router R:

```
for each packet w
    let x be a random number from [0..1)
    if x < p then
        write R into w.start and 0 into w.distance
    else
        if w.distance = 0 then
            write R into w.end
        increment w.distance
```

Path reconstruction procedure at victim v:

```
let G be a tree with root v
let edges in G be tuples (start,end,distance)
for each packet w from attacker
    if w.distance = 0 then
        insert edge (w.start,v,0) into G
    else
        insert edge (w.start,w.end,w.distance) into G
remove any edge (x,y,d) with d ≠ distance from x to v in G
extract path (R_i..R_j) by enumerating acyclic paths in G
```

Figure 4: Edge sampling algorithm.

was sampled.[3] Any packets written by the attacker will necessarily have a distance greater or equal to the length of the true attack path. Therefore, a single attacker is unable to forge any edge between themselves and the victim (for a distributed attack, of course, this applies only to the closest attacker) and the victim does not have to worry about "chaff" while reconstructing the valid suffix of the attack path. Consequently, since we no longer use the sampling rank approach to distinguish "false" samples, we are free to use arbitrary values for the marking probability p.

The victim uses the edges sampled in these packets to create a graph (much as in Figure 1) leading back to the source, or sources, of attack. The full algorithm is described in Figure 4. Because the probability of receiving a sample is geometrically smaller the further away it is from the victim, the time for this algorithm to converge is dominated by the the time to receive a sample from the furthest router, $\frac{1}{p(1-p)^{d-1}}$ in expectation, for a router d hops away. However, there is a small probability that we will receive a sample from the furthest router, but not from some nearer router. We can bound this effect to a factor of $ln(d)$ by the following argument: We conservatively assume that samples from all of the d routers appear with the same likelihood as the furthest router. Since these probabilities are disjoint, the probability that a given packet will deliver a sample from some router is at least $dp(1-p)^{d-1}$. Finally, as per the well-known *coupon collector* problem, the number of trials required to select one of each of d equi-probable items is $d(ln(d) + O(1))$ [19]. Therefore, the number of packets, X, required for the victim to reconstruct a path of length d has the following bounded expectation:

$$E(X) < \frac{ln(d)}{p(1-p)^{d-1}}$$

For example, if $p = \frac{1}{10}$, and the attack path has a length of 10, then a victim can typically reconstruct this path after receiving 75 packets from the attacker. While this choice of $p = \frac{1}{d}$ is optimal, the convergence time is not overly sensitive to this parameter for the path lengths that occur in the Internet. So long as $p \leq \frac{1}{d}$, the results are generally within a small constant of optimal. In the rest of this paper we will use $p = \frac{1}{25}$ since few paths exceed this length [7, 42, 17]. For comparison, the previous example converges with only 108 packets using $p = \frac{1}{25}$.

This same algorithm can efficiently discern multiple attacks because attackers from different sources produce disjoint edges in the tree structure used during reconstruction. The number of packets needed to reconstruct each path is independent, so the number of packets needed to reconstruct all paths is a linear function of the number of attackers. Finally, edge sampling is also robust. That is it is impossible for any edge closer than the closest attacker to be spoofed, due to the robust distance determination. Conversely, in a distributed attack this also means that it is impossible to trust the contents of any edge *further* away than the closest attacker. As with the ICMP Traceback approach [4], an additional mechanism incorporating a shared secret is required to completely address the problem of attackers spoofing edges.

Of course, a significant practical limitation of this approach is that it requires additional space in the IP packet header and therefore is not backwards compatible. In the next section we discuss a modified version of edge-sampling that addresses this problem, albeit at some cost in performance and a reduction in robustness during large distributed attacks.

5. ENCODING ISSUES

The edge sampling algorithm requires 72 bits of space in every IP packet (two 32-bit IP addresses and 8 bits for distance to represent the theoretical maximum number of hops allowed using IP). It would be possible to directly encode these values into an MPLS label stack [36], to enable traceback within a single homogeneous ISP network. However, our focus is on a heterogeneous environment based purely on IP datagrams. One obvious approach is to store the edge sample data in an IP option, but this is a poor choice for many of the same reasons that the node append algorithm is infeasible – appending additional data to a packet in flight is expensive and there may not be sufficient space to append this data. We could also send this data out-of-band – in a separate packet – but this would add both router and network overhead plus the complexity of a new and incompatible protocol.

Instead, we have developed a modified version of edge sampling that dramatically reduces the space requirement in return for a modest increase in convergence time and a reduction in robustness to multiple attackers. Following an analysis of our algorithm we explore the practical implementation issues and discuss one concrete encoding of this scheme based on overloading the 16-bit IP *identification* field used for fragmentation. Any solution involving such overloading necessarily requires compromises and we stress that our solution reflects only one design point among many potential implementation tradeoffs for this class of algorithm and *does not* necessarily reflect an optimal balance among them.

5.1 Compressed edge fragment sampling

We use three techniques to reduce per-packet storage requirements while preserving robustness. First, we encode each edge in half the space by representing it as the *exclusive-or* (XOR) of the two IP addresses making up the edge, as depicted in Figure 5. When

[3] It is important that distance field is updated using a saturating addition. If the distance field were allowed to wrap, then the attacker could spoof edges close to the victim by sending packets with a distance value close to the maximum.

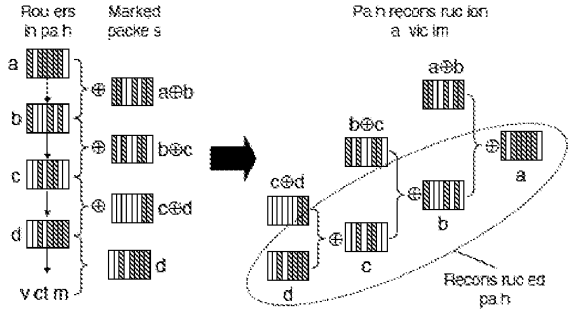


Figure 5: Edge data can be communicated in half the space by sending the XOR of the two nodes (i.e. router IP addresses) making up an edge, rather than sending each node separately. Over time the victim receives the messages d, $c \oplus d$, $b \oplus c$, and $a \oplus b$. By XORing these messages together the original path can be reconstructed.

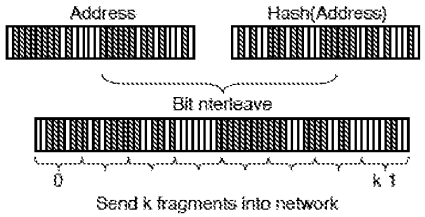


Figure 6: Each router calculates a uniform hash of its IP address once, at startup, using a well-known function. This hash is interleaved with the original IP address (the original address on odd bits, the hash on even bits). The resulting quantity is then broken into k fragments, which the router selects among randomly when marking a packet. Although it is not shown, each of these fragments is further labeled with its offset. The next downstream router uses this offset to select the appropriate fragment to XOR – thereby encoding part of an edge.

some router decides to mark a packet it writes its address, a, into the packet. The following router, b, notices that the distance field is 0 and (assuming it does not mark the packet itself) reads a from the packet, XORs this value with its own address and writes the resulting value, $a \oplus b$, into the packet. We call the resulting value the *edge-id* for the edge between a and b. The edge-ids in the packets received by the victim always contain the XOR of two adjacent routers, except for samples from routers one hop away from the victim, which arrive unmodified. Since $b \oplus a \oplus b = a$, marked packets from the final router can be used to decode the previous edge id, and so on, hop-by-hop until we reach the first router.

Our second modification further reduces our per-packet space requirements by subdividing each edge-id into some number, k, of smaller non-overlapping fragments. When a router decides to mark a packet, it selects one of these fragments at random and stores it in the packet. We use a few additional bits ($log_2 k$) to store the offset of this fragment within the original address – this is necessary to ensure that both fragments making up an edge-id are taken from the same offset. If enough packets are sent by the attacker, the victim will eventually receive all fragments from all edge-ids.

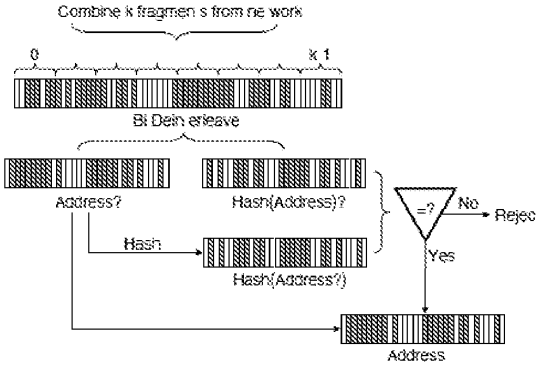


Figure 7: When reconstructing a candidate edge, the victim combines k fragments to produce a bit string. By deinterleaving this string, the address portion and the hash portion are extracted. We recalculate the hash over this address portion using the same hash function used by the router. If the resulting hash is the same as the hash portion extracted, then the address is accepted as valid. This procedure protects against accidentally combining fragments of different edges.

Finally, unlike full IP addresses, edge-id fragments are not unique and multiple fragments from different edge-ids may have the same value. If there are multiple attackers, a victim may receive multiple edge fragments with the same offset and distance. To reduce the probability that we accidentally reconstruct a "false" edge-id by combining fragments from different paths, we add a simple error detection code to our algorithm. We *increase* the size of each router address, and hence each edge-id, by bit-interleaving its IP address with a random hash of itself (depicted in Figure 6). As described earlier, this value is split into fragments, each fragment is selected randomly and stored with an offset, and downstream routers use XOR to combine fragments at the same offset to make up edge-id fragments. The victim constructs *candidate edge-ids* by combining all combinations of fragments at each distance with disjoint offset values. As shown in Figure 7, a candidate edge-id is only accepted if the hash portion matches the data portion for each of its two nodes. As we increase the size of the hash, the probability of a collision is reduced. We provide a describe the full procedure in Figure 8.

The expected number of packets for this algorithm to converge is similar to the edge sampling approach, except now we need k fragments for each edge-id, rather than just one, a total of kd fragments. If we again assume conservatively that each of these fragments is delivered equi-probably with probability $p(1-p)^{d-1}$, the expected number of packets required for path reconstruction is bounded by:

$$E(X) < \frac{k \cdot ln(kd)}{p(1-p)^{d-1}}$$

For example, if there are 8 fragments per edge-id, an attacker is 10 hops away, and $p = \frac{1}{25}$, then a victim can reconstruct the full path after receiving slightly less than 1,300 packets on average. Using techniques similar to those used to show sharp concentration results for the coupon collectors problem, we can further show that the approximate the number of packets required to ensure that a path

Marking procedure at router R:

let R' = BitInterleave(R, Hash(R))
let k be the number of non-overlapping fragments in R'
for each packet w
  let x be a random number from [0..1)
  if $x < p$ then
    let o be a random integer from [0..$k-1$]
    let f be the fragment of R' at offset o
    write f into w.frag
    write 0 into w.distance
    write o into w.offset
  else
    if w.distance = 0 then
      let f be the fragment of R' at offset w.offset
      write $f \oplus w$.frag into w.frag
    increment w.distance

Path reconstruction procedure at victim v:

let $FragTbl$ be a table of tuples (frag,offset,distance)
let G be a tree with root v
let edges in G be tuples (start,end,distance)
let $maxd := 0$
let $last := v$
for each packet w from attacker
  $FragTbl$.Insert(w.frag,w.offset,w.distance)
  if w.distance > $maxd$ then
    $maxd := w$.distance
for $d := 0$ to $maxd$
  for all ordered combinations of fragments at distance d
    construct edge z
    if $d \neq 0$ then
      $z := z \oplus last$
    if Hash(EvenBits(z)) = OddBits(z) then
      insert edge (z,EvenBits(z),d) into G
      $last$:= EvenBits(z);
remove any edge (x,y,d) with $d \neq$ distance from x to v in G
extract path (R_i..R_j) by enumerating acyclic paths in G

Figure 8: Compressed edge fragment sampling algorithm.

can be reconstructed with probability $1 - \frac{1}{c}$ is:

$$\frac{k \cdot ln(kdc)}{p(1-p)^{d-1}}$$

packets. To completely reconstruct the previous path with 95% certainty should require no more than 2150 packets. Many denial-of-service attacks send this many packets in a few seconds.

Finally, we explore the robustness of this algorithm with respect to multiple attackers. For a random hash of length h, the probability of accepting an arbitrarily constructed candidate edge-id is $\frac{1}{2^h}$. In the event that there are m attackers, then at any particular distance d, in the worst case there may be up to m distinct routers.[4] Consequently the probability that any edge-id at distance d is accepted incorrectly is at most:

$$1 - (1 - \frac{1}{2^h})^{m^k}$$

[4]In practice, the number of distinct routers is likely to be smaller for the portion of the path closest to the receiver, since many attackers will still share significant portions of their attack path with one another.

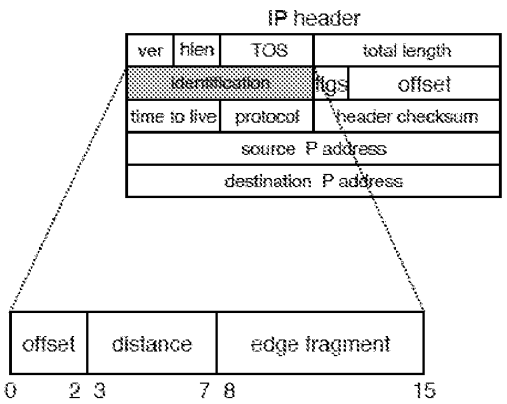


Figure 9: Encoding edge fragments into the IP identification field.

since there are m^k possible combinations of fragments in the worst case. For $h = 32$ and $k = 4$ this means that 100 distinct routers at the same distance (i.e. disjoint attack paths) will be resolved with no errors with a probability of better than 97%. For $h = 32$ and $k = 8$, (the values we use for our implementation) the same certainty can only be provided for 10 distinct routers at the same distance. Our use of the XOR function further complicates reconstruction since all combinations of XOR values must be tried as attack paths diverge. This is somewhat mitigated as the probability of propagating an error from a single edge all the way to the attacker is is extremely small because the resulting edge-id, when XORed with the previous edge-id, must again produce a correct hash.

The most significant drawback to this scheme is the large number of combinations that must be considered as the multiple attack paths diverge. While these combinations can be computed off-line, for large values of k and m even this can become intractable. For example, even with $k = 8$ and $m = 10$, if the separate attack paths diverge such that there are 10 completely independent edges per attacker, this will require roughly a billion combinations to be considered. Consequently, there is a design tension in the size of k – per-packet space overhead is reduced by a larger k, while computational overhead and robustness benefits from a smaller k.

5.2 IP header encoding

To allow for practical deployment requires that we "overload" existing header fields in a manner that will have minimal impact on existing users. This is a difficult task, especially given that even after prodigious effort we require 16 bits of space. Nonetheless, we believe it possible to obtain this space by overloading the 16-bit IP identification field. This field is currently used to differentiate IP fragments that belong to different packets. We describe our proposed encoding below, and then discuss the issues of backwards-compatibility that it raises. However, we note that because the issue of backwards-compatible encoding is largely separate from our traceback algorithms, we could adopt any reasonable encoding that comes to light.

Figure 9 depicts our choice for partitioning the identification field: 3 offset bits to represent 8 possible fragments, 5 bits to represent

the distance, and 8 bits for the edge fragment. We use a 32-bit hash, which doubles the size of each router address to 64 bits. This implies that 8 separate fragments are needed to represent each edge – each fragment indicated by a unique offset value. Finally, 5 bits is sufficient to represent 31 hops, which is more than almost all Internet paths [7, 42, 17].[5]

The observant reader will note that this layout is chosen to allow the highest performance software implementation of our algorithm, which already had a low per-packet router overhead. In the common case, the only modification to the packet is to increment its distance field. Because of its alignment within the packet, this increment precisely offsets the required decrement of the time-to-live field implemented by each router [1]. Consequently, the header checksum *does not need to be altered at all* and the header manipulation overhead could be even lower than in current software-based routers – simply an addition to the distance field, a decrement to the TTL field, and a comparison to check if either has overflowed. In the worst case, our algorithm must read the IP identification field, lookup an edge fragment and XOR it, and fold the write-back into the existing checksum update procedure (a few ALU operations). Of course, for modern ASIC-based routers these optimizations are unnecessary.

As we reuse of the IP identification field, we must address issues of backwards-compatibility for IP fragment traffic. Ultimately, there is no perfect solution to this problem and we are forced to make compromises that disadvantage fragmented traffic. Fortunately, recent measurements suggest that less than 0.25% of packets are fragmented [40, 10]. Moreover, it has long been understood that network-layer fragmentation is detrimental to end-to-end performance [27] so modern network stacks implement automatic MTU discovery to prevent fragmentation regardless of the underlying media [30]. Consequently, we believe that our encoding will interoperate seamlessly with existing protocol implementations in the vast majority of cases.

However, there is a small but real fraction of legitimate traffic that is fragmented, and we wish to ensure that it is not affected by our modifications to the extent that this is possible. Normally if a packet is fragmented, its identification field is copied to each fragment so the receiver can faithfully reassemble the fragments into the original packet. Our marking procedure can violate this property in one of two ways: by writing different values into the identification fields of fragments from the same datagram or by writing the same values into the identification fields of fragments from different datagrams. These two problems present different challenges and have different solutions.

First, a datagram may be fragmented *upstream* from a marking router. If the fragment is subsequently marked and future fragments from the same datagram are not marked consistently then reassembly may fail or data may be corrupted. While the simplest solution to this problem is to simply not mark fragments, an adversary would quickly learn to evade traceback by exploiting this limitation. In fact, some current denial-of-service attacks already use IP fragments to exploit errors in host IP reassembly functions [12]. Instead, we propose an alternative marking mechanism for fragments. We use a separate marking probability, q, for fragments. When we decide to mark a fragment, we prepend a new ICMP "echo reply" header, along with the *full* edge data – truncating the tail of the packet. This ICMP packet is considered "marked" and its distance field is set to zero, thereby guaranteeing that the distance field reflects the number of edges traversed on the way to the victim. The packet is consequently "lost" from the standpoint of the receiver, but the edge information is delivered in a way that does not impact legacy hosts. Because we can use the full edge sampling algorithm, q can be more than an order of magnitude smaller than p and yet achieve the same convergence time. This solution increases the loss rate of fragmented flows somewhat (more substantially for longer paths) but preserves the integrity of the data in these flows.

A more insidious problem is presented by fragmentation that occurs *downstream* from a marking router. If a marked packet is fragmented, but one of the fragments is lost, then the remaining fragments may linger in the victim's reassembly buffer for an extended period [5]. Future packets marked by the same router can have the same IP identification value and consequently may be incorrectly reassembled with the previous fragments. One possibility is to leave this problem to be dealt with by higher layer checksums. However, not all higher layer protocols employ checksums, and in any case it is dangerous to rely on such checksums because they are typically designed only for low residual error rates. Another solution is to set the *Don't Fragment* flag on every marked packet. Along rare paths that require fragmentation, this solution will degrade communication between hosts not using MTU path discovery, and may filter marked packets if a reduced MTU edge is close to the victim, but it will never lead to data corruption.

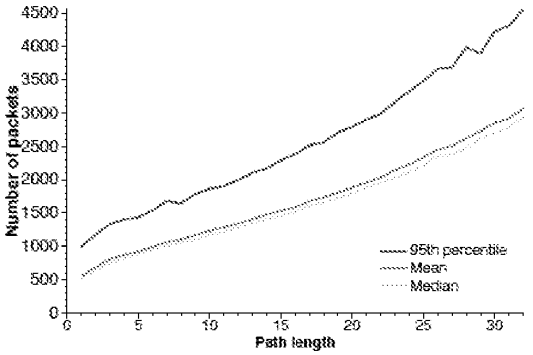


Figure 10: Experimental results for number of packets needed to reconstruct paths of varying lengths. The marking probability, p, is set to $\frac{1}{25}$. Each path length result represents the results of 1,000 independent simulation runs.

5.3 Experience

We have implemented the marking and reconstruction portions of our algorithm and have tested it using a simulator that creates random paths and originates attacks. In Figure 10 we graph the mean, median and 95th percentile for the number of packets required to reconstruct paths of varying lengths over 1,000 random test runs for each length value. We assume a marking probability of $\frac{1}{25}$. Note that while the convergence time is theoretically exponential in the path length, all three lines appear linear due to the finite path length and appropriate choice of marking probability.

We see that most paths can be resolved with between one and two

[5] It is also reasonable to turn off marking on any routers that cannot be directly connected to an attacking host (e.g. core routers). This both reduces the convergence time, and increases the "reach" of the distance field.

thousand packets, and even the longest paths can be resolved with a very high likelihood within four thousand packets. To put these numbers in context, most flooding-style denial of service attacks send many hundreds or thousands of packets each second. The analytic bounds we described earlier are conservative, but in our experience they are no more than 30% higher than our experimental results.

6. LIMITATIONS AND FUTURE WORK

There are still a number of limitations and loose ends in our approach. We discuss the most important of these here:

- backwards compatibility,
- distributed attacks,
- path validation, and
- approaches for determining the attack origin.

6.1 Backwards compatibility

The IP header encoding as we have described it has several practical limitations. It negatively impacts users that require fragmented IP datagrams and is currently incompatible with parts of IPsec [28] (the Authentication Header provides cryptographic protection for the identification field and therefore the field cannot be safely modified by routers). These problems are hardly unique to traceback and are inherent limitations that come about from attempting to coexist with or co-opt protocol features that did not anticipate a new use. One way to address this issue, originally proposed by John Hawkinson, is to selectively enable traceback support in response to operational needs. A "request for traceback" from a particular network could be encoded as a BGP attribute in the network's route advertisement. Routers receiving such an advertisement would enable traceback support on packets destined for that network. Since a network requesting such support is presumably already suffering under an attack, any minor service degradation for fragmented or authenticated flows would be acceptable.

Finally, our scheme does not address implementation in IPv6, the proposed successor to IPv4, which does not have an identification field [18]. While we do not attempt to propose a complete encoding here, we believe that the same techniques we have proposed could also be employed within IPv6, perhaps by overloading the 24-bit *flow label* field (without any further modifications this would result in roughly a factor of three increase in the number of packets required to reconstruct a path).

6.2 Distributed attacks

For large distributed attacks, the practical implementation we have described has serious limitations due to the difficulty in correctly grouping fragments together. Consequently, the probability of misattributing an edge, as well as the amount of state needed to evaluate this decision, increases very quickly with the fan-out of an attack. There is significant future work in designing alternative encoding methods that scale their robustness as they receive more data.

6.3 Path validation

Some number of the packets sent by the attacker are unmarked by intervening routers. The victim cannot differentiate between these packets and genuine marked packets. Therefore an attacker could insert "fake" edges by carefully manipulating the identification fields in the packets it sends. While the distance field prevents an attacker from spoofing edges between it and the victim – what we call the *valid suffix* – nothing prevents the attacker from spoofing extra edges past the end of the true attack path.

There are several ways to identify the valid suffix within a path generated by the reconstruction procedure. With minimal knowledge of Internet topology one can differentiate between routers that belong to transit networks (e.g. ISPs) and those which belong to stub networks (e.g. enterprise networks). Generally speaking, a valid path will never enter a stub network and then continue into a transit network. Moreover, simple testing tools such as `traceroute` should enable a victim to determine if two networks do, in fact, connect. More advanced network maps [8, 23] can resolve this issue in an increasing number of cases.

A more general mechanism is to provide each router with a "secret" that is sent along with each marked packet (perhaps in the single unallocated bit in the IP flags field). When the victim wants to validate a router in the path, it contacts the associated network (possibly out of band, via telephone or e-mail) and obtains the secret used by the router at the time of the attack. To guard against replay, the secret can be time-varying and hashed with the packet contents. Since the attacker will not know the router's secret, they will not be able to include the proper bit in their forged edge-id fragments. By eliminating edge-ids for which the secret in their constituent fragments can not be validated, we can prune a candidate attack path to only include the valid suffix.

6.4 Attack origin detection

While our IP-level traceback algorithm could be an important part of the solution for stopping denial-of-service attacks, it is by no means a complete solution. Our algorithm attempts to determine the approximate origin of attack traffic – in particular, the earliest traceback-capable router involved in forwarding attack traffic from the source that directly generated it. As mentioned earlier, there are a number of reasons why this may differ from the true source of the attack: attackers can hide their true identities by "laundering" attacks through third parties, either indirectly (e.g. smurf attacks [13] or DNS reflectors [15]) or directly via compromised "stepping stone" machines or IP-in-IP tunnels. While there is ongoing work on following attackers through intermediate hosts [45, 39], there are still significant challenges in developing a generally applicable and universally deployable solution to this problem. One interesting possibility enabled by the packet marking approach is to extend traceback across "laundering points". For example, identifying marks could be copied from a DNS request packet into the associated DNS reply, thereby allowing the victim to trace the full causal path.

Even in absence of such "laundering", our approach does not reveal the actual host originating the attack. Moreover, since hosts can forge both their IP source address and MAC address the origin of a packet may never be explicitly visible. On shared media such as FDDI rings, this problem can only be solved by explicit testing. However, on point-to-point media, the input port a packet arrives on is frequently enough to determine its true origin. On other media, there may be a MAC address, cell number, channel, or other hint that would help to locate the attack origin. In principle, our algorithm could be modified to report this information by occasionally marking packets with a special edge-id representing a link between the router and the input port on which the packet arrived (or other

"hint" information). We have not explored the design of such a feature in any depth.

Finally, traceback is only effective at finding the source of *attack traffic*, not necessarily the *attacker* themselves. Stopping an attack may be sufficient to eliminate an immediate problem, but long term disincentives may require a legal remedy and therefore the forensic means to determine an attacker's identity. Even with perfect traceback support, unambiguously identifying a sufficiently skilled and paranoid attacker is likely to require cooperation from law enforcement and telecommunications organizations.

7. CONCLUSION

In this paper we have argued that denial-of-service attacks motivate the development of improved traceback capabilities and we have explored traceback algorithms based on packet marking in the network. We have shown that this class of algorithm, best embodied in *edge sampling*, can enable efficient and robust multi-party traceback that can be incrementally deployed and efficiently implemented. As well, we have developed variant algorithms that sacrifice convergence time and robustness for reduced per-packet space requirements. Finally, we have suggested one potential deployment strategy using such an algorithm based on overloading existing IP header fields and we have demonstrated that this implementation is capable of fully tracing an attack after having received only a few thousand packets. We believe our solution represents a valuable first step towards an automated network-wide traceback facility. Several areas remain to be addressed in future work, such as the combination of widely distributed attacks and points of indirection such as reflectors.

Acknowledgments

This paper has benefited from conversations with many different people – far more than can be acknowledged completely here. Still we would like to particularly thank Bill Cheswick and Hal Burch for early access to their work in this area, Sean McCreary and KC Claffy for access to their packet trace data, Alec Wolman for help with jgraph, Przemek Pardyak, Mike Swift, Neil Spring, Gretta Bartels, Robert Grimm and Geoff Voelker for commenting on early drafts of the paper, and constructive feedback from John Hawkinson, Robert Stone, Jeff Mogul, Robert Moskowitz, Greg Minshall, Tony Li, Curtis Villamizar, Steve Corbato, and countless others. Finally, we thank both the anonymous reviewers and our shepherds, Vern Paxson and Craig Partridge, for their efforts.

8. REFERENCES

[1] F. Baker. Requirements for IP Version 4 Routers. RFC 1812, June 1995.

[2] G. Banga, P. Druschel, and J. Mogul. Resource Containers: A New Facility for Resource Management in Server Systems. In *Proceedings of the 1999 USENIX/ACM Symposium on Operating System Design and Implementation*, pages 45–58, Feb. 1999.

[3] S. M. Bellovin. Security Problems in the TCP/IP Protocol Suite. *ACM Computer Communications Review*, 19(2):32–48, Apr. 1989.

[4] S. M. Bellovin. ICMP Traceback Messages. Internet Draft: draft-bellovin-itrace-00.txt, Mar. 2000.

[5] R. Braden. Requirements for Internet Hosts – Communication Layers. RFC 1122, Oct. 1989.

[6] H. Burch and B. Cheswick. Tracing Anonymous Packets to Their Approximate Source. Unpublished paper, Dec. 1999.

[7] R. L. Carter and M. E. Crovella. Dynamic Server Selection Using Dynamic Path Characterization in Wide-Area Networks. In *Proceedings of the 1997 IEEE INFOCOM Conference*, Kobe, Japan, Apr. 1997.

[8] B. Cheswick and H. Burch. Internet Mapping Project. `http://cm.bell-labs.com/who/ches/map/index.html`, 2000.

[9] Cisco Systems. Configuring TCP Intercept (Prevent Denial-of-Service Attacks). Cisco IOS Documentation, Dec. 1997.

[10] K. Claffy and S. McCreary. Sampled Measurements from June 1999 to December 1999 at the AMES Inter-exchange Point. Personal Communication, Jan. 2000.

[11] Computer Emergency Response Team. CERT Advisory CA-96.26 Denial-of-Service Attack via pings. `http://www.cert.org/advisories/CA-96.26.ping.html`, Dec. 1996.

[12] Computer Emergency Response Team. CERT Advisory CA-97.28 IP Denial-of-Service Attacks. `http://www.cert.org/advisories/CA-97.28.smurf.html`, Dec. 1997.

[13] Computer Emergency Response Team. CERT Advisory CA-98.01 "smurf" IP Denial-of-Service Attacks. `http://www.cert.org/advisories/CA-98.01.smurf.html`, Jan. 1998.

[14] Computer Emergency Response Team. CERT Advisory CA-2000-01 Denial-of-Service Developments. `http://www.cert.org/advisories/CA-2000-01.html`, Jan. 2000.

[15] Computer Emergency Response Team. CERT Incident Note IN-2000-04 Denial-of-Service Attacks using Nameservers. `http://www.cert.org/incident_notes/IN-200-04.html`, Apr. 2000.

[16] Computer Security Institute and Federal Bureau of Investigation. 1999 CSI/FBI Computer Crime and Security Survey. Computer Security Institute publication, Mar. 1999.

[17] Cooperative Associationfor Internet Data Analysis. Skitter Analysis. `http://www.caida.org/Tools/Skitter/Summary/`, 2000.

[18] S. Deering. Internet protocol, version 6 (ipv6). RFC 2460, Dec. 1998.

[19] W. Feller. *An Introduction to Probability Theory and Its Applications (2nd edition)*, volume 1. Wiley and Sons, 1966.

[20] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing. RFC 2267, Jan. 1998.

[21] J. Glave. Smurfing Cripples ISPs. Wired Technolgy News: (`http://www.wired.com/news/news/technology/story/9506.html`), Jan. 1998.

[22] I. Goldberg and A. Shostack. Freedom Network 1.0 Architecture and Protocols. Zero-Knowledge Systems White Paper, Nov. 1999.

[23] R. Govindan and H. Tangmunarunkit. Heuristics for Internet Map Discovery. In *Proceedings of the 2000 IEEE INFOCOM Conference*, Tel Aviv, Israel, Mar. 2000.

[24] L. T. Heberlein and M. Bishop. Attack Class: Address Spoofing. In *1996 National Information Systems Security Conference*, pages 371–378, Baltimore, MD, Oct. 1996.

[25] J. D. Howard. *An Analysis of Security Incidents on the Internet*. PhD thesis, Carnegie Mellon University, Aug. 1998.

[26] P. Karn and W. Simpson. Photuris: Session-Key Management Protocol. RFC 2522, Mar. 1999.

[27] C. Kent and J. Mogul. Fragmentation Considered Harmful. In *Proceedings of the 1987 ACM SIGCOMM Conference*, pages 390–401, Stowe, VT, Aug. 1987.

[28] S. Kent and R. Atkinson. Security architecture for the internet protocol. RFC 2401, Nov. 1998.

[29] C. Meadows. A Formal Framework and Evaluation Method for Network Denial of Service. In *Proceedings of the 1999 IEEE Computer Security Foundations Workshop*, Mordano, Italy, June 1999.

[30] J. Mogul and S. Deering. Path MTU Discovery. RFC 1191, Nov. 1990.

[31] R. T. Morris. A Weakness in the 4.2BSD Unix TCP/IP Software. Technical Report Computer Science #117, AT&T Bell Labs, Feb. 1985.

[32] V. Paxson. End-to-End Routing Behavior in the Internet. *IEEE/ACM Transactions on Networking*, 5(5):601–615, Oct. 1997.

[33] C. Perkins. IP Mobility Support. RFC 2002, Oct. 1996.

[34] J. Postel. Internet Protocol. RFC 791, Sept. 1981.

[35] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous Connections and Onion Routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, May 1998.

[36] E. C. Rosen, Y. Rekhter, D. Tappan, D. Farinacci, G. Fedorkow, T. Li, and A. Conta. MPLS Label Stack Encoding. Internet Draft: draft-ietf-mpls-label-encaps-07.txt (expires March 2000), Sept. 1998.

[37] G. Sager. Security Fun with OCxmon and cflowd. Presentation at the Internet 2 Working Group, Nov. 1998.

[38] O. Spatscheck and L. Peterson. Defending Against Denial of Service Attacks in Scout. In *Proceedings of the 1999 USENIX/ACM Symposium on Operating System Design and Implementation*, pages 59–72, Feb. 1999.

[39] S. Staniford-Chen and L. T. Heberlein. Holding Intruders Accountable on the Internet. In *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, pages 39–49, Oakland, CA, May 1995.

[40] I. Stoica and H. Zhang. Providing Guaranteed Services Without Per Flow Management. In *Proceedings of the 1999 ACM SIGCOMM Conference*, pages 81–94, Boston, MA, Aug. 1999.

[41] R. Stone. CenterTrack: An IP Overlay Network for Tracking DoS Floods. In *to appear in Proceedings of thje 2000 USENIX Security Symposium*, Denver, CO, July 2000.

[42] W. Theilmann and K. Rothermel. Dynamic Distance Maps of the Internet. In *Proceedings of the 2000 IEEE INFOCOM Conference*, Tel Aviv, Israel, Mar. 2000.

[43] C. Villamizar. Personal Communication, Feb. 2000.

[44] M. Vivo, E. Carrasco, G. Isern, and G. O. Vivo. A review of port scanning techniques. *ACM Computer Communications Review*, 29(2):41–48, Apr. 1999.

[45] Y. Zhang and V. Paxson. Stepping Stone Detection. In *to appear in Proceedings of thje 2000 USENIX Security Symposium*, Denver, CO, July 2000.

Declaration of Dr. Stefan Savage

ATTACHMENT B
(EXHIBIT 1016)

Stefan Savage

Department of Computer Science and Engineering
University of California, San Diego
9500 Gilman Drive, MC 0404
La Jolla, CA 92093-0404

Email: savage@cs.ucsd.edu
Phone: (858) 822-4895
Fax: (858) 534-7029
URL: http://www.cs.ucsd.edu/~savage/

Education

Ph.D., Computer Science and Engineering — January, 2002
University of Washington, Seattle, WA.

B.S., Applied History — May 1991
Carnegie Mellon University, Pittsburgh, PA.

Research Interests

Design and analysis of experimental computer and communications systems, with a focus on security

Academic Employment

University of California at San Diego, Department of Computer Science and Engineering
- Assistant Professor, July 2000 - June 2006
- Associate Professor, July 2006 – June 2010
- Professor, July 2010 – present

University of California at San Diego, Center for Networked Systems
- Associate Director, July 2009 – July 2010
- Director, July 2010 – present

Professional Experience

ZeroFox, Baltimore, MD — 2013-2016
Technical Advisory Board

Impermium, San Francisco, CA — 2011-2013
Technical Advisory Board

Anonymizer Inc, San Diego, CA — 2010-2012
Consultant

Microsoft, Mountain View, CA — 2010
Consultant

Monitor 360, San Francisco, CA — 2010
Consultant

Netsift, San Diego, CA — 2004-2005
Co-founder & Consultant

Rendition Networks, Redmond, WA — 2001-2005
Strategy Advisory Council Member

Asta Networks, Seattle, WA — 2000-2002
Co-founder and Chief Scientist

Appliant, Seattle, WA — 2000
Consultant, Research Group

Digital Equipment Corporation, Palo Alto, CA — June 1996- September 1996
Intern, Systems Research Center

Hewlett Packard Corporation, Palo Alto, CA — July 1995
Visiting Researcher, Storage Systems Program

Honeywell Corporation, Clearwater, FL — July 1993
Consultant, Space Systems Center

Carnegie Mellon University, Pittsburgh, PA — 1991-1994

A variety of expert witness and legal support engagements

Awards and Honors

IEEE Computer Society Technical Committee on Security and Privacy Distinguished Service Award, 2019
MacArthur Fellow, 2018-2023
Irwin Mark and Joan Klein Jacobs Chair in Information and Computer Science, 2016
ACM Prize in Computing, 2016 (*previously the ACM Infosys Award*)
University of Washington, Alumni Achievement Award, 2016
ACM SIGOPS Mark Weiser Award, 2013
ACM Fellow, 2010
Alfred P. Sloan Research Fellowship, 2004-2006.

16 Award/Honored papers
- IETF **Applied Networking Research Prize**, 2021 (Trufflehunter)
- IEEE Security & Privacy **Test of Time Award**, 2020 (Security Analysis of a Modern Automobile)
- ACM Computer and Communications Security **Test of Time Award**, 2019 (Hey, you, get off of my cloud)
- Distinguished Paper, USENIX Security, 2019 (Lateral Phishing)
- USENIX Security **Test of Time Award**, 2017 (Backscatter)
- Selected Paper, ACM ASPLOS, 2011 (forwarded to ACM TOCS) (LogEnhancer)
- Jay Lepreau Best Paper Award, USENIX OSDI, 2008 (invited to CACM) (Difference Engine)
- Selected Paper, ACM Computer and Communications Security, 2008 (invited to CACM) (Spamalytics)
- Selected Paper, ACM Computer and Communications Security, 2008 (forwarded to ACM TISSEC) (RoC)
- IEEE William C. Carter Award Paper, IEEE DSN, 2005 (forwarded to IEEE TDSC) (Fatih)
- Selected Paper, International Conference on Network Protocols, 2001 (to IEEE/ACM ToN) (ECN nonce)
- Best Paper, USENIX Security Symposium, 2001 (Backscatter)
- Selected Paper, ACM SIGCOMM, 2001 (forwarded to IEEE/ACM TON) (Traceback)
- Best Student Paper, USENIX Symposium on Internet Technologies and Systems, 1999 (Sting)
- Selected Paper, IEEE Micro, 1999 (forwarded from 1998 IEEE Hot Interconnects) (Detour)
- Best Paper, ACM Symposium on Operating Systems Principles, 1997 (forwarded to ACM TOCS) (Eraser)
- Best Student Paper, USENIX Technical Conference, 1996 (AFRAID)

Network World Magazine's "Top 10 Wicked Cool Algorithms for 2008" (for XL), 2008.
William Chan Dissertation Award, University of Washington, 2002.
Jacobs Faculty Fellowship, 2001-2003.

Past/Current Research Funding

Google Faculty Research Award, 2020, PI (co-PI Aaron Schulman), "**Passive Network Analysis for Security Analysis and Assessment**" $70,750.

Facebook, 2018, PI (co-PI Geoff Voelker), "**Securing the Internet**" Award, $75,000.

Samsung GRO, 2017-2018, co-PI (PI Tajana Rosing, co-PI's Gary Cottrell and Kirill Levchenko), "**Smart and Safe Home**", $300,000.

Department of Homeland Security, 2018-2012, co-PI (Kirill Levchenko PI, Geoff Voelker, Alex Snoeren and Vern Paxson co-PIs), "**Foundations of Threat Intelligence Metrics**", $1,045,000.

Department of Homeland Security (via a subcontract from PNNL), 2017-2018, "**Security Evaluation of the Boeing 757 Environmental Systems (ACI)**", co-PI (PI Kirill Levchenko, Co-PI Alex Snoeren), $350,000.

National Science Foundation, 2017-2020, PI (co-PI's Geoff Voelker and Kirill Levchenko), "**Large-scale Characterization of DNS Abuse**", $1,200,000.

Hewlett Foundation, 2016-20187, PI (co-PI's Alex Snoeren and Kirill Levcheko), "**Quantifying the Legal Risks on Third-party Vulnerability Research**", $100,000.

Hewlett Foundation, 2016-2017, PI (co-PI's Geoff Voelker and Kirill Levchenko), "**Reasoning about Zero-Day Vulnerabilities**", $100,000.

National Science Foundation, 2016-2019, co-investigator (PI Kirill Levchenko, co-PIs Ranjit Jhala and Alex Snoeren), "**Foundations of Security Cyber-Physical Systems of Systems**", $560,000.

National Science Foundation, 2016-2019, co-investigator (PI Alex Snoeren, co-PIs Kirill Levchenko, George Porter and Geoff Voelker), "**II-New: A Dual-Purpose Data Analytics Laboratory**", $548,000

Google Research Gift, 2016, $100,000 (PI)

Google Research Gift, 2014, $250,000 (co-investigator w/Geoff Voelker and Kirill Levchenko)

Facebook gift, 2014, $50,000 (PI)

Google Research Gift, 2012, $200,000 (co-investigator w/Geoff Voelker and Kirill Levchenko)

National Science Foundation, 2012-2017, PI (co-PIs Geoff Voelker, James Fowler, Alex Snoeren, Lawrence Saul, Kirill Levchenko, Erin Kenneally, Vern Paxson, Mark Allman, Chris Grier, Chris Hoofnagle, Dan Klein, Christian Kreibich, Deirdre Mulligan, Nicholas Weaver and Damon McCoy), "**Beyond Technical Security: Developing an Empirical Basis for Socio-Economic Perspectives**", $10,000,000 (4.7M to UCSD, 4.7M to ICSI, 06M to GMU)

Microsoft Research Gift, 2012, $50,000 (co-investigator w/Geoff Voelker)

National Science Foundation, 2011-2014, Co-Investigator (PI Alex Snoeren, co-PI Kirill Levchenko), "**Understanding Network Failure**", $450,000.

Yahoo Research Gift, 2011, $10,000 (co-investigator w/Geoff Voelker and Kirill Levchenko)

Google Research Gift, 2010, $400,000 (co-investigator w/Geoff Voelker and Kirill Levchenko)

Cisco Research Gift, 2010, $90,000 (co-investigator w/Geoff Voelker and Kirill Levchenko)

National Science Foundation, 2010-2012, Principal Investigator (Co-PIs Yoshi Kohno and Ingolf Kreuger), "**TC:Medium:Collaborative Research:Foundations, Architectures, and Methodologies for Secure and Private Cyber-physical Vehicles**", $1,200,000 ($600k to UCSD, $600k to UW).

Google Research Gift (DC Office), 2010, $5,000.

Google Research Gift, 2010, Co-Investigator (PI Kirill Levchenko), "**Understanding the Economics of CAPTCHAs**", $50,000.

UCSD Center for Network Systems, 2009-2011, Co- Investigator (PI Hovav Shacham), "**Future is Cloudy: Security and Availability Issues in Outsourced Computing**", $104,000.

UCSD Center for Network Systems, 2009-2011, Co- Investigator (PI Alex Snoeren), "**Understanding Failure Patterns in Real Networks**", $104,000.

Yahoo! Research Gift, 2009, $50,000.

National Science Foundation, 2009-2012, Co-Investigator (PI Mohan Paturi, Co-PIs Alex Snoeren, Mathew McCubbins, Joel Sobel), "**NetSE:Medium:Network Structure, Incentives, and Outcomes**", $900,000.

Office of Naval Research, Multi-University Research Initiative, 2009-2014, Principal Investigator (CoPIs Geoff Voelker, Vern Paxson, Dawn Song and Dan Klein) "**Infiltration of Botnet Command & Control and Support Ecosystems**", $7,500,000 (~3.8M to UCSD, ~3.7M to Berkeley)

Cisco gift, 2009, Principal Investigator (Co-PI Geoff Voelker), $37,500.

Google Research Gift, 2008, Principal Investigator (Co-PI Geoff Voelker), "**Scam Analysis and Defense via Botnet Infiltration**", $70,000.

Yahoo! Research Gift, 2008, $80,000.

Microsoft Research Gift, 2008, $12,500.

National Science Foundation, 2008-2010. Principal Investigator (Co-PI Geoff Voelker), "**Understanding and Exploiting Economic Incentives in Internet-based Scams**", CNS-0831138, $500,000.

UCSD Center for Network Systems, Principal Investigator (Co-PI Geoff Voelker), "**Bluesky: System Support for Transparent Cloud Computing**", $129,500.

UCSD Center for Network Systems, 2008-2010, Principal Investigator (Co-PI Geoff Voelker), "**Scam Analysis and Defense via Botnet Infiltration**", $129,500.

UC Discovery Grant, 2007. Principal Investigator (Co-PIs Geoff Voelker and Alex Snoeren), "**System Support for Derived Data Management**", $94,602.

UC Discovery Grant, 2007: Co-Investigator (Geoff Voelker PI, Amin Vahdat and Alex Snoeren co-PIs), "**Extending Virtual Cluster Management and Resource Utilization**", $94,602

Cisco Systems Gift, 2007. Principal Investigator (Co-PIs Geoff Voelker) , "**Data-based Security Policies and Miscreant Infrastructure Analysis**", $200,000.

Google Gift, 2007. Principal Investigator (Co-PIs Geoff Voelker and Alex Snoeren), "**Automated Cross-layer Diagnosis of 802.11 Wireless Networks**", $65,000.

National Science Foundation, 2007-2009. Co-Investigator (Alex Snoeren, PI, Amin Vahdat and Yoshi Kohno Co-PIs), "**Privacy Preserving Attribution and Provenance**", CNS-0722031, $636,000.

Microsoft Research Gift, 2007. $25,000.

CNA Insurance Gift, 2007. $50,000.

National Science Foundation, 2006-2007. Co-Investigator (Alex Snoeren, PI, Amin Vahdat Co-PI), "**Enabling Defense and Deterrence through Private Attribution**", CNS-0627157, $400,000.

UCSD Center for Network Systems, 2006-2008, Principal Investigator (Co-PIs Geoff Voelker and Alex Snoeren), "**Automated Cross-layer Diagnosis of Enterprise Wireless Networks**", $110,000.

Microsoft Research Gift, 2006, $25,000.

UCSD Center for Network Systems, 2005-2007, Co-Investigator (Amin Vahdat PI), "**Agile Virtual Clusters**", $140,000.

National Science Foundation, 2004-2009. Principal Investigator (co-PIs Vern Paxson, Geoffrey Voelker, George Varghese and Nicholas Weaver), Cybertrust Center Grant, "**Center for Internet Epidemiology and Defenses**", CNS-0433668 and CNS-0433702, $6,200,000 ($3.1M to UCSD, $3.1M to ICSI).

UCSD Center for Network Systems, 2004-2006, Principal Investigator (Co-PIs Kimberly Claffy, Alex Snoeren, Amin Vahdat, George Varghese and Geoff Voelker), "**Internet Epidemiology**", $240,000.

UCSD Center for Network Systems, 2004-2006, Co-Investigator (Geoff Voelker PI), "**Cross-layer Wireless Measurement**", $160,000.

Microsoft Research Gift, 2004, $50,000.

Sloan Research Fellowship, 2004-2005**,** $40,000**.**

National Science Foundation, 2003-2006. Co-Investigator (Andrew Chien PI, David Kriegman and Joseph Pasquale Co-PIs), Infrastructure Grant, "**FWGrid: A Research Infrastructure with Fast Wireless, Wired, Compute and Data Infrastructure for Next Generation Systems and Applications**", EIA-0303622, $1,800,000.

National Science Foundation, 2003-2006. Co-Investigator (David Moore PI and Geoffrey Voelker Co-PI). "**Quantitative Network Security Analysis**", CNS-0311690**,** $372,183.

AT&T Network Engineering Center Gift, 2003 (managed w/George Varghese)**,** $500,000.

National Institute of Standards, 2001-2004. Co-Investigator (George Varghese PI), 60NANB1D0118, "**Real-Time Intrusion Detection**", $909,951.

Defense Advanced Research Projects Agency, 2001-2004. Principal Investigator (Co-PIs Geoffrey M. Voelker and Keith Marzullo), N66001-01-1-8933, "**Reliable Adaptive Multi-Path Routing (RAMP)**", $2,293,239.

Student Advising

Ph.D. Students Graduated (17):

- Ranjita Bhagwan, "**Automated Availability Management**", (Ph.D. 2004, co-advised with Geoff Voelker -> IBM Research -> Microsoft Research),
- John Bellardo, "**Secure and Efficient Wireless Networks**", (Ph.D. 2006 -> California Polytechnic State University),
- Yu-Chung Cheng, "**Automating Cross-Layer Diagnosis of Enterprise 802.11 Wireless Networks**" (Ph.D. 2007, co-advised with Geoff Voelker -> Google),
- Alper Mizrak, "**Detecting Malicious Routers**" (Ph.D. 2007, co-advised with Keith Marzullo -> VMWare),
- Kirill Levchenko, "**XL: A Communication-Efficient Routing Protocol**", (Ph.D. 2008, co-advised with Mohan Paturi -> Research Scientist at UCSD),
- Justin Ma, "**Learning to Detect Malicious URLs**", (Ph.D. 2010, co-advised with Geoff Voelker and Lawrence Saul -> Postdoc at UC Berkeley -> Google),
- Michael Vrable, "**Migrating Enterprise Storage Applications to the Cloud**", (Ph.D. 2011, co-advised with Geoff Veolker -> Google),
- Marti Motoyama, "**Understanding the Role of Outsourced Labor in Web Service Abuse**", (Ph.D. 2011, co-advised with Geoff Voelker and George Varghese -> FitBit),
- Chris Kanich , "**Characterizing Internet Scams Through Underground Infiltration**", (Ph.D. 2012, co-advised with Geoff Voelker -> University of Illinois, Chicago), Andreas Pittsilidis, "**Spam Value Chain: Defensive Intervention Analysis**", (Ph.D. 2013, co-advised with Geoff Voelker -> Google),
- Daniel Turner, "**Opportunistically Reconstructing a Network's Failure History**", (Ph.D. 2013, co-advised with Alex Snoeren -> TurboSquid),
- Sarah Meiklejohn, "**Flexible Models for Secure Systems**", (Ph.D. 2014, co-advised with Mihir Bellare -> University College London),
- David Wang, "**A Comprehensive Approach to Undermining Search Result Poisoning**", (Ph.D. 2014, co-advised with Geoff Voelker -> Google),
- He "Lonnie" Liu, "**A High Performance Hybrid ToR for Data Centers**", (Ph.D. 2015, co-advised with Geoff Voelker -> Google),
- Tristan Halvorson, "**Registration Intent in the Domain Name Market**", (Ph.D. 2015, co-advised with Geoff Voelker -> Google),
- Matt Der, "**Investigating Large-Sale Internet Abuse Through Web Page Classification**", (Ph.D. 2015, co-advised with Lawrence Saul and Geoff Voelker -> notch.io),
- Neha Chachra, "**Understanding URL Abuse for Profit**" (Ph.D. 2016, co-advised with Geoff Voelker -> Facebook),
- Wilson Lian, "**JIT Spraying Threats on ARM and Defense by Diversification**" (Ph.D. 2016, co-advised with Hovav Shacham -> Google),
- Louis Dekoven, "**Addresing Device Compromise from the Perspective of Large Organizations**" (Ph.D. 2019, co-advised with Geoff Voelker -> Facebook)

- Guo "Vector" Li, "**An Empirical Analysis on Threat Intelligence: Data Characteristic and Real-World Uses**" (Ph.D. 2020, co-advised with Kirill Levchenko -> Google)

Current Ph.D. Thesis Advisor (5): Ariana Miran, Gautam Akiwate and Alex Liu (w/Geoff Voelker) and Audrey Randall (w/Geoff Voelker and Aaron Schulman), Evan Johnson (co-advised with Deian Stefan)

M.S. Students Graduated (19):

- Doug Brown (2003 -> NYU Law School, co-advised w/Geoff Voelker),
- Ryan Sit (2004 -> Novatel, co-advised with William Griswold),
- Ishwar Ramani (2005 -> Juniper Networks),
- Chris Tuttle (2006 -> Google, co-advised with Alex Snoeren),
- Varun Almaula (2008 -> Cisco Systems),
- Ryan Roemer (2009 -> Microsoft, co-advised with Hovav Shacham),
- Sebastian Becerra (2009 -> Peace Corps),
- Grace Wang (2010 -> Qualcomm, co-advised with Hovav Shacham),
- Kourosh Derakshan (2011 -> Qualcomm),
- Sushma Bannur (2011 -> Microsoft, co-advised with Lawrence Saul),
- Devin Lundberg (2014 -> Pintrest, co-advised with Kirill Levchenko),
- Chris Gardner (2014 -> Arris),
- Malathi Raghavan (2014 -> Google, co-advised with Geoff Voelker),
- Gauytam Akiwate (2015 -> PureStorage, co-advised with Lawrence Saul and Geoff Voelker),
- Mohit Kothari (2015 -> Groupon, co-advised with Lawrence Saul and Geoff Voelker),
- Ian Foster (2015 -> Salesforce, co-advised with Kirill Levchenko),
- Louis Dekovan (2015 -> UCSD Ph.D., co-advised with Geoff Voelker),
- Ben Braun (2016 -> Facebook),
- Matthew Bohl (2016 -> Adobe)

Undergraduate independent projects (2): Derrick Wong (2004 -> Intel), Duong Harrison (2006 -> UPenn Ph.D. program), Brandon Enright (2007-9), Jason Bradshaw (2012), Tiffany Allan (2014-2015), Aaron Hurtado (2014-2015), Andrew Jabassa (2014-2015), Kelsey Ma (2014-2015), Ansel Blum (2018-2019)

Teaching Experience

CSE 123, "Communications Networks",
Fall 2005 CAPES: enrollment: 109, recommend class: 95%, recommend instructor: 86%
Winter 2005 CAPES: unranked
Fall 2009 CAPES: enrollment: 52, recommend class: 89%, recommend instructor: 97%
Fall 2011 CAPES: enrollment: 129, recommend class: 90%, recommend instructor: 84%
Spring 2015 CAPES: enrollment: 102, recommend class: 88%, recommend instructor: 92%
Fall 2016 CAPES: enrollment 103, recommend class: 89.7%, recommend instructor, 89.7%

CSE 123b, "Communications Software",
Spring 2002 CAPES: enrollment: 72, recommend class: 92%, recommend instructor: 89%
Spring 2003 CAPES: enrollment: 67, recommend class: 93%, recommend instructor: 90%
Spring 2004 CAPES: enrollment: 101, recommend class: 93%, recommend instructor: 93%

CSE 127, “Computer Security”,
Spring 2005 CAPES: enrollment: 63, recommend class: 98%, recommend instructor: 96%
Winter 2006 CAPES: enrollment: 71, recommend class: 98%, recommend instructor: 88%
Spring 2007 CAPES: enrollment: 83, recommend class: 98%, recommend instructor: 98%
Spring 2008 CAPES: enrollment: 59, recommend class: 100%, recommend instructor: 100%
Spring 2009 CAPES: enrollment: 54, recommend class: 100%, recommend instructor: 100%
Spring 2012 CAPES: enrollment: 137, recommend class: 96%, recommend instructor: 91%
Fall 2015 CAPES: enrollment: 152, recommend class: 96%, recommend instructor: 94%
Spring 2018 CAPES: enrollment: 170, recommend class: 100%, recommend instructor: 100%
Fall 2018 CAPES: enrollment 168, recommend class: 91.7%, recommend instructor: 91.7%
Spring 2019 CAPES: enrollment: 220, recommend class :96%, recommend instructor: 98%
Spring 2020 CAPES: enrollment: 206, recommend class: 100%, recommend instructor: 100%

CSE 221, “Graduate Operating Systems”,
Fall 2003: enrollment: 49, recommend class: 83%, recommend instructor: 89%

CSE 222a, “Graduate Communications Networks”,
Fall 2001: enrollment: 30, recommend class: 96%, recommend instructor: 94%
Winter 2003: enrollment: 46, recommend class: 92%, recommend instructor: 95%

CSE 227, “Graduate Computer Security”,
Winter 2008: enrollment: 21, recommend instructor: 100%
Spring 2010: enrollment: 22, recommend instructor: 100%
Winter 2012: enrollment: 28, recommend instructor: 100%
Winter 2015, Spring 2017, Fall 2019

GPAA477 (previously IRGN490) “Cyber Security” (co-taught w/Peter Cowhey IR/PS, Spring 2011, Spring 2013, Spring 2014, Spring 2016, Spring 2018, Spring 2019, Spring 2020.

CSE 291, “Internet Crime”, Fall 2008. (graduate seminar)
CSE 291, “Cyber Security, Privacy and US Law, Winter 2021 (graduate)

CSE 294, “Systems and Networking Seminar”, Winter 2003, Spring 2005, Fall 2005, Fall 2006, Spring 2007, Fall 2007.

CSE 291, “Selected Topics in Wide-Area Networking”, Winter 2001, Spring 2002.

Program Committee Service

2020 USENIX CSET

2019 ACM Symposium on Computer Science and Law

Program Co-chair, 2016 USENIX Security Symposium

2015 ACM SIGCOMM

2012 IEEEE CREDS

2012 USENIX/ACM OSDI

Program Co-chair, 2011 IEEE eCrime Symposium

2011 ACM SIGCOMM

2010 USENIX Workshop on Large-scale Exploits and Emerging Threats (LEET)

2010 USENIX/ACM Networked Systems: Design and Implementation Conference (NSDI)

2009 ACM Symposium on Operating System Principles (SOSP).

2009 IEEE Symposium on Security and Privacy (Oakland).

Program Co-chair, 2008 ACM SIGCOMM.

2008 USENIX Workshop on Organizing Workshops, Conferences and Symposia for Computer Systems (WOWCS).

Program Co-chair, 2007 USENIX/ACM Hot Networks Workshop.

2007 ACM Workshop On Recurring Malware (WORM) at the ACM Computer and Communications Security Conference.

2007 ACM Symposium on Operating System Principles (SOSP).

2007 ACM SIGCOMM Conference.

2006 IEEE Symposium on Security and Privacy (Oakland).

2005 ACM Symposium on Operating Systems Principles (SOSP).

2005 USENIX/ACM Networked Systems: Design and Implementation Conference (NSDI).

2004 ACM Workshop On Rapid Malcode (WORM)at the ACM Computer and Communications Security Conference.

Program Co-chair, 1st USENIX/ACM Networked Systems: Design and Implementation Conference (NSDI), 2004.

Program Chair, 1st Workshop On Rapid Malcode (WORM) at the ACM Computer and Communications Security Conference, 2004.

2003 ACM SIGCOMM Conference.

2003 DARPA Information Survivability Conference and Exposition.

2003 ACM Symposium on Operating System Principles (SOSP).

2002 ACM Hot Networks Workshop.

2002 ACM SIGCOMM Conference.

2001 IEEE Globecom Global Internet Symposium.

Other Professional Service Activities

Steering committee, CRA Computing Innovation Fellows, 2019-present

Steering Committee, USENIX Enigma, 2015-present

Steering Committee, ACM Conference on Computer and Communications Security (CCS), 2015-present

Member, Boeing Industry Cyber Technical Counsel, 2019-present

Distinguished Reviewer, National Security Agency Best Scientific Cybersecurity Paper Competition, 2017-present

Panelist, ISAT Study on Biocyber Security, 2020

Panelist, ACM SIGCOMM Panel on Privacy and Security, 2020

Steering Committee, ACM Computer Science and Law Symposium, 2019.

IEEE Security and Privacy, Chair Test-of-Time Award committee, 2018-2019

Carnegie Mellon University, President's Advisory Committee on the School of Computer Science, 2018.

IEEE Security and Privacy, Chair Test-of-Time Award establishment committee, 2017

Committee Member, National Academies Forum on Cyber Resilience, 2016

Computing Community Consortium, Sociotechnical Cybersecurity Workshop Steering Committee, 2016-2017

Panelist, IEEE CREDS, 2013

Panelist, International Anti-Counterfeiting Coalition (IACC) Conference, 2013

RSA Conference Panel speaker, 2013

Cryptographers Panel Speaker, RSA Conference, 2012

Panelist, RSA eFraud Global Forum, 2012

Presenter and Panelist, CCC "**The Impact of NITRD Symposium**", Washington D.C, 2012

Advisory Committee, National Science Foundation Directorate for Computer and Information Science and Engineering, 2011 (Chair, "Breakthrough Proposals" Working Group)

ACM Distinguished Dissertation Award Committee, 2011-2015 (Chair 2013)

Panelist, IEEE eCrime Symposium ethics panel, 2011

Presenter, National Academy of Science Study on Electronic Vehicle Controls and Unintended Acceleration, 2011.

Participant, ISAT Cybereconomics study, 2011.

ACM SIGCOMM Steering Committee, 2010-2011.

Member, ISAT Red Team, 2010.

Presenter, JASON Study group, "Science of Security", 2010.

Member, ISAT Black Cloud Study, 2010.

Director, Collaborative Center for Internet Epidemiology and Defenses (joint effort between UCSD and ICSI), 2004-2010.

Panel Member, ViaSAT/NSA Information Assurance Workshop on Information Assurance

Content Judge, Google NaCl Security Contest, 2009.

External Advisory Board, Stanford Programmable Open Mobile Internet (POMI) 2020 Project, 2008-

Committee Member, 3rd Santa Fe Institute Workshop on Malware, 2008.

Co-founder and Steering Committee Member, USENIX Workshop on Large-scale Exploits and Emerging Threats (LEET), 2007.

Member, GENI Science Council, 2007-2008.

Committee Member, United States Air Force Workshop on the Risks of Homogeneity, 2007.

Committee Member, 2nd Santa Fe Institute Workshop on Malware, 2007.

Committee Member, National Research Council's Computer Science and Telecommunications Board (NRC/CSTB) Study, "**Improving Cybersecurity Research in the United States"**, 2004-2007.

Committee Member, 1st Santa Fe Institute Workshop on Malware, 2006.

Committee Member, Potomac Institute for Policy Studies, "**DARPA Computer Science Futures Study**", 2006.

Committee Member, Department of Homeland Security/DARPA/Army Research Office Workshop on Botnets, 2006.

Committee Member, IETF Internet Architecture Board Meeting on Unwanted Traffic, 2006.

Panel Member, "Smoke, Mirrors and Virtualization", Network and Distributed Systems Security Symposium, 2006.

Committee Member, Department of Homeland Security/Army Research Office Workshop on Malcode Detection, 2005.

Co-Chair, DARPA ISAT Study on "Bolt-on Security", 2004.

Panelist, Microsoft Research Mesh Networking Summit, 2004.

Co-Organizer, Department of Homeland Security (DHS) Program Development Workshop on Large-scale Datasets for Network Security Testing and Evaluation, 2004.

Co-founder and Steering Committee Member, ACM Workshop on Rapid Malcode, 2003-2007.

Committee Member, CRA Conference on Grand Challenges in Information Assurance and Security, 2003.

Co-Organizer, DIMACS Workshop on Large-scale Internet Attacks, 2003.

Panelist, Institute for Pure & Applied Mathematics Workshop on Large Scale Communications Networks, 2003.

Panelist, UW-MSR-CMU Software Security Summer Institute, 2003.

Co-Organizer, ACM Workshop on Network Research: Exploration of Dimensions and Scope (NREDS), 2003.

Co-Chair, DARPA ISAT Incubator Study on Software Design for Large-Scale Embedded Systems, 2003.003.

Committee Member, DARPA Cognitive Networking Workshop, 2002.

Session Organizer, Institute for Pure & Applied Mathematics Workshop on Large Scale Communications Networks, 2002.

Committee Member, Internet2 Academic Advisory Board, 2002-2004.

Committee Member, Defense Advanced Projects Research Agency Information Science And Technology Study Group (DARPA ISAT), 2001-2004 (mandated 3 year term)

Peer Review Committee Member, Active Warden Research Project, Los Alamos National Lab, 2001-2004.

Committee Member, National Research Council's Computer Science and Telecommunication Board study, *Research Horizons in Networking*, 2001.

Executive Committee Member, IEEE Technical Committee on Operating Systems, 1999-2004.

Referee for ACM SIGCOMM, TON, SOSP, PLDI, ASPLOS, TOCS, and CCR, USENIX TC and OSDI, IEEE INFOCOM, TPDS, RTSS and others.

University/Department Service Activities

Co-chair, CSE Fellows postdoctoral award committee, 2020-2022

Recruiting committee, GPS Dean search, 2020-2021

Hiring committee, UCSD Center for Peace and Security Studies postdoctoral program, 2021.

Co-Chair, UCSD Department of Computer Science, Brochure design and content committee, 2018-2020

Co-chair, UCSD Department of Computer Science, Faculty Recruiting Committee, 2018.

Advisory Board Member, University of California (systemwide) Cyber Risk Governance Committee, 2016-present.

Ad hoc member, University of California, San Diego, Cybersecurity Governance Committee, 2017-present.

Recruiting committee, University Librarian search, 2017-2018.

Chair, UCSD Department of Computer Science, Faculty Recruiting Committee, 2007, 2014, 2020

Chair, UCSD Department of Computer Science and Engineering Publicity Committee, 2005, 2006, 2007. 2008, 2009, 2010, 2011, 2012. 2015

UCSD Department of Computer Science, Executive Committee, 2004 -- 2010.

UCSD Academic Senate Representative, 2004-2005.

UCSD Department of Computer Science and Engineering, Faculty Recruiting Committee, 2001, 2002, 2003, 2006, 2008 (*ex officio*) 2012, 2015, 2020.

UCSD Department of Computer Science and Engineering, External Relations Committee, 2002, 2003.

UCSD Department of Computer Science and Engineering, Ad Hoc Hiring Committee for Manager of Academic and External Affairs, 2001.

San Diego Network Access Point Advisory Board, Department of Computer Science and Engineering representative, 2002.

San Diego Supercomputer Center, Committee on Networking and Security Visions, 2001.

University of Washington Department of Computer Science and Engineering, Ad Hoc Committee on Commercialization in the Academic Environment, 1998-1999.

Articles

1. Jeffrey Tully, Mark Jarrett, Stefan Savage, Joshua Corman, Christian Dameff, "**Digital Defenses for Hacked Hearts: Why Software Patching Can Save Lives**", Journal of the American College of Cardiology, 72(1): 126-127, July 2018. **(note: letter)**
2. Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker and Stefan Savage, "**A Fistful of Bitcoins: Characterizing Payments Among Men with No Names**", USENIX 'login: 38(6), December 2013. . **(note: ;login not externally refereed, single editor)**
3. Harsha V. Madhyastha, John C. McCullough, George Porter, Rishi Kapoor, Stefan Savage, Alex C. Snoeren and Amin Vahdat, "**scc: Cluster Storage Provisioning Informed by Application Characteristics and SLAs**", USENIX ;login: 36(3), June 2012. **(note: ;login not externally refereed, single editor)**
4. Erik Buchanan, Ryan Roemer, Hovav Shacham and Stefan Savage, "**Return-oriented Programming: Systems, Languages and Applications**", *ACM Transactions on Information and Systems Security* (TISSEC), 15(1), March 2012.
5. Ding Yuan, Soyeon Park, Jing Zheng, Yuanyuan Zhou and Stefan Savage, "**LogEnhancer: Automatic Tailored Log Enhancement to Improve Software Diagnosability against Production-Run Failures**", *ACM Transactions on Computer Systems*, 30(1), February 2012.
6. Rik Farrow interview with Stefan Savage, "**On the Spam Payment Trail**", USENIX ;login: 35(4), August 2011. **(note: ;login not externally refereed, invited interview w/single editor)**
7. Mikhail Afanasyev, Justin Ma, Nicholas Murphy, Yoshi Kohno, Stefan Savage, Alex Snoeren and Geoffrey M. Voelker, **"Network Support for Privacy-preserving Forensic Attribution"**, *Communications of the ACM* (CACM), 54(5):78-87, May 2011.
8. Justin Ma, Lawrence Saul, Stefan Savage and Geoffrey M. Voelker, "**Learning to Detect Malicious URLs**", In *ACM Transactions on Intelligent Systems and Technology* (TIST), 2(3), April 2011.
9. Diwaker Gupta, Sangmin Lee, Michael Vrable, Stefan Savage, Alex C. Snoeren, George Varghese, and Geoffrey M. Voelker, "**The Difference Engine: Harnessing Memory Redundancy in Virtual Machines**", *Communications of the ACM* (CACM), 53(10): 85-93, October 2010.
10. Michael Vrable, Stefan Savage and Geoffrey M. Voelker, "**Cumulus: Filesystem Backup to the Cloud**", *ACM Transactions on Storage*, 5(4): 1-28, December 2009.
11. Michael Vrable, Stefan Savage and Geoffrey M. Voelker, "**Cumulus: Filesystem Backup to the Cloud**", USENIX ;login: 34(4):7-13, August 2009. **(note: ;login not externally refereed, single editor)**
12. Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Vern Paxson, Geoff Voelker and Stefan Savage, **"Spamalytics: an Empirical Analysis of Spam Marketing Conversion"**, *Communications of the ACM* (CACM), 52(9):99-107, September 2009.
13. Diwaker Gupta, Sangmin Lee, Michael Vrable, Stefan Savage, Alex C. Snoeren, George Varghese, and Geoffrey M. Voelker, "**The Difference Engine: Harnessing Memory Redundancy in Virtual Machines**", USENIX ;login: 34(2): 24-31, April 2009. **(note: ;login not externally refereed, single editor)**
14. Alper Mizrak, Keith Marzullo and Stefan Savage, "**Detecting Malicious Packet Losses**", *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 20(2): 191-206, February 2009.
15. Brandon Enright, Geoff Voelker, Stefan Savage, Chris Kanich and Kirill Levchenko, "**Storm: when researchers collide**", USENIX ;login: 33(4): 6-13, August 2008. **(note: ;login not externally refereed, single editor)**
16. Alper Mizrak, Stefan Savage and Keith Marzullo, "**Detecting Compromised Routers via Packet Forwarding Behavior**", *IEEE Network*, 22(2):34-39, March 2008.

17. Fan Chung, Ron Graham, Ranjita Bhagwan, Geoffrey M. Voelker and Stefan Savage, "**Maximizing Data Locality in Distributed Systems**", *Journal of Computer and System Sciences*, 72(8):1309-1306, December 2006.
18. Alper Mizrak, Yu-Chung Cheng, Keith Marzullo and Stefan Savage, "**Detecting Malicious Routers via Traffic Validation**", *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 3(3): 230-244, July 2006.
19. David Moore, Geoffrey M. Voelker and Stefan Savage, "**Inferring Internet Denial-of-Service Activity"**, *ACM Transactions on Computer Systems (TOCS)*, 24(2): 115-139, May 2006.
20. David Moore, Vern Paxson Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver, "**Inside the Slammer Worm**", *IEEE Security and Privacy*, 1(4):33-39, July 2003.
21. Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson, "**Network Support for IP Traceback**", *IEEE/ACM Transactions on Networking (ToN)*, 9(3):226-237, June 2001.
22. Stefan Savage, Neal Cardwell, David Wetherall, and Tom Anderson, "**TCP Congestion Control with a Misbehaving Receiver**", *Computer Communications Review*, 29(5):71-78, October 1999.
23. Stefan Savage, Tom Anderson, Amit Aggarwal, David Becker, Neal Cardwell, Andy Collins, Eric Hoffman, John Snell, Amin Vahdat, Geoffrey M. Voelker and John Zahorjan, "**Detour: a Case for Informed Internet Routing and Transport**", *IEEE Micro,* 19(1):50-59, January 1999.
24. Stefan Savage, Michael Burrows, Greg Nelson, Patrick Sobalvarro and Tom Anderson, "**Eraser: A Dynamic Data Race Detector for Multithreaded Programs**", *ACM Transactions on Computer Systems (TOCS)*, 15(4):391-411, November 1997.

Refereed Conference and Workshop Papers

1. Evan Johnson, Maxwell Bland, Yifei Zhu, Joshua Mason, Stephen Checkoway, Stefan Savage and Kirill Levchenko, "**Jetset: Targeted Firmware Rehosting for Embedded Systems"**, In *Proceedings of the USENIX Security Symposium,* Vancouver, B.C., Canada, August 2021.
2. Anil Yelam, Ariana Mirian, Keerthana Ganesan, Shibani Subbareddy, and Stefan Savage, **"CoResident Evil: Covert Communications in the Cloud with Lambdas"** In *Proceedings of the Web Conference (WWW)*, Ljubljana, Solvenia, April 2021.
3. Guo Li, Gautam Akiwate, Kirill Levchenko, Geoffrey M. Voelker and Stefan Savage, "**Clairvoyance: Inferring Blocklist Use on the Internet**", *Proceedings of the Passive and Active Measurement Conference (PAM)*, Brandenberg, Germany, March 2021. (44% acceptance, 33/75)
4. Evan Johnson, David Thien, Yousef Alhessi, Shravan Narayan, Fraser Brown, Sorin Lerner, Tyler McMullen, Stefan Savage, and Deian Stefan, "**Доверяй, но проверяй: SFI safety for native-compiled Wasm**", In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2021.
5. Gautam Akiwate, Raffaele Sommese, Mattijs Jonker, Ian Foster, Stefan Savage Geoffrey M. Voelker and kc Claffy, "**Unresolved Issues: Prevalence, Persistence and Perils of Lame Nameservers**", In *Proceedings of the Internet Measurement Conference (IMC)*, Pittsburgh, Pennsylvania, October 2020. (25% acceptance, 53/216)
6. Audrey Randall, Enze Liu, Gautam Akiwate, Ramakrishna Padmanabhan, Stefan Savage, Geoffrey M. Voelker and Aaron Schulman, "**TruffleHunter: Cache Sniffing Rare Domains at Large Public DNS Resolvers**", In *Proceedings of the Internet Measurement Conference (IMC)*, Pittsburgh, Pennsylvania, October 2020. 25% acceptance, 53/216)
7. Louis DeKoven, Audrey Randall, Ariana Mirian, Gautam Akiwate, Ansel Blume, Lawrence K. Saul, Aaron Shulman, Geoffrey M. Voelker and Stefan Savage, "**Measuring Security Practices and How They Impact Security**", In *Proceedings of the ACM Internet Measurement Conference (IMC)*, Amsterdam, Netherlands, October 2019. (19% acceptance, 38/197)

8. Sam Crow, Brown Farinholt, Brian Johannesmeyer, Karl Koscher, Stephen Checkoway, Stefan Savage, Aaron Schulman, Alex C. Snoeren and Kirill Levchenko, "**Triton: Software-Reconfigurable Federated Avionics Testbed**", In *Proceedings of the Workshop on Cyber Security Experimentation and Test (CSET)*, Santa Clara, CA August 2019. (31% acceptance, 19/61)

9. Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M. Voelker, Stefan Savage and Kirill Levchenko, "**Reading the Tea Leaves: A Comparative Analysis of Threat Intelligence**", In *Proceedings of the USENIX Security Symposium*, Santa Clara, CA, August 2019. (19% acceptance, 88/465)

10. Grant Ho, Asaf Cidon, Lior Gavish, Marco Schweighauser, Vern Paxson, Stefan Savage, Geoffrey M. Voelker and David Wagner, "**Detecting and Characterizing Lateral Phishing at Scale**", In *Proceedings of the USENIX Security Symposium*, Santa Clara, CA, August 2019. (19% acceptance, 88/465) (**Award paper**)

11. Ariana Mirian, Joe DeBlasio, Stefan Savage, Geoffrey M. Voelker and Kurt Thomas, "**Hack for Hire: Exploring the Emerging Market for Account Hijacking**", In Proceedings of the World Wide Web Conference, San Francisco, CA, May 2019. (18% acceptance, 225/1247)

12. Ariana Mirian, Nikunj Bhagat, Caitlin Sadowski, Adrienne Porter Felt, Stefan Savage and Geoffrey M. Voelker, "**Web Feature Deprecation: A Case Study for Chrome**", In *Proceedings of the International Conference on Software Engineering* (ICSE), Montreal, QC, Canada, May 2019. (25% acceptance, 30/118 SEIP track)

13. Stefan Savage, "**Lawful Device Access without Mass Surveillance Risk**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Toronto, Canada, October 2018. (17% acceptance 134/809)

14. Louis F Dekoven, Trevor Pottinger, Stefan Savage, Geoffrey M. Voelker and Nektarios Leontiadis, "**Following their Footsteps: Characterizing Account Automation Abuse and Defenses**", In *Proceedings of the Internet Measurement Conference (IMC)*, Chicago, IL, November 2018. (23% acceptance, 40/174)

15. Alexander Gamero-Garrido, Stefan Savage, Kirill Levchenko and Alex C. Snoeren, "**Quantifying the Pressure of Legal Risks on Third Party Vulnerability Research**", In Proceedings of the ACM Conference on Computer and Communication Security, Dallas, TX, October 2017. (18% acceptance, 151/836)

16. Joe DeBlasio, Stefan Savage, Geoffrey M. Voelker and Alex C. Snoeren, "**Tripwire: Inferring Internet Site Compromise**", In *Proceedings of the Internet Measurement Conference*, London, UK, November 2017. (23% acceptance, 42/179)

17. Louis DeKoven, Stefan Savage, Geoffrey M. Voelker and Nektarious Leontiadis, "**Malicious Browser Extensions at Scale: Bridging the Observability Gap between Web Site and Browser**", In *Proceedings of the USENIX Workshop on Cybersecurity Experimental and Test (CSET)*, Vancouver, Canada, August 2017.

18. Mortiz Contag, Guo Li, Andre Pawlowski, Flexi Domke, Stefan Savage, Kirill Levchenko, and Thorsten Holz, "**How They Did It: An Analysis of Emission Defeat Devices in Modern Automobiles**", In *Proceedings of the IEEE Symposium on Security and Privacy*, San Jose, CA, May 2017 (13% acceptance, 60/450)

19. Wilson Lian, Hovav Shacham and Stefan Savage, "**A Call to ARMs: Understanding the Costs and Benefits of JIT Spraying Mitigations**", In *Proceedings of the Network and Distributed Systems Symposium* (NDSS), San Diego, CA, February 2016. (16% acceptance, 68/463)

20. Frank Li, Zakir Durumeric, Jakub Czyz, Damon McCoy, Stefan Savage, Michael Bailey, and Vern Paxson, "**You've Got Vulnerability: Exploring Effective Vulnerability Notifications"**, In*Proceedings of the USENIX Security Symposium*, Austin, TX, August 2016. (17% acceptance, 72/423)

21. He Liu, Matthew K. Mukerjee, Conglong Li, Nicolas Feltman, George Papen, Stefan Savage, Srinivasan Seshan, Geoffrey M. Voelker, David G. Anderson, Michael Kaminsky, George Porter and Alex C. Snoeren, "**Scheduling Techniques for Hybrid Circuit/Packet Networks**", In *Proceedings of the ACM CoNEXT Conference*, Heidelberg, Germany, December 2015. (20% acceptance, 37/186)

22. Ian Foster, Jon Larson, Max Masich, Alex C. Snoeren, Stefan Savage and Kirill Levchenko, "**Security by Any Other Name: On the Effectiveness of Provider Based Email Security**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Denver, CO, October 2015. (20% acceptance, 128/646)

23. Suqi Liu, Ian Foster, Stefan Savage, Geoffrey M. Voelker and Lawrence K. Saul,, "**Who is .com? Learning to Parse WHOIS Records**", In *Proceedings of the ACM Internet Measurement Conference* (IMC), Tokyo, Japan, October 2015. (26% acceptance, 44/146)

24. Tristan Halvorson, Matthew F. Der, Ian Foster, Stefan Savage, Geoffrey M. Voelker and Lawrence K. Saul,, "**From .academy to .zone: An Analysis of the new TLD Land Rush**", In *Proceedings of the ACM Internet Measurement Conference* (IMC), Tokyo, Japan, October 2015. (26% acceptance, 44/146)

25. Neha Chachra Stefan Savage and Geoffrey M. Voelker, "**Affiliate Crookies: Characterizing Affiliate Marketing Abuse**", In *Proceedings of the ACM Internet Measurement Conference* (IMC), Tokyo, Japan, October 2015. (16% acceptance,, 12/73)

26. Kurt Thomas, Danny Yuxing Huang, David Wang, Elie Burszstein, Chris Grier, Tom Holt, Christopher Kruegel, Damon McCoy, Stefan Savage and Giovanni Vigna, "**Framing Dependencies Introduced by Underground Commoditization**", In *Proceedings of the Workshop on the Economics of Information Security (WEIS)*, Delft, The Netherlands, June 2015.

27. Wilson Lian, Hovav Shacham and Stefan Savage, "**Too LeJIT to Quite: Implementing Jit-spryaing on ARM**", In *Proceedings of the ISOC Network and Distributed Systems Symposium (NDSS)*, San Diego, February 2015. (17% acceptance rate, 15/302)

28. David Y. Wang and Matt Der and Mohammad Karami and Lawrence Saul and Damon McCoy and Stefan Savage and Geoffrey M. Voelker, "**Search + Seizure: The Effectiveness of Interventions on SEO Campaigns**", In *Proceedings of the ACM Internet Measurement Conference (IMC)*, Vancouver, BC, Canada, Nov 2014. (23% acceptance, 43/188)

29. Borbala Benko, Elie Bursztein, Daniel Margolis, Tadek Pietraszek, Andy Archer, Allan Aquino, Andreas Pitsillidis and Stefan Savage, "**Handcrafted Fraud and Extortion: Manual Account Hijacking in the Wild**", In *Proceedings of the ACM Internet Measurement Conference (IMC)*, Vancouver, BC, Canada, Nov 2014. . (23% acceptance, 43/188)

30. David Lundberg, Brown Farinholt , Edward Sullivan, Ryan Mast, Stefan Savage, Alex Snoeren and Kirill Levchenko, "**On the Security of Mobile Cockpit Information Systems**", In *Proceedings of the ACM Conference on Computer and Communication Security (CCS)*, Scottsdale, AZ, Nov 2014. (19% acceptance, 114/585)

31. Paul Pearce, Chris Grier, Vern Paxson, Vacha Dave, Damon McCoy, Kirill Levchenko, Geoffrey M. Voelker, Stefan Savage and Saikat Guha, "**Characterizing Large-Scale Click Fraud in ZeroAccess**", In *Proceedings of the ACM Conference on Computer and Communication Security (CCS),* Scottsdale, AZ, Nov 2014. (19% acceptance, 114/585)

32. Mathew F. Der, Lawrence K. Saul, Stefan Savage and Geoffrey M. Voelker, "**Knock it Off: Profiling the Online Storefronts of Counterfeit Merchandise**", In *Proceedings of the ACM SIG KDD Conference*, New York, NY, August 2014. (22% acceptance, 44/197)

33. Neha Chachra, Damon McCoy, Stefan Savage and Geoffrey M. Voelker, "**Empirically Characterizing Domain Abuse and the Revenue Impact of Blacklisting**", In *Proceedings of the Workshop on the Economics of Information Security (WEIS)*, State College, PA, June 2014. (23% acceptance, 9/40)

34. Tristan Halvorson, Kirill Levchenko, Stefan Savage and Geoffrey M. Voelker, "**XXXtortion? Inferring Registration Intent in the .XXX TLD**", In *Proceedings of the World Wide Web Conference (WWW)*, Seoul, Korea, April 2014. (13% acceptance, 84/650)

35. Danny Huang, Sarah Meiklejohn, Damon McCoy, Chris Grier, Kirill Levchenko, Alex Snoeren and Stefan Savage, "**Botcoin: Monetizing Stolen Cycles**", In *Proceedings of the ISOC Network and Distributed Security Symposium (NDSS)*, San Diego, CA, February 2014. (19% acceptance, 55/295)

36. Daniel Turner, Kirill Levchenko, Stefan Savage and Alex Snoeren, "**A Comparison of Syslog and IS-IS for Monitoring Link State**", In *Proceedings of the ACM Internet Measurement Conference* (IMC), Barcelona, Spain, October 2013. (24% acceptance, 42/178)

37. Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker and Stefan Savage, "**A Fistful of Bitcoins: Characterizing Payments Among Men with No Names**", In

Proceedings of the ACM Internet Measurement Conference (IMC), Barcelona, Spain, October 2013. (24% acceptance, 42/178)

38. Wilson Lian, Eric Rescorla, Hovav Shacham and Stefan Savage, "**Measuring the Practical Impact of DNSSEC Deployment**", In *Proceedings of the USENIX Security Symposium*, Washington D.C., August 2013. (16% acceptance, 45/277)

39. David Wang, Stefan Savage and Geoffrey M. Voelker, "**Juice: A Longitudinal Study of an SEO Campaign**", In *Proceedings of the ISOC Network and Distributed Security Symposium (NDSS)*, San Diego, CA, February 2013. (19% acceptance, 47/250)

40. Andreas Pittsilidis, Chris Kanich, Kirill Levchenko, Geoffrey M. Voelker and Stefan Savage, "**Taster's Choice: A Comparative Analysis of Spam Feeds**", In *Proceedings of the ACM Internet Measurement Conference* (IMC), Boston, MA, November 2012. (25% acceptance, 45/183)

41. Ding Yuan, Soyeon Park, Peng R. Huang, Yang R. Liu, Mihnjong M. Lee, Xiaoming Tang, Yuanyuan Zhou and Stefan Savage, "**To Log or Not to Log: Characterizing and Improving Software Error Reporting**", In *Proceedings of the ACM/USENIX Operating System Design and Implementation Conference (OSDI)*, Hollywood, CA, October 2012. (12% acceptance, 25/215)

42. Chris Grier, Lucas Ballard, Juan Caballero, Neha Chachra, Christian J. Dietrich, Kirill Levchenko, Panayiotis Mavrommatis, Damon McCoy, Antonio Nappa, Andreas Pitsillidis, Niels Provos, Zubair Rafique, Moheed Abu Rajab, Christian Rossow, Kurt Thomas, Vern Paxson, Stefan Savage and Geoffrey M. Voelker, "**Browser Exploits as a Service: The Monetization of Driveby Downloads**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Raleigh, NC, October 2012. (19% acceptance, 80/423)

43. Damon McCoy, Hitesh Dharmdasani, Christian Kreibich, Geoffrey M. Voelker and Stefan Savage, "**Priceless: The Role of Payments in Abuse-advertised Goods**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Raleigh, NC, October 2012. (19% acceptance, 80/423)

44. Damon McCoy, Andreas Pitsillidis, Grant Jordan, Nicholas Weaver, Christian Kreibich, Brian Krebs, Geoffrey M. Voelker, Stefan Savage and Kirill Levchenko, "**PharmaLeaks: Understanding the Business of Online Pharmaceutical Affiliate Program**s*", Proceedings of the USENIX Security Symposium*, Bellevue, WA, August 2012. (19% acceptance, 43/222)

45. Feng Lu, Jiaqi Zhang and Stefan Savage, "**When Good Services Go Wild: Reassembling Web Services for Unintended Pruposes**", *Proceedings of the USENIX Workshop on Hot Topics in Security (HotSec)*, Bellevue, WA, August 2012. (28% acceptance, 11/39)

46. Ross Anderson, Chris Barton, Rainer Boehme, Richard Clayton, Michel J.G. van Eeten, Michael Levi , Tyler Moore and Stefan Savage, "**Measuring the Cost of Cybercrime**", *Proceedings of the Workshop on the Economics of Information Security* (WEIS), Berlin, Germany, June 2012. (38% acceptance, 20/53)

47. Michael Vrable, Stefan Savage and Geoffrey M. Voelker, "**A Cloud-Backed File System for the Enterprise**", *Proceedings of the USENIX Conference on File and Storage Technologies* (FAST), San Jose, CA, February 2012. (19% acceptance, 26/137)

48. Harsha V. Madhyastha, John C. McCullough, George Porter, Rishi Kapoor, Stefan Savage, Alex C. Snoeren and Amin Vahdat, "**scc: Cluster Storage Provisioning Informed by Application Characteristics and SLAs**", *Proceedings of the USENIX Conference on File and Storage Technologies* (FAST), San Jose, CA, February 2012. (19% acceptance, 26/137)

49. Sushma Nagesh Bannur, Lawrence K. Saul and Stefan Savage, "**Judging a site by its content: learning the textual, structural, and visual features of malicious Web pages**", *Proceedings of the ACM Workshop on Artificial Intelligence and Security* (AISEC), Chicago, IL, October 2011.

50. Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage and Geoffrey M. Voelker, "**An Analysis of Underground Forums**", In *Proceedings of the ACM Internet Measurement Conference* (IMC), Berlin, Germany, November 2011. (19% acceptance, 42/220)

51. David Wang, Stefan Savage and Geoffrey M. Voelker, "**Cloak and Dagger: Dynamics of Web Search Cloaking**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Chicago, IL, October 2011. (14% acceptance, 60/429)

52. Daniel J. Turner, Stefan Savage and Alex Snoeren, **"On the Empirical Performance of Self-calibrating WiFi Location**", In *Proceedings of the IEEE Conference on Local Computer Networks* (LCN), Bonn, Germany, October 2011. (<30% acceptance)

53. Keaton Mowrey, Sarah Meiklejohn and Stefan Savage, "**Heat of the Moment: Characterizing the Efficacy of Thermal Camera-Based Attacks**", In *Proceedings of the USENIX Workshop on Offensive Technologies (WOOT)*, August 2011. (43% acceptance, 15/35)

54. Chris Kanich, Neha Chachra, Damon McCoy, Chris Grier, David Wang, Marti Motoyama, Kirill Levchenko, Geoffrey M. Voelker and Stefan Savage, "**No Plan Survives Contact: Experience with Cybercrime Measurement**", In *Proceedings of the USENIX Workshop on Cyber Security Experimentation and Test (CSET)*, August 2011. (40% acceptance, 12/30)

55. Chris Kanich, Nicholas Weaver, Damon McCoy, Tristan Halvorson, Christian Kreibich, Kirill Levchenko, Vern Paxson, Geoffrey M. Voelker and Stefan Savage, "**Show Me the Money: Characterizing Spam-advertised Revenue**", In *Proceedings of the USENIX Security Symposium*, August 2011. (17% acceptance, 35/204)

56. Stephen Checkoway, Damon McCoy, Danny Anderson, Brian Kantor, Hovav Shacham, Stefan Savage , Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno , "**Comprehensive Experimental Analysis of Automotive Attack Surfaces**", In *Proceedings of the USENIX Security Symposium*, August 2011. (17% acceptance, 35/204)

57. Marti Motoyama, Damon McCoy, Kirill Levchenko, Geoffrey M. Voelker, Stefan Savage, "**Dirty Jobs: The Role of Freelance Labor in Web Service Abuse**", In *Proceedings of the USENIX Security Symposium*, August 2011. (17% acceptance, 35/204)

58. Mashael AlSabah, Kevin Bauer and Ian Goldberg and Dirk Grunwald and Damon McCoy and Stefan Savage and Geoffrey M. Voelker, "**DefenestraTor: Throwing out Windows in Tor**", In *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, Waterloo, Canada, July 2011. (25% acceptance, 15/61)

59. Kirill Levchenko, Neha Chachra, Brandon Enright, Mark Felegyhazi, Chris Grier, Tristan Halvorson, Chris Kanich, Christian Kreibich, He Liu, Damon McCoy, Andreas Pitsillidis, Nicholas Weaver, Vern Paxson, Geoffrey M. Voelker and Stefan Savage, "**Click Trajectories: End-to-End Analysis of the Spam Value Chain**", In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 2011. (11% acceptance, 34/306)

60. He Liu, Kirill Levchenko, Mark Felegyhazi, Christian Kreibich, Gregor Maier, Geoffrey M. Voelker and Stefan Savage, "**On the Effects of Registrar-level Intervention**", In *Proceedings of the USENIX Workshop on Large-Scale Exploits and Emerging Threats (LEET)*, Boston, MA, March 2011. (44% acceptance; 11/25)

61. Qing Zhang, Thomas Ristenpart, Stefan Savage and Geoffrey M. Voelker, "**Got Traffic? An Evaluation of Click Traffic Providers**", In *Proceedings of the WICOW/AIRWeb Workshop on WebQuality (WebQuality)*, Hyderabad, India, March 2011.

62. Ding Yuan, Soyeon Park, Jing Zheng, Yuanyuan Zhou and Stefan Savage, "**LogEnhancer: Automatic Tailored Log Enhancement to Improve Software Diagnoseability against Production-Run Failures**", In *Proceedings of the ACM Architecture Support for Programming Languages and Operating Systems (ASPLOS) Conference*, March 2011. (21% acceptance, 32/152)

63. Andreas Pitsillidis, Yinglian Xie, Fang Yu, Martin Abadi, Geoffrey M. Voelker, and Stefan Savage, "**How to Tell and Airport from a Home: Techniques and Applications**", In *Proceedings of the ACM Workshop on Hot Topics in Networks*, Monterey, CA, October 2010. (21% acceptance, 22/104)

64. Daniel Turner, Kirill Levchenko, Alex C. Snoeren and Stefan Savage, "**California Fault Lines: Understanding the Causes and Impact of Network Failures**", In *Proceedings of the ACM SIGCOMM Conference*, New Dehli, India, August 2010. (12% acceptance, 33/276)

65. Marti Motoyamai, Kirill Levchenko, Chris Kanich, Damon McCoy, Geoffrey M. Voelker and Stefan Savage, **"Re: CAPTCHAs -- Understanding CAPTCHA Breaking from an Economic Context**", In *Proceedings of the USENIX Security Symposium*, Washington, D.C., August 2010. (15% acceptance, 30/202)

66. Mehran Bozorgi, Lawrence Saul, Stefan Savage and Geoffrey M. Voelker, "**Beyond Heuristics: Learning to Classify Vulnerabilities and Predict Exploits**", In *Proceedings of the ACM SIG KDD Conference,* pp. 105—113, Washington D.C. July 2010. (20% acceptance, 20/101)

67. Marti Motoyama, Brendan Meeder, Kirill Levchenko, Geoffrey M. Voelker and Stefan Savage, "**Measuring Online Service Availability Using Twitter**", In *Proceedings of the Workshop on Social Networks (WOSN),* Boston, MA, June 2010. (33% acceptance, 13/33)

68. Yuvraj Agarwal, Stefan Savage and Rajesh Gupta, "**Sleep-Servers: Energy Savings for Enterprise PCs by Allowing them to Sleep**", In *Proceedings of the USENIX Annual Technical Conference*, Boston, MA, June 2010. (17% acceptance, 24/141)

69. Karl Koscher, Alexei Czeskis, Fanziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham and Stefan Savage, **"Experimental Security Analysis of a Modern Automobile**", In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 2010. (11% acceptance, 26/237)

70. Qing Zhang, John McCullough, Justin Ma, Navil Schear, Michael Vrable, Geoffrey M. Voelker, Alex C. Snoeren and Stefan Savage, "**Neon: System Support for Derived Data Management**", In *Proceedings of the ACM Conference on Virtual Execution Environments (VEE)*, Pittsburgh, PA, March 2010. (27% acceptance 14/51).

71. Andreas Pitsillidis, Kirill Levchenko, Chris Kanich, Christian Kreibich, Brandon Enright, Vern Paxson, Geoff Voelker and Stefan Savage, "**Botnet Judo: Attacking Spam with Itself**", In *Proceedings of the ISOC Network and Distributed Security Symposium (NDSS)*, San Diego, CA, February 2010. (15% acceptance, 24/156)

72. Tom Ristenpart, Eran Tromer, Hovav Shacham and Stefan Savage, "**Hey, You, Get off of my Cloud! Exploring Information Leakage in Third-Party Compute Clouds**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Chicago IL, November 2009. (18% acceptance, 58/315)

73. Scott Yilek, Eric Rescorla, Hovav Shacham, Brandon Enright and Stefan Savage, **"When Private Keys are Public: Understanding the Debian OpenSSL Debacle**", In *Proceedings of the USENIX Internet Measurement Conference*, Chicago, IL, November 2009. (22% acceptance, 41/13)

74. Justin Ma, Lawrence K. Saul, Geoffrey M. Voelker and Stefan Savage, **"Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs**", In *Proceedings of the ACM SIGKDD Conference*, Paris, France, July 2009. (28% acceptance, 34/122)

75. Justin Ma, Lawrence Saul, Stefan Savage and Geoffrey M. Voelker, "**Identifying Suspicious URLs: An Application of Large-scale Online Learning**", In *Proceedings of the International Conference on Machine Learning (ICML)*, Montreal, Quebec, June 2009. (26% acceptance, 160/595)

76. Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Vern Paxson, Geoff Voelker and Stefan Savage, "**Spamcraft: an Inside Look at Spam Campaign Orchestration**", In *Proceedings of the USENIX Workshop on Large-Scale Exploits and Emerging Threats (LEET)*, Boston, MA, April 2009 (36% acceptance, 9/25)

77. Michael Vrable, Stefan Savage and Geoffrey M. Voelker, "**Cumulus: Filesystem Backup to the Cloud**", In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST)*, pp. 225-238, San Francisco, CA, February 2009. (23% acceptance, 23/102)

78. Diwaker Gupta, Sangmin Lee, Michael Vrable, Stefan Savage, Alex C. Snoeren, George Varghese, and Geoffrey M. Voelker, "**The Difference Engine: Harnessing Memory Redundancy in Virtual Machines**", In *Proceedings of the ACM/USENIX Operating System Design and Implementation Conference (OSDI)*, pp. 309-322, San Diego, CA, December 2008. (13% acceptance, 26/193) **(Jay Lepreau Best Paper Award)**

79. Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Vern Paxson, Geoff Voelker and Stefan Savage, "**Spamalytics: an Empirical Analysis of Spam Marketing Conversion**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pp. 3-14, Alexandria, VA, October 2008. (18% acceptance, 51/281)

80. Benjamin Laxton, Kai Wang and Stefan Savage, "**Reconsidering Physical Key Secrecy: Teleduplication via Optical Decoding**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pp. 469-478, Alexandria, VA, October 2008. (18% acceptance, 51/281)

81. Erick Buchanan, Ryan Roemer, Hovav Shacham and Stefan Savage, "**When Good Instructions Go Bad: Generalizing Return-oriented Programming to the SPARC**", In *Proceedings of the ACM Conference on*

Computer and Communications Security (CCS), pp. 27-38, Alexandria, VA, October 2008. (18% acceptance, 51/281)

82. Kirill Levchenko, Geoffrey M. Voelker, Mohan Paturi and Stefan Savage, "**XL: An Efficient Network Routing Protocol**", In *Proceedings of ACM SIGCOMM*, pp. 15-26, Seattle, WA, August 2008. (13% acceptance, 37/288)

83. Christian Kriebeck, Chris Kanich, Kirill Levchenko, Brandon Enright, Vern Paxson, Geoff Voelker and Stefan Savage, "**On the Spam Campaign Trail**", In *Proceedings of the USENIX Workshop on Large-Scale Exploits and Emerging Threats (LEET)*, San Francisco, CA, April, 2008. (32% acceptance, 11/34)

84. Brandon Enright, Chris Kanich, Kirill Levchenko, Geoff Voelker and Stefan Savage, "**Understanding the Heisenbot Problem**", In Proceedings of the *USENIX Workshop on Large-Scale Exploits and Emerging Threats (LEET)*, San Francisco, CA, April, 2008. (32% acceptance, 11/34)

85. Jason Franklin, Adrian Perrig, Vern Paxson and Stefan Savage, "**An Inquiry into the Nature and Causes of the Wealth of Internet Miscreants**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Alexandria, VA, October 2007. (18% acceptance, 55/303)

86. Chris Fleizach, Geoffrey M. Voelker and Stefan Savage, "**Slicing Spam with Occam's Razor**", In *Proceedings of the Conference on E-mail and Anti-Spam (CEAS)*, Mountain View, CA, August 2007. (32% acceptance, 25/79)

87. Yu-Chung Cheng, Mikhail Afanasyev, Peter Benko, Patrick Verkaik, Jennifer Chiang, Alex C. Snoeren, Stefan Savage, and Geoffrey M. Voelker, "**Automating Cross-Layer Diagnosis of Enterprise Wireless Networks**", In *Proceedings of the ACM SIGCOMM Conference*, Kyoto, Japan, August 2007 (14% acceptance, 35/258)

88. David S. Anderson, Chris Fleizach, Stefan Savage and Geoffrey M. Voelker, "**Spamscatter: Characterizing Internet Scam Hosting Infrastructure**", In *Proceedings of the USENIX Security Symposium*, Boston, MA, August 2007. (12% acceptance, 23/187)

89. Justin Ma, Kirill Levchenko, Christian Kriebeck, Stefan Savage and Geoffrey M. Voelker, "**Automated Protocol Inference: Unexpected Means of Identifying Protocols**", In *Proceedings of the ACM/USENIX Internet Measurement Conference (IMC)*, Rio de Janerio, Brazil, October 2006. (22% acceptance, 34/155 – one of 19 long papers)

90. Justin Ma, John Dunagan, Helen J. Wang, Stefan Savage and Geoffrey M. Voelker, "**Finding Diversity in Remote Code Injection Exploits**", In *Proceedings of the ACM/USENIX Internet Measurement Conference (IMC)*, Rio de Janerio, Brazil, October 2006. (22% acceptance, 34/155 – one of 19 long papers)

91. Yu-Chung Cheng, John Bellardo, Peter Benko, Alex Snoeren, Geoffrey M. Voelker and Stefan Savage, "**Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis**", In *Proceedings of the ACM SIGCOMM Conference*, Pisa, Italy, September 2006 (12% acceptance, 37/298).

92. Martin Casado, Tal Garfinkel, Weidong Cui, Vern Paxson, and Stefan Savage, "**Opportunistic Measurement: Spurious Network Events as a Light in the Darkness**", In *Proceedings of the 4th ACM Workshop on Hot Topics in Networks (HotNets)*, College Park, MD, November 2005. (20% acceptance, 20/102)

93. Justin Ma, Geoffrey M. Voelker and Stefan Savage, "**Self-stopping Worms**", In *Proceedings of the ACM Workshop on Rapid Malcode (WORM)*, Fairfax, VA, November 2005. (25% acceptance, 9/37)

94. Michael Vrable, Justin Ma, Jay Chen, David Moore, Erik Vadekeift, Alex S. Snoeren, Geoffrey M. Voelker and Stefan Savage, "**Scalability, Fidelity and Containment in the Potemkin Virtual Honeyfarm**", In *Proceedings of the ACM Symposium on Operating System Principles (SOSP)*, Brighton, UK, October 2005. (13% acceptance, 20/155)

95. Alper Mizrak, Yu-Chung Cheng, Keith Marzullo and Stefan Savage, "**Fatih: Detecting Malicious Routers via Traffic Validation**", In *Proceedings of the IEEE Symposium on Dependable Systems and Networks (DSN)*, Yokohama, Japan, June 2005. (24% acceptance, 78/319) (**Award paper**)

96. Ishwar Ramani and Stefan Savage, "**SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks**", In *Proceedings of the IEEE Infocom Conference*, Miami, FL, March 2005. (17% acceptance, 244/1419)

97. Sumeet Singh, Cristian Estan, George Varghese and Stefan Savage, "**Automatic Worm Fingerprinting**", In *Proceedings of the ACM/USENIX Operating System Design and Implementation Conference (OSDI),* San Francisco, CA, December 2004. (14% acceptance, 27/193)

98. Yu-Chung Cheng, Urs Hoelzle, Neal Cardwell, Stefan Savage and Geoffrey M. Voelker, "**Monkey See, Monkey Do: A Tool for TCP Tracing and Replaying**", In *Proceedings of the USENIX Technical Conference*, Boston, MA, June 2004. (13% acceptance, 21/164)

99. Ranjita Bhagwan, Kiran Tati, Yu-Chung Cheng, Stefan Savage and Geoffrey M. Voelker, "**Total Recall: System Support for Automated Availability Management**", In *Proceedings of the USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, March 2004. (23% acceptance; 27/118)

100. Renata Teixeira, Keith Marzullo, Stefan Savage and Geoffrey M. Voelker, "**In Search of Path Diversity in ISP Networks**", In *Proceedings of the USENIX/ACM Internet Measurement Conference (IMC)*, Miami, FL, October 2003. (30% acceptance; 33/109)

101. Cristian Estan, Stefan Savage and George Varghese, "**Automatically Inferring Patterns of Resource Consumption In Network Traffic**", In *Proceedings of ACM SIGCOMM Conference*, Karlsruhe, Germany, August 2003. (10% acceptance; 33/319)

102. John Bellardo and Stefan Savage, "**802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions**", In *Proceedings of the USENIX Security Symposium*, Washington D.C, August 2003. (17% acceptance; 21/125)

103. Alper Mizrak, Yu-Chung Cheng, Vineet Kumar and Stefan Savage, , "**Structured Superpeers: Leveraging Heterogeneity to Provide Constant-Time Lookup**", In *Proceedings of the IEEE Workshop on Internet Applications*, San Jose, CA, June 2003. (19% acceptance; 10/53 full papers)

104. Flavio Junqueira, Ranjita Bhagwan, Keith Marzullo, Stefan Savage and Geoffrey M. Voelker, "**The Phoenix Recovery System: Rebuilding from the Ashes of an Internet Catastrophe**", In *Proceedings of the USENIX/IEEE Workshop on Hot Topics in Operating Systems*, Lihue, Hawaii, May 2003. (21% acceptance; 31/150; one of 13% invited to present; 19/150)

105. David Moore, Colleen Shannon, Geoffrey M. Voelker and Stefan Savage, "**Internet Quarantine: Requirements for Containing Self-Propagating Code**", In *Proceedings of the IEEE Infocom Conference*, San Francisco, CA, April 2003. (20% acceptance; 224/1100)

106. Ranjita Bhagwan, Stefan Savage and Geoffrey M. Voelker, "**Understanding Availability**", In *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS)*, Berkeley, CA, February 2003. (16% acceptance; 26/165)

107. John Bellardo and Stefan Savage, "**Measuring Packet Reordering**", In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, October, 2002. (26% acceptance; 16/62 full papers)

108. David Ely, Neil Spring, David Wetherall, Stefan Savage and Tom Anderson, "**Robust Congestion Signaling**", In *Proceedings of the International Conference on Network Protocols*, pages 332-341, Riverside, CA, November 2001. (23% acceptance; 36/157) (**Selected Paper**)

109. David Moore, Geoffrey M. Voelker and Stefan Savage, "**Inferring Internet Denial-of-Service Activity**", In *Proceedings of the USENIX Security Symposium*, Washington, D.C., August 2001. (29% acceptance; 24/83) (**Best Paper**)

110. David Ely, Stefan Savage and David Wetherall, "**Alpine: A User-Level Infrastructure for Network Protocol Development**", In *Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS)*, pages 171-183, San Francisco, CA, March 2001. (27% acceptance; 19/70)

111. Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson, "**Practical Network Support for IP Traceback**", In *Proceedings of the ACM SIGCOMM Conference*, pages 295-306, Stockholm, Sweden, August 2000. (11% acceptance; 26/238) (**Selected Paper**)

112. Neal Cardwell, Stefan Savage and Tom Anderson, "**Modeling TCP Latency**", In *Proceedings of the IEEE Infocom Conference*, pages 1742-1751, Tel-Aviv, Israel, March 2000. (26% acceptance; 192/735)

113. Amit Aggarwal, Stefan Savage and Tom Anderson, "**Understanding the Performance of TCP Pacing**", In *Proceedings of the IEEE Infocom Conference*, pages 1157-1165, Tel-Aviv, Israel, March 2000. (26% acceptance; 192/735)

114. Stefan Savage, "**Sting: a TCP-based Network Measurement Tool**", In *Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS)*, pages 71-79, Boulder, Co, October 1999. (33% acceptance; 22/67) (**Best Student Paper**)

115. Stefan Savage, Andy Collins, Eric Hoffman, John Snell, and Tom Anderson, "**The End-to-end Effects of Internet Path Selection**", In *Proceedings of the ACM SIGCOMM Conference*, pages 289-299, Cambridge, MA, September 1999. (13% acceptance; 24/190)

116. Stefan Savage, Neal Cardwell and Tom Anderson, "**The Case for Informed Transport Protocols**", In *Proceedings of the IEEE Workshop on Hot Topics in Operating Systems (HotOS)*, pages 58-63, Rio Rico, AZ, March 1999.

117. Stefan Savage, Michael Burrows, Greg Nelson, Patrick Sobalvarro and Tom Anderson, "**Eraser: A Dynamic Data Race Detector for Multithreaded Programs**", In *Proceedings of the ACM Symposium on Operating System Principles (SOSP)*, pages 27-37, Saint-Malo, France, October 1997. (21% acceptance; 23/109) (**Award Paper**)

118. Wilson Hsieh, Marc Fiuczynski, Charles Garrett, Stefan Savage, David Becker and Brian Bershad, "**Language Support for Extensible Operating Systems**", In *Proceedings of the Workshop on Compiler Support for System Software*, pages 127-133, Tucson, AZ, February 1996.

119. Emin Gun Sirer, Stefan Savage, Przemyslaw Pardyak, Greg Defouw, Mary Ann Alapat and Brian Bershad, "**Writing an Operating System with Modula-3**", In *Proceedings of the Workshop on Compiler Support for System Software*, pages 134-140, Tucson, AZ, February 1996.

120. Stefan Savage and John Wilkes, "**AFRAID – A Frequently Redundant Array of Independent Disks**", In *Proceedings of the USENIX Technical Conference*, pages 27-39, San Diego, CA, January 1996. (22% acceptance; 28/130) (**Best Student Paper**)

121. Brian Bershad, Stefan Savage, Przemyslaw Pardyak, Emin Gun Sirer, David Becker, Marc Fiuczynski, Craig Chambers and Susan Eggers, "**Extensibility, Safety and Performance in the SPIN Operating System**", In *Proceedings of the ACM Symposium on Operating System Principles (SOSP),* pages 267-284, Copper Mountain, CO, December 1995. (26% acceptance; 22/84)

122. Brian Bershad, Stefan Savage, Przemyslaw Pardyak, David Becker, Marc Fiuczynski and Emin Gun Sirer, "**Protection is a Software Issue**", In *Proceedings of the IEEE Workshop on Hot Topics in Operating Systems*, pages 62-65, Orcas Island, WA, May 1995.

123. Brian Bershad, Craig Chambers, Susan Eggers, Chris Maeda, Dylan McNamee, Przemyslaw Pardyak, Stefan Savage and Emin Gun Sirer, "**SPIN – An Extensible Microkernel for Application-specific Operating System Services**", In *Proceedings of the ACM SIGOPS European Workshop*, Dagstuhl, Germany, September 1994.

124. Clifford Mercer, Stefan Savage and Hideyuki Tokuda, "**Processor Capacity Reserves: Operating System Support for Multimedia Applications**", In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pages 90-99, Boston, MA, May 1994.

125. Clifford Mercer, Stefan Savage and Hideyuki Tokuda, "**Processor Capacity Reserves: An Abstraction for Managing Processor Usage**", In *Proceedings of the IEEE Workshop on Workstation Operating Systems*, pages 129-134, Napa, CA, October 1993.

126. Stefan Savage and Hideyuki Tokuda, "**Real-Time Mach Timers: Exporting Time to the User**", In *Proceedings of the USENIX Mach Symposium*, pages 111-118, Santa Fe, NM, April 1993.

Other Papers and Reports

1. Stefan Savage, "**In Planning Digital Defenses, the Biggest Obstacle Is Human Ingenuity**", *New York Times* Science Section, page D10, December 6th, 2011.

2. Ge Wang, He Liu, Sebastian Becerra, Kai Wang, Serge Belongie, Hovav Shacham and Stefan Savage, "**Verilogo: Proactive Phishing Detection via Logo Recognition**", Technical Report CS2011-0969, August 2011.
3. Stefan Savage And Fred B. Schneider, "**Security is Not a Commodity: The Road Forward for Cybersecurity Research**", Computing Research Initiatives for the 21st Century, Computing Community Consortium, February 2009.
4. Yu-Chung Cheng, Mikhail Afanasyev, Patrick Verkaik, Peter Benko, Jennifer Chiang, Alex C. Snoeren, Stefan Savage and Geoffrey M. Voelker, "**The Shaman Automatic 802.11 Wireless Diagnosis System**", *Technical Report CS2010-0959*, Mary 2010.
5. Yuvraj Agarwal, Stefan Savage and Rajesh Gupta, "**SleepServer: Energy Savings for Enterprise PCs By Allowing them to Sleep**", *Technical Report CS2009-0953*, November 2009.
6. Mikhail Afanasyev, Tadayoshi Kohno, Justin Ma, Nick Murphy, Stefan Savage, Alex C. Snoern and Geoffrey M. Voelker, "**Network Support for Privacy-Preserving Forensic Attribution**", *Technical Report CS2009-0940*, March 2009.
7. Qing Zhang, John McCullough, Justin Ma, Nabil Schear, Michael Vrable, Amin Vahdat, Alex C. Snoeren, Geoffrey M. Voelker and Stefan Savage, "**Neon: System Support for Derived Data Management**", *Technical Report CS2008-0934*, December 2008.
8. Michael Vrable, Stefan Savage and Geoffrey M. Voelker, "**Cumulus: Filesystem Backup to the Cloud**", *Technical Report CS2008-0927*, July 2008.
9. Alper Mizrak, Keith Marzullo and Stefan Savage, "**Detecting Compromised Routers via Packet Forwarding Behavior**", *Technical Report CS2007-0899*, June 2007.
10. Chris Fleizach, Geoffrey M. Voelker and Stefan Savage, "**Slicing Spam with Occam's Razor**", *Technical Report CS2007-0893*, June 2007.
11. Alper Mizrak, Keith Marzullo and Stefan Savage, "**Detecting Malicious Packet Losses**", *Technical Report CS2007-0889*, March 2007.
12. David S. Anderson, Chris Fleizach, Stefan Savage and Geoffrey M. Voelker, "**Spamscatter: Characterizing Internet Scam Hosting Infrastructure**", *Technical Report CS2007-0887*, March 2007.
13. Yu-Chung Cheng, Mikhail Afanasyev, Peter Benko, Patrick Verkaik, Jennifer Chiang, Alex C. Snoeren, Stefan Savage, and Geoffrey M. Voelker, "**Automating Cross-Layer Diagnosis of Enterprise Wireless Networks**", *Technical Report CS2007-0885*, March 2007.
14. Justin Ma, Kirill Levchenko, Christian Kreibich, Stefan Savage and Geoff Voelker, "**Automatic Protocol Inference: Unexpected Means of Identifying Protocols**", *Technical Report CS2006-0850*, February 2006.
15. Yu-Chung Cheng, John Bellardo, Peter Benko, Alex Snoeren, Geoff Voelker and Stefan Savage, "**Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis**", *Technical Report CS2006-0849*, February 2006.
16. Sumeet Singh, Cristian Estan, George Varghese and Stefan Savage, "**Accuracy Bounds for the Scaled Bitmap Data Structure",** *Technical Report CS2005-0819*, March 2005.
17. David Moore, Colleen Shannon, Geoffrey M. Voelker and Stefan Savage, "**Network Telescopes",** *Technical Report CS2004-0795*, July 2004.
18. Alper Mizrak, Keith Marzullo and Stefan Savage, "**Fault-Tolerant Forwarding in the Face of Malicious Routers**", In *Proceedings of the International Workshop on Future Directions in Distributed Computing*, Bertinoro, Italy, June 2004. (invited paper)
19. Alper Mizrak, Keith Marzullo and Stefan Savage, "**Detecting Malicious routers",** *Technical Report CS2004-0789*, May 2004.
20. Ishwar Ramani and Stefan Savage, "**SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks**", *UCSD Technical Report CS2004-0787*, May 2004.

21. Fan Chung, Ronald Graham, Ranjita Bhagwan, Stefan Savage and Geoffrey M. Voelker, "**A Near-Optimal Algorithm for a Locality-Maximizing Placement Problems**", *UCSD Technical Report CS2004-0777*, January 2004.

22. Sumeet Singh, Cristian Estan, George Varghese and Stefan Savage, "**Earlybird: Automated Worm Fingerprinting**", In *Proceedings of the ACM Symposium on Operating System Principles*, Bolton Landing, NY, October 2003. (poster, award finalist)

23. Ranjita Bhagwan, Kiran Tati, Yu-Chung Cheng, Stefan Savage and Geoffrey M. Voelker, "**TotalRecall: System Support for Automated Availability Management**", In *Proceedings of the ACM Symposium on Operating System Principles*, Bolton Landing, NY, October 2003. (poster)

24. Sumeet Singh, Cristian Estan, George Varghese and Stefan Savage, "**The Earlybird System for Real-Time Detection of Unknown Worms**", *CS2003-0761*, August 2003

25. Renata Teixeira, Keith Marzullo, Stefan Savage and Geoffrey M. Voelker, "**Characterizing and Measuring Path Diversity of Internet Topologies**", In *Proceedings of the ACM SIGMETRICS Conference*, San Diego, CA, June 2003. (short paper & poster)

26. Abhishek Agrawal, Douglas Brown, Aditya Ojha and Stefan Savage, "Bucking **Free-Riders: Distributed Accounting and Settlement in Peer-to-Peer Networks***", CS2003-0751*, June 2003

27. Cristian Estan, Stefan Savage and George Varghese, "**Automatically Inferring Patterns of Resource Consumption in Network Traffic**", *UCSD Technical Report CS2003-0746*, June 2003

28. Sumeet Singh, Cristian Estan, George Varghese and Stefan Savage, "**Real-time Detection of Known and Unknown Worms**", *UCSD Technical Report CS2003-0745*, May 2003.

29. Flavio Junqueira, Ranjita Bhagwan, Keith Marzullo, Stefan Savage and Geoffrey M. Voelker, "**The Phoenix Recovery System: Rebuilding from the Ashes of an Internet Catastrophe**", *UCSD Technical Report CS2003-0732*, January 2003.

30. Cristian Estan, Stefan Savage and George Varghese, "**Automated Measurement of High-Volume Traffic Clusters**", In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, October, 2002. (short paper)

31. Ranjita Bhagwan, Stefan Savage and Geoffrey M. Voelker, "**Replication Strategies for Highly Available Peer-to-Peer Storage Systems**", *UCSD Technical Report CS2002-0726*, November 2002.

32. Ranjita Bhagwan, David Moore, Stefan Savage and Geoffrey M. Voelker, "**Replication Strategies for Highly Available Peer-to-Peer Storage**", In *Proceedings of the International Workshop on Future Directions in Distributed Computing*, Bertinoro, Italy, June 2002. (invited paper)

33. Stefan Savage, **"Protocol Design in an Uncooperative Internet"**; Ph.D. Thesis, Department of Computer Science and Engineering, University of Washington, January 2002.

34. David Patterson, David Clark, Anna Karlin, Jim Kurose, Edward Lazowska, David Liddle, Derek McAuley, Vern Paxson, Stefan Savage and Ellen Zegura, **Looking Over the Fence at Networks: A Neighbor's View of Networking Research***, National Academies Press*, 2001.

35. Eric J. Anderson, Thomas E. Anderson, Steven D. Gribble, Anna R. Karlin, Stefan Savage, "**A Quantitative Evaluation of Traffic-Aware Routing Strategies",** In *Proceedings of the 2001 ACM SIGCOMM Conference*, San Diego, CA, August 2001. (poster)

36. Stefan Savage, Tom Anderson, Amit Aggarwal, David Becker, Neal Cardwell, Andy Collins, Eric Hoffman, John Snell, Amin Vahdat, Geoffrey M. Voelker and John Zahorjan, "**Detour: a Case for Informed Internet Routing and Transport**", *University of Washington Technical Report UW-CSE-98-10-05*, October 1998.

37. Wilson Hsieh, Marc Fiuczynski, Charles Garrett, Stefan Savage, David Becker and Brian Bershad, "**Language Support for Extensible Operating Systems**", *University of Washington Technical Report UW-CSE-95-11-02*, November 1995.

38. Emin Gun Sirer, Stefan Savage, Przemyslaw Pardyak, Greg Defouw, Mary Ann Alapat and Brian Bershad, "**Writing an Operating System with Modula-3**", *University of Washington Technical Report UW-CSE-95-11-03*, November 1995.

39. Brian Bershad, Craig Chambers, Susan Eggers, Chris Maeda, Dylan McNamee, Przemyslaw Pardyak, Stefan Savage and Emin Gun Sirer, "**SPIN – An Extensible Microkernel for Application-specific Operating System Services**", *University of Washington Technical Report UW-CSE-94-03-03*, March 1994.

40. Clifford Mercer, Stefan Savage and Hideyuki Tokuda, "**Processor Capacity Reserves for Multimedia Operating Systems**", *Carnegie Mellon Technical Report, CMU-CS-93-157*, May 1993.

Invited Presentations

"Cyberphysical System Security", Invited talk, ISAT Study on Bio-Cyber Security, 2020.

"Experiences in Tracking Cybercrime", Invited talk, Emerging Tech and Partnerships for Counterproliferation Workshop, National Academies of Science, 2020.

"Cybersecurity as an evidence-based discipline", Keynote, NSF Cybersecurity Summit, 2019.

"Modern Automotive Vulnerabilities: Causes, Disclosures and Outcomes", Distinguished Lecture, University of Massachusetts, Amherst, 2019.

"Modern Automotive Vulnerabilities: Causes, Disclosures and Outcomes", Distinguished Lecture, University of Massachusetts, Amherst, 2019.

"Automotive Cybersecurity: Research driving Practice", San Diego Chapter Computer Science Teachers Association, 2019.

"Modern Automotive Vulnerabilities: Causes, Disclosures and Outcomes", Distinguished Lecture, University of California, Riverside, 2019.

"Lawful Device Access without Mass Surveillance Risk", Qualcomm Invited Lecture, 2018.

"Lawful Device Access without Mass Surveillance Risk", Cornell Tech Distinguished Lecture, 2018.

"**Cybersecurity in the Age of the Internet**", Fleet Science Center "Suds & Science" series, 2018.

"**Driving Under the Influence: Cybersecurity of Modern Automobiles**", Invited Talk, Bohemian Club, 2018.

"**Modern Automotive vulnerabilities: Causes, Disclosures and Outcomes"**, Invited Talk, UCSD Oshur Center, 2018.

"**Modern Automotive vulnerabilities: Problems, Causes and Outcomes**", Qualcomm Mobile Security Summit, 2018.

"**Ethical Issues in Cybersecurity Research**", Design@Large UCSD Lecture series, 2018.

"**Modern Automotive vulnerabilities: Problems, Causes and Outcomes**", NSF Distinguished Lecture, 2018.

"**Fighting cybercrime by Understanding Economics**", UCSD Town and Gown, 2018.

"**Experiences with Automotive Security: Vulnerabilities, Challenges and Outcomes**", JASON Autonomous Driving study, 2017.

"**Experiences with Automotive Security: Vulnerabilities, Challenges and Outcomes**", Microsoft Research, 2016.

"**Understanding the Effects of Security Interventions**", Microsoft Research, 2016.

"**Modern Automotive Vulnerabilities: Causes, Disclosures and Outcomes**", USENIX Enigma Conference, 2016.

"**Abuse Economics: Spam, Drugs, Affiliate Programs Payments and Interventions**", Google Security Summit, 2015.

"**Experiences for Automotive Security: Vulnerabilities, Causes and Challenges**", Stanford Cyber BootCamp for Congressional Staffers, 2015.

"**Data Driven Vulnerability Assessment**", CMU/SwissRe Workshop on Cyber Risk, 2015.

"**Demonetizing Advertising Based eCrime**", Michigan State University Interdisciplinary Conference on Cybercrime, 2015.

"**Cybersecurity: Challenges and Opportunities**", National Research Council Symposium on Continuing Innovation in Information Technology, 2014.

"**On the Effects of Civil Interventions**", Interpol/Cymru Underground Economy 2014,

"**Experiences from ten years of data-driven security research**", UW/MSR Security Analytics Workshop, 2014.

"**Abuse Economics: Spam, Drugs, Affiliate Programs, Payments and Interventions**", Google Abuse Summit, Mountain View, 2014.

"**Demonetizing advertising-based eCrime**", Microsoft Research (NYC), 2014.

"**Automotive Cyber Threats**", ITSG, 2014.

"**Buying Drugs for Science: Addressing the Cybercrime through Economics**", Andreesen Horowitz Academic Roundtable, 2013.

"**Buying Drugs for Science: Addressing the Economics of Cybercrime**", UCSD Gordon Center Think Tank, 2013.

"**Bitcoin Abuse: from Botnets to Payments**", Interpol/Cymru Underground Economy, 2013.

"**Experiences with Automotive Security: Vulnerabilities, Causes and Challenges**", IEEE NTS (Boeing) Invited Talk, 2013.

"**Buying Drugs for Science: Addressing the Economics of Cybercrime**", Osher Lifelong Learning Institute, 2013.

"**Buying Drugs for Science: Addressing the Economics of Cybercrime**", Microsoft Faculty Summit, 2013.

"**Experiences with Automotive Security: Vulnerabilities, Causes and Challenges**", ESCAR USA, 2013.

"**Buying Drugs for Science: Addressing the Economics of Cybercrime**", Department of Justice Invited Talk, 2013.

"**The Role of Payments in Monetizing Abusive Advertising**", Microsoft Digital Crime Conference, 2013.

"**Experiences with Automotive Security: Vulnerabilities, Causes and Challenges**", DARPA HACMS PI Meeting, 2013.

"**Experiences with Automotive Security: Vulnerabilities, Causes and Challenges**", NSF Future Directions in Formal Methods keynote, 2013.

"**Reflecting on Automotive Cybervulnerabilities**", DoD CyberVehicle Security Workshop, 2012.

"**Buying Drugs for Science: Addressing Cybercrime from an Economic Perspective**", Yahoo Big Thinker Talk, 2012.

"**Payment Processing for Advertising-based E-crime**", Interpol/Cymru Underground Economy, 2012.

"**Buying Drugs for Science: Addressing Cybercrime from an Economic Perspective**", Carnegie Mellon, 2012.

"**Inside the Economics of Illicit Online Pharma: Demand, Marketing and Payments**", NCFTA Pharma Conference, 2012.

"**Buying Drugs for Science: Understanding the Economics of Cybercrime"**, Facebook, 2012.

"**Experimental Security Analysis of a Modern Automobile**", JASON invited talk, 2012.

"**Looking before you Leap: the argument for data-driven security**", Distinguished Lecture, University of California, Berkeley, 2012.

"**Successes and Challenges of Computer Security Research**", CCC's "The Impact of NITRD" Symposium, 2012.

"**Looking before you Leap: the argument for data-driven security**", Distinguished Lecture, University of Illinois 2012.

"**Looking before you Leap: the argument for data-driven security**", Distinguished Lecture, Stanford University 2012.

"**Why the hard problem of computer security needs the soft sciences**", Invited talk, NSF Washington Area Trustworthy Computing Hour (WATCH), 2011.

"**The Spam Ecosystem: Demand, Revenue and Payments**", Trend Micro, 2011.

"**Monetizing the botnet ecosystem**", Panel session, Microsoft Digital Crime Consortium (DCC), 2011

"**The Spam Ecosystem: Demand, Revenue and Payments**", Interpol/Cymru Underground Economy, 2011.

"**Looking before you leap: an argument for data-driven security**", Distinguished Lecture, University of Maryland, 2011.

"**Looking before you leap: an argument for data-driven security**", MIT 2011.

"**Experimental Security Analysis of a Modern Automobile**", Google, 2011.

"**Follow the Money: Understanding the Spam Value Chain**", UCSD Campus LISA, 2011.

"**Experimental Security Analysis of a Modern Automobile**", Distinguished Lecture, University of British Columbia, 2011.

"**Beyond Heuristics: Learning to Classify Vulnerabilities and Predict Exploits**", HP Labs, 2011.

"**Analyzing and Understanding Online Criminals and Behavior** ", In-Q-Tel, 2011.

"**Experimental Security Analysis of a Modern Automobile**", In-Q-Tel, 2011. (talk given w/Yoshi Kohno)

"**The Value Chain of Spam-advertised Goods"**, Galois Systems, 2011.

"**An Agenda for Empirical Cybercrime Research**", Keynote USENIX Annual Technical Conference, 2011.

"**Experimental Security Analysis of a Modern Automobile**", Qualcomm Security Symposium, 2011.

"**The Value Chain of Spam-advertised Pharmacies"**, US Food and Drug Administration, 2011.

"**Experimental Security Analysis of a Modern Automobile**", National Academy Science, Study group on Unintended Acceleration, 2011. (talk given w/Yoshi Kohno)

"**Security and Criminal Economics**", ISAT Study group on Security Economics, 2011.

"**An Agenda for Empirical Cybercrime Research**", Google, 2011.

"**Drug Dealing for Science and Other Stories…**", Duke University, 2010.

"**Understanding Modern Cybercrime: Technology and Economics**", BAE Systems, 2010.

"**An Agenda for Empirical Cybercrime Research**", Microsoft, 2010.

"**Defining the Threat: Technical/Commercial Drivers**", Lawrence Livermore National Lab, 2010.

"**Beyond Heuristics: Learning to Classify Vulnerabilities and Predict Exploits"**, invited talk USENIX Workshop On Offensive Technology (WOOT), 2010.

"**Re: CAPTCHAs: Understanding CAPTCHA Solving from an Economic Context**", Google, 2010.

"**Computer Security: Science, Engineering, Economics and a bit of Reality**", JASON study group, 2010.

"**Cybereconomics**", Cyber Leap Year panel at IEEE Security and Privacy, 2010.

"**The Underground Economy: A broad overview**", Invited Talk, INFOSEC Research Council, 2010.

"**The Security Arms Race: Yesterday, Today and Tomorrow**", Invited talk, CONNECT Research and Innovation Summit, 2010.

"**Re: CAPTCHAs: Understanding CAPTCHA Solving from an Economic Context**", Yahoo, 2010.

"**Exploring the Economics of Cybercrime**", National Defense Industry Association Cyber Symposium, 2009.

"**Exploring Internet Crime, Economics and Security**", Invited talk, CONNECT Frontiers in Science and Technology Seminar Series, 2009.

"**Internet Criminal Economics**", Invited talk, La Jolla Conference on Innovation Support for National Security, 2009.

"**Internet Outbreaks: Epidemiology and Defenses**", Invited talk, ViaSAT/NSA Information Assurance Summit, Carlsbad, CA 2009

Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams", Invited Talk, Harvard University, 2009.

Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams", Distinguished Lecture, University of Pennsylvania, 2009.

Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams", Invited Talk, Mail Anti-Abuse Working Group meeting, 2009.

"**Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams**", Invited Talk, University of Washington, 2008.

"**Reconsidering Physical Key Secrecy: Teleduplication via Optical Decoding**", Invited Talk, University of Washington, 2008.

"**Issues of Security and Risk for Third-Party Cloud Computing**", Cisco Cloud Computing Research Symposium, 2008.

"**Exploring Internet Criminal Economics**", Distinguished Lecture, University of Virginia, 2008.

"**Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams**", Invited Talk, Yahoo!, Sunnyvale, CA, 2008.

"**Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams**", Invited Talk, Microsoft Research, Mountain View, CA, 2008.

"**Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams**", Invited Talk, Campus LISA, UCSD, 2008.

"**Exploring the Underground Economy**", Invited Lecture at University of Chicago, 2008.

"**Understanding the Growth of the Underground Economy"**, Distinguished Lecture at Dartmouth College, 2008.

"**Underground Economics**", Keynote speaker at Anti-Spyware Coalition Workshop, 2008.

"**Underground Economics**", Santa Fe Institution Malware Workshop, 2007.

"**Understanding and Addressing the Threat of Internet Worms**" (joint w/Vern Paxson, Geoff Voelker and Nick Weaver), Day-long tutorial at USENIX Security Symposium, Vancouver, B.C., 2006.

"**Large-scale Internet Threats: Epidemiology and Defenses**", Keynote talk, World Conference on Structural Control and Monitoring, San Diego, CA, 2006.

"**Scalable Internet Threat Monitoring**", Hewlett-Packard Laboratories, Palo Alto, CA 2006.

"**Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis**", Google, Mountain View, CA, 2006.

"**Active Research in Addressing Unwanted Traffic**", Internet Architecture Board review, Marina del Rey, CA, 2006.

"**Scalable Internet Threat Monitoring**", UCSD Jacobs School of Engineering Research Expo, San Diego, CA, 2006.

"**Scalable Internet Threat Monitoring**", University of Toronto Distinguished Lecture, Toronto, 2005.

"**Scalable Internet Threat Monitoring**", Intel Research Council Workshop on Network Security, Hillsboro, OR, 2005.

"**Scalable Internet Threat Monitoring**", ACM Workshop on Rapid Malcode, Fairfax, VA, 2005.

"**Scalable Internet Threat Monitoring**", ARO/DHS Workshop on Malware 2005, Washington, D.C., 2005.

"**Internet Outbreaks: Epidemiology and Defenses**", Cornell University, Ithaca, NY 2005.

"**Internet Outbreaks: Epidemiology and Defenses**", Jacobs School of Engineering, Corporate Affiliates Program Board Meeting, San Diego, CA 2005.

"**Internet Outbreaks: Epidemiology and Defenses**", Network and Distributed System Security Symposium, San Diego, CA 2005.

"**Internet Outbreaks: Epidemiology and Defenses**", University of Washington, Seattle, WA 2005.

"**Low-cost 802.11 Wireless Infrastructure Networks**", Microsoft Mesh Networking Summit, Snoqualmie, WA, 2004.

"**The UCSD Network Telescope: A Real-time Monitoring System for Tracking Internet**", UCSD Jacobs School of Engineering Research Review Plenary Session, San Diego, CA, 2004..

"**Network Worms: Research and Engineering Challenges**", IPAM Large Scale Communication Networks Workshop, Lake Arrowhead, CA, 2003.

"**Network Worms: Research and Engineering Challenges**", Microsoft Security Retreat, Stevenson, WA, 2003.

"**Network Worms: Research and Engineering Challenges**", Intel, Hillsboro, OR, 2003.

"**Defending against Network Worms**", Hewlett-Packard Labs, Palo Alto, CA, 2003 (cancelled).

"**Denying Denial-of-Service**", Networld+Interop, Atlanta, GA, 2001.

"**DDoS Attacks: Why We Should Still Worry**", The Internet Security Conference, Los Angeles, CA, 2001.

"**Estimating Global Denial-of-Service Activity**", North American Network Operators Group, Scottsdale, AZ, 2001.

"**Internet Denial-of-Service Activity**", Stanford, Department of Computer Science, Palo Alto, CA, 2001.

"IP Traceback Techniques", North American Network Operators Group, Albuquerque, NM, 2000.

"**Sting: a TCP-based Network Measurement Tool**", CMU School of Computer Science, Pittsburgh, PA, 1999.

Awarded U.S. Patents

Thomas Anderson, David Wetherall and Stefan Savage, "**Detecting network misuse**", *US patent 8,509.086, Filed June 20, 2002, Issued August 13th, 2013.*

Sumeet Singh, George Varghese, Cristian Estan and Stefan Savage, "**Detecting public network attacks using signatures and fast content analysis**", *US patent 8,296,842, Filed Dec 1st 2004, Issued October 23rd, 2012.*

David Wetherall, Stefan Savage and Thomas Anderson, "**Independent detection and filtering of undesirable packets**", *US patent 8,271,678, Filed April 3rd 2001, Issued September 18t, 2012.*

Thomas Anderson, David Wetherall and Stefan Savage, "**Detecting and preventing undesired network traffic from being sourced out of a network domain**", *US patent 7970886, Filed November 2nd 2000, Issued June 28th 2011.*

Sumeet Singh, George Varghese, Cristian Estan and Stefan Savage, "**Detecting public network attacks using signatures and fast content analysis**", *US patent 7996658, Filed April 8th 2004, Issued June 21st 2011.*

David Wetherall, Stefan Savage and Thomas Anderson, "**Distributed service level management for network traffic**", *US patent 7475141, Filed July 31s 2001, Issued January 6th 2009.*

Thomas Anderson, Stefan Savage and David Wetherall, "**Network traffic regulation including consistency based detection and filtering of packets with spoof source addresses**", *US patent 7444404, Filed February 5th, 2001, Issued October 28th, 2008.*

David Wetherall, Thomas Anderson, and Stefan Savage, "**Distributed solution for regulating network traffic**", *US patent 7058015*, Filed August 4th 2000, Issued June 6th, 2006.

Thomas Anderson, David Wetherall and Stefan Savage, "**Progressive and distributed regulation of selected network traffic destined for a network node**", *US patent 6801503*, Filed October 9th, 2000, Issued October 5th 2004.

Michael Burrows, Charles G. Nelson, Stefan Savage and Patrick Sobalvarro, "**Detecting concurrency errors in multi-threaded programs**", *US patent 6009269*, Filed March 10th 1997, Issued December 28th, 1999.

John Wilkes and Stefan Savage, "**Frequently-redundant array of independent disks**", *US patent 5720025*, Filed January 18th 1996, Issued February 17th 1998.

Software & Data

WHOIS parser and parsed .com records. Worlds most accurate parser of WHOIS records and easily updatable via labelling (machine learning-based). Also, distributed parsed records for 95% of .com domain names. Code freely available under GPL, data selectively available on request.

BlueSky Cloud-based Log-Structured File System. Code freely available under GPL.

Click Trajectory database. Multi-terabyte dataset mapping registrars, name server hosting, content hosting, sponsoring businesses and acquiring bank data for virtually all counterfeit pharmaceutical, replica and software spam. Not publicly available, but particular records shared with researchers and law enforcement on request.

Malicious URL classifier. Online machine learning system for classifying Web URLs as likely pointing to unwanted sites (e.g., spam-advertised, phishing, malware, etc.) Selectively available.

Judo spam template inference system. System for automatically inferring regular expression from concentrated spam output from a given botnet. Selectively available.

Honeyfarm spam data. Spam output generated from series of botnets operated in ICSI/UCSD botnet studies. Selectively available.

Cumulus backup system. Cloud-based (Amazon S3) backup system optimized for simple storage interface with capacity pricing. Freely available.

Storm botnet crawler and command and control database. Crawler for querying proxy nodes in Storm peer-to-per botnet and populate database. Also many months of complete data on Storm-based spam campaigns based on this data. Made available selectively (primarily to key industrial partners and law enforcement).

UCSD CSE 802.11 traces. Anonymized traces of complete 802.11activity in the UCSE CSE building (across all four floors and all active frequencies) as captured by Jigsaw. Freely available.

Jigsaw Wireless Analysis Tool. System to synchronize and unify hundreds of simultaneous radio traces into a single global view of wireless activity spanning the physical, datalink, network and transport layers. Freely available under GPL.

Potemkin VMM. Copy-on-write extensions to Xen Virtual Machine Monitor – allows highly scalable implementation of virtual machines that share state. Freely available under GPL.

SyncScan. Driver and system modifications to allow wireless 802.11 clients to continuously monitor the signal strength from different access points without disrupting data communication. Freely available to academic and research sites.

Earlybird. Network traffic analysis system that automatically identifies and extracts textual signatures for novel worms and viruses. Exclusively licensed to Netsift Inc -> Cisco Inc.

AutoFocus. Traffic analysis system that converts Cisco Netflow data from routers into concise lists of high-volume traffic clusters according to algorithms in 2003 ACM SIGCOMM paper. Freely available to academic and research sites.

P2P Host Availability Dataset. Week-long active measurement trace of individual host availability for 2400 random hosts deployed Overnet P2P file sharing system. Particularly useful because Overnet maintains globally unique id's and data isn't subject to aliasing. Distributed widely to academic sites and industrial research labs.

Swat 802.11 Vulnerability Testing Tool. Low-level library and interface to sending arbitrary 802.11 frames using undocumented features of Intersil firmware interface. Distribution limited.

DoS Backscatter Dataset. Packet-level header trace from UCSD Network Telescope identifying global denial-of-service attacks utilizing spoofed packet floods via "backscatter analysis". Distributed widely to academic sites and government research labs – the only dataset of its kind publicly available.

Sting. A network measurement tool that estimates one-way path information, such as drop rate, re-ordering, queuing delay, and bandwidth, by manipulating the behavior of the popular TCP protocol. The alpha release, for FreeBSD and Linux, has been distributed to more than 500 sites, including a product effort at Hewlett-Packard/Agilent. Publicly available.

Eraser. A dynamic race detection tool that automatically detects errors in multi-threaded software using binary rewriting on the Alpha platform. Owned by DEC/Compaq/Hewlett Packard. This tool led to DEC/Compaq/HP's *Visual Threads* software debugging package.

SPIN. An extensible operating system written in Modula-3 for Alpha and Intel platforms. Publicly available.

Real-Time Mach. A real-time extension of the Mach operating system with support for multiple real-time scheduling and synchronization policies and a framework to support CPU reservation and priority inheritance. Publicly available.

ATTACHMENT C

Publications

2021 | 2020 | 2019 | 2018 | 2017 | 2016 | 2015 | 2014 | 2013 | 2012 | 2011 | 2010 | 2009 | 2008
2007 | 2006 | 2005 | 2004 | 2003 | 2002 | 2001 | 2000 | 1999 | 1997 | 1996 | 1995 | 1994 | 1993

2021

Jetset: Targeted Firmware Rehosting for Embedded Systems, Evan Johnson, Maxwell Bland, Yifei Zhu, Joshua Mason, Stephen Checkoway, Stefan Savage, and Kirill Levchenko, *Proceedings of the USENIX Security Symposium*, Vancouver, B.C., Canada, August 2021.

Доверяй, но проверяй: SFI safety for native-compiled Wasm, Evan Johnson, David Thien, Yousef Alhessi, Shravan Narayan, Fraser Brown, Sorin Lerner, Tyler McMullen, Stefan Savage, and Deian Stefan, *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2021.

2020

Unresolved Issues: Prevalence, Persistence and Perils of Lame Nameservers, Gautam Akiwate, Raffaele Sommese, Mattijs Jonker, Ian Foster, Stefan Savage, Geoffrey M. Voelker, and kc Claffy, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Pittsburgh, Pennsylvania (via the Internet), October 2020.

Trufflehunter: Cache Sniffing Rare Domains at Large Public DNS Resolvers, Audrey Randall, Enze Liu, Gautam Akiwate, Ramakrishna Padmanabhan, Stefan Savage, Geoffrey M. Voelker, and Aaron Schulman, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Pittsburgh, Pennsylvania (via the Internet), October 2020.

2019

Measuring Security Practices and How They Impact Security, Louis F. DeKoven, Audrey Randall, Ariana Mirian, Gautam Akiwate, Ansel Blume, Lawrence K. Saul, Aaron Schulman, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Amsterdam, Netherlands, October 2019, pages 36-49.

Triton: A Software-Reconfigurable Federated Avionics Testbed, Sam Crow, Brown Farinholt, Brian Johannesmeyer, Karl Koscher, Stephen Checkoway, Stefan Savage, Aaron Schulman, Alex C. Snoeren, and Kirill Levchenko, *Proceedings of Workshop on Cyber Security Experimentation and Test (CSET)*, Santa Clara, CA, August 2019.

Detecting and Characterizing Lateral Phishing at Scale, Grant Ho, Asaf Cidon, Lior Gavish, Marco Schweighauser, Vern Paxson, Stefan Savage, Geoffrey M. Voelker, and David Wagner, *Proceedings of the USENIX Security Symposium*, Santa Clara, CA, August 2019, pages 1273-1290. (**Award paper**).

Reading the Tea Leaves: A Comparative Analysis of Threat Intelligence, Vector Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M. Voelker, Stefan Savage, and Kirill Levchenko, *Proceedings of the USENIX Security Symposium*, Santa Clara, CA, August 2019, pages 851-867.

Web Feature Deprecation: A Case Study for Chrome, Ariana Mirian, Nikunj Bhagat, Caitlin Sadowski, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the 41st ACM/IEEE International Conference on Software Engineering: Software Engineering in Practice, ICSE (SEIP) 2019*, Montréal, QC, Canada, May 2019, pages 302-311.

Cloudflare, Inc. Exhibit 1003 Page 1138

Hack for Hire: Exploring the Emerging Market for Account Hijacking, Ariana Mirian, Joe DeBlasio, Stefan Savage, Geoffrey M. Voelker, and Kurt Thomas, *Proceedings of the Web Conference (WWW)*, San Francisco, CA, May 2019, pages 1279-1289.

2018

Following Their Footsteps: Characterizing Account Automation Abuse and Defenses, Louis F. DeKoven, Trevor Pottinger, Stefan Savage, Geoffrey M. Voelker, and Nektarios Leontiadis, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Boston, MA, October 2018, pages 43-55.

Lawful Device Access without Mass Surveillance Risk: A Technical Design Discussion, Stefan Savage, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Toronto, Canada, October 2018.

2017

Quantifying the Pressure of Legal Risks on Third-party Vulnerability Research, Alex Gamero-Garrido, Stefan Savage, Kirill Levchenko, and Alex C. Snoeren, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Dallas, Texas, November 2017.

Tripwire: Inferring Internet Site Compromise, Joe DeBlasio, Stefan Savage, Geoffrey M. Voelker, and Alex C. Snoeren, *Proceedings of the ACM Internet Measurement Conference (IMC)*, London, UK, November 2017, pages 341-354.

Malicious Browser Extensions at Scale: Bridging the Observability Gap between Web Site and Browser, Louis F. DeKoven, Stefan Savage, Geoffrey M. Voelker, and Nektarios Leontiadis, *Proceedings of Workshop on Cyber Security Experimentation and Test (CSET)*, August 2017.

How They Did It: An Analysis of Emission Defeat Devices in Modern Automobiles, Mortiz Contag, Guo Li, Andre Pawlowski, Felix Domke, Stefan Savage, Kirill Levchenko, and Thorsten Holz, *Proceedings of the IEEE Symposium on Security and Privacy*, San Jose, CA, May 2017.

A Call to ARMs: Understanding the Costs and Benefits of JIT Spraying Mitigations, Wilson Lian, Hovav Shacham, and Stefan Savage, *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2017.

2016

You've Got Vulnerability: Exploring Effective Vulnerability Notifications, Frank Li, Zakir Durumeric, Jakub Czyz, Damon McCoy, Stefan Savage, Michael Bailey, and Vern Paxson, *Proceedings of the USENIX Security Symposium*, Austin, TX, August 2016.

2015

Scheduling Techniques for Hybrid Circuit/Packet Networks, He Liu, Matthew K. Mukerjee, Conglong Li, Nicolas Feltman, George Papen, Stefan Savage, Srinivasan Seshan, Geoffrey M. Voelker, David G. Andersen, Michael Kaminsky, George Porter, and Alex C. Snoeren, *Proceedings of ACM CoNEXT*, Heidelberg, Germany, December 2015.

Security by Any Other Name: On the Effectiveness of Provider Based Email Security, Ian Foster, Jon Larson, Max Masich, Alex C. Snoeren, Stefan Savage, and Kirill Levchenko, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Denver, Colorado, October 2015.

Cloudflare, Inc. Exhibit 1003 Page 1139

From .academy to .zone: An Analysis of the New TLD Land Rush, Tristan Halvorson, Matthew F. Der, Ian Foster, Stefan Savage, Lawrence K. Saul, and Geoffrey M. Voelker, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Tokyo, Japan, October 2015.

Who is .com? Learning to Parse WHOIS Records, Suqi Liu, Ian Foster, Stefan Savage, Geoffrey M. Voelker, and Lawrence K. Saul, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Tokyo, Japan, October 2015.

Affiliate Crookies: Characterizing Affiliate Marketing Abuse, Neha Chachra, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Tokyo, Japan, October 2015.

Fast and Vulnerable: A Story of Telematic Failures, Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage, *Proceedings of the USENIX Workshop On Offensive Technologies (WOOT)*, Washington D.C., August 2015.

Framing Dependencies Introduced by Underground Commoditization, Kurt Thomas, Danny Yuxing Huang, David Wang, Elie Bursztein, Chris Grier, Tom Holt, Christopher Kruegel, Damon McCoy, Stefan Savage, and Giovanni Vigna, *Proceedings of the Workshop on the Economics of Information Security (WEIS)*, Delft, The Netherlands, June 2015.

Too LeJIT to Quit: Extending JIT Spraying to ARM, Wilson Lian, Hovav Shacham, and Stefan Savage, *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2015.

2014

On The Security of Mobile Cockpit Information Systems, Devin Lundberg, Brown Farinholt, Edward Sullivan, Ryan Mast, Stephen Checkoway, Stefan Savage, Alex C. Snoeren, and Kirill Levchenko, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Scottsdale, Arizona, November 2014.

Characterizing Large-Scale Click Fraud in ZeroAccess, Paul Pearce, Vacha Dave, Chris Grier, Kirill Levchenko, Saikat Guha, Damon McCoy, Vern Paxson, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Scottsdale, Arizona, November 2014.

Search + Seizure: The Effectiveness of Interventions on SEO Campaigns, David Wang, Matthew Der, Mohammad Karami, Lawrence Saul, Damon McCoy, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Vancouver, BC, Canada, November 2014, pages 359-372.

Handcrafted Fraud and Extortion: Manual Account Hijacking in the Wild, Elie Bursztein, Borbala Benko, Daniel Margolis, Tadek Pietraszek, Andy Archer, Allan Aquino, Andreas Pitsillidis, and Stefan Savage, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Vancouver, BC, Canada, November 2014.

Knock It Off: Profiling the Online Storefronts of Counterfeit Merchandise, Matthew Der, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the ACM SIGKDD Conference*, New York, NY, August 2014.

Empirically Characterizing Domain Abuse and the Revenue Impact of Blacklisting, Neha Chachra, Damon McCoy, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the Workshop on the Economics of Information Security (WEIS)*, State College, PA, June 2014, pages 4:1:1-4:1:13.

XXXtortion? Inferring Registration Intent in the .XXX TLD, Tristan Halvorson, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the International World Wide Web Conference (WWW)`*, Seoul, Korea, April 2014, pages 901-912.

Botcoin: Monetizing Stolen Cycles, Danny Yuxing Huang, Hitesh Dharmdasani, Sarah Meiklejohn, Vacha Dave, Chris Grier, Damon McCoy, Stefan Savage, Nicholas Weaver, Alex C. Snoeren, and Kirill Levchenko, *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2014.

Cloudflare, Inc. Exhibit 1003 Page 1140

2013

A Fistful of Bitcoins: Characterizing Payments Among Men with No Names, Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage, *USENIX ;login:* 38(6):10-14, December 2013.

A Comparison of Syslog and IS-IS for Monitoring Link State, Daniel Turner, Kirill Levchenko, Stefan Savage, and Alex C. Snoeren, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Barcelona, Spain, October 2013.

A Fistful of Bitcoins: Characterizing Payments Among Men with No Names, Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Barcelona, Spain, October 2013, pages 127-140.

Measuring the Practical Impact of DNSSEC Deployment, Wilson Lian, Eric Rescorla, Hovav Shacham, and Stefan Savage, *Proceedings of the USENIX Security Symposium*, Washington D.C., August 2013.

Juice: A Longitudinal Study of an SEO Campaign, David Wang, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2013, pages 7:4:1-7:4:17.

2012

Taster's Choice: A Comparative Analysis of Spam Feeds, Andreas Pitsillidis, Chris Kanich, Geoffrey M. Voelker, Kirill Levchenko, and Stefan Savage, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Boston, MA, November 2012, pages 427-440.

Priceless: The Role of Payments in Abuse-advertised Goods, Damon McCoy, Hitesh Dharmdasani, Christian Kreibich, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Raleigh, NC, October 2012, pages 845-856.

Manufacturing Compromise: The Emergence of Exploit-as-a-Service, Chris Grier, Lucas Ballard, Juan Caballero, Neha Chachra, Christian J. Dietrich, Kirill Levchenko, Panayiotis Mavrommatis, Damon McCoy, Antonio Nappa, Andreas Pitsillidis, Niels Provos, M. Zubair Rafique, Moheeb Abu Rajab, Christian Rossow, Kurt Thomas, Vern Paxson, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Raleigh, NC, October 2012, pages 821-832.

Be Conservative: Enhancing Failure Diagnosis with Proactive Logging, Ding Yuan, Soyeon Park, Peng Huang, Yang Liu, Michael Lee, Xiaoming Tang, Yuanyuan Zhou, and Stefan Savage, *Proceedings of the 10th ACM/USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, Hollywood, CA, October 2012.

PharmaLeaks: Understanding the Business of Online Pharmaceutical Affiliate Programs, Damon McCoy, Andreas Pitsillidis, Grant Jordan, Nicholas Weaver, Christian Kreibich, Brian Krebs, Geoffrey M. Voelker, Stefan Savage, and Kirill Levchenko, *Proceedings of the USENIX Security Symposium*, Bellevue, WA, August 2012.

When Good Services Go Wild: Reassembling Web Services for Unintended Purposes, Feng Lu, Jiaqi Zhang, and Stefan Savage, *Proceedings of the USENIX Workshop on Hot Topics in Security*, Bellevue, WA, August 2012.

scc: Cluster Storage Provisioning Informed by Application Characteristics and SLAs, Harsha V. Madhyastha, John C. McCullough, George Porter, Rishi Kapoor, Stefan Savage, Alex C. Snoeren, and Amin Vahdat, *USENIX ;login:* 37(3), June 2012.

Measuring the Cost of Cybercrime, Ross Anderson, Chris Barton, Rainer Boehme, Richard Clayton, Michel J.G. van Eeten, Michael Levi, Tyler Moore, and Stefan Savage, *Proceedings of the Workshop on the Economics of Information Security (WEIS)*, Berlin, Germany, June 2012.

Cloudflare, Inc. Exhibit 1003 Page 1141

Return-Oriented Programming: Systems, Languages and Applications, Ryan Roemer, Erik Buchanan, Hovav Shacham, and Stefan Savage, *ACM Transactions on Information and System Security* 15(1), March 2012.

Improving Software Diagnosability via Log Enhancement, Ding Yuan, Jing Zheng, Soyeon Park, Yuanyuan Zhou, and Stefan Savage, *ACM Transactions on Computer Systems* 30(1), February 2012.

BlueSky: A Cloud-Backed File System for the Enterprise, Michael Vrable, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the 10th USENIX Conference on File and Storage Technologies (FAST)*, San Jose, CA, February 2012, pages 19:1-19:14.

scc: Cluster Storage Provisioning Informed by Application Characteristics and SLAs, Harsha V. Madhyastha, John C. McCullough, George Porter, Rishi Kapoor, Stefan Savage, Alex C. Snoeren, and Amin Vahdat, *Proceedings of the 10th USENIX Conference on File and Storage Technologies (FAST)*, San Jose, CA, February 2012.

2011

In Planning Digital Defenses, the Biggest Obstacle is Human Ingenuity, Stefan Savage, *New York Times*, Dec 6 2011.

An Analysis of Underground Forums, Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Berlin, Germany, November 2011, pages 71-80.

Judging a site by its content: learning the textual, structural, and visual features of malicious Web pages, Sushma Nagesh Bannur, Lawrence K. Saul, and Stefan Savage, *Proceedings of the ACM Workshop on Artificial Intelligence and Security (AISEC)*, Chicago, IL, October 2011.

Cloak and Dagger: Dynamics of Web Search Cloaking, David Wang, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Chicago, IL, October 2011, pages 477-490.

On the Empirical Performance of Self-calibrating WiFi Location Systems, Daniel Turner, Stefan Savage, and Alex C. Snoeren, *Proceedings of IEEE Conference on Local Computer Networks (LCN)*, Bonn, Germany, October 2011.

Interview with Stefan Savage: On the Spam Payment Trail, Rik Farrow and Stefan Savage, *USENIX ;login:* 36(4):7-20, August 2011.

Heat of the Moment: Characterizing the Efficacy of Thermal Camera-Based Attacks, Keaton Mowery, Sarah Meiklejohn, and Stefan Savage, *Proceedings of the USENIX Workshop On Offensive Technologies (WOOT)*, San Francisco, CA, August 2011.

No Plan Survives Contact: Experience with Cybercrime Measurement, Chris Kanich, Neha Chachra, Damon McCoy, Chris Grier, David Wang, Marti Motoyama, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of Workshop on Cyber Security Experimentation and Test (CSET)*, August 2011, pages 2:1-2:8.

Show Me the Money: Characterizing Spam-advertised Revenue, Chris Kanich, Nicholas Weaver, Damon McCoy, Tristan Halvorson, Christian Kreibich, Kirill Levchenko, Vern Paxson, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the USENIX Security Symposium*, San Francisco, CA, August 2011, pages 219-234.

Dirty Jobs: The Role of Freelance Labor in Web Service Abuse, Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the USENIX Security Symposium*, San Francisco, CA, August 2011, pages 203-218.

Comprehensive Experimental Analyses of Automotive Attack Surfaces, Stephen Checkoway, Damon McCoy, Danny Anderson, Brian Kantor, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno, *Proceedings of the USENIX Security Symposium*, San Francisco, CA, August 2011.

DefenestraTor: Throwing out Windows in Tor, Mashael AlSabah, Kevin Bauer, Ian Goldberg, Dirk Grunwald, Damon McCoy, Stefan Savage, and Geoffrey M. Voelker, *Privacy Enhancing Technologies Symposium*, Waterloo, Canada, July 2011, pages 134-154.

Cloudflare, Inc. Exhibit 1003 Page 1142

Privacy-preserving Network Forensics, Mikhail Afanasyev, Tadayoshi Kohno, Justin Ma, Nick Murphy, Stefan Savage, Alex C. Snoeren, and Geoffrey M. Voelker, *Communications of the Association for Computing Machinery* 54(5):78-87, May 2011.

Click Trajectories: End-to-End Analysis of the Spam Value Chain, Kirill Levchenko, Andreas Pitsillidis, Neha Chachra, Brandon Enright, Márk Félegyházi, Chris Grier, Tristan Halvorson, Chris Kanich, Christian Kreibich, He Liu, Damon McCoy, Nicholas Weaver, Vern Paxson, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 2011, pages 431-446.

Learning to Detect Malicious URLs, Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker, *ACM Transactions on Intelligent Systems and Technology (TIST)* 2(3):30:1-30:24, April 2011.

On the Effects of Registrar-level Intervention, He Liu, Kirill Levchenko, Márk Félegyházi, Christian Kreibich, Gregor Maier, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, Boston, MA, March 2011, pages 1-8.

Got Traffic? An Evaluation of Click Traffic Providers, Qing Zhang, Thomas Ristenpart, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the WICOM/AIRWeb Workshop on Web Quality (WebQuality)*, Hyderabad, India, March 2011, pages 19-26.

Improving Software Diagnosability via Log Enhancement, Ding Yuan, Jing Zheng, Soyeon Park, Yuanyuan Zhou, and Stefan Savage, *Proceedings of Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Newport Beach, CA, March 2011.

2010

How to Tell an Airport from a Home: Techniques and Applications, Andreas Pitsillidis, Yinglian Xie, Fang Yu, Martin Abadi, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the 9th ACM Workshop on Hot Topics in Networks (HotNets-IX)*, Monterey, CA, October 2010, pages 13:1-13:6.

Difference Engine: Harnessing Memory Redundancy in Virtual Machines, Diwaker Gupta, Sangmin Lee, Michael Vrable, Stefan Savage, Alex C. Snoeren, George Varghese, Geoffrey M. Voelker, and Amin Vahdat, *Communications of the Association for Computing Machinery* 53(10):85-93, October 2010.

California Fault Lines: Understanding the Causes and Impact of Network Failures, Daniel Turner, Kirill Levchenko, Alex C. Snoeren, and Stefan Savage, *Proceedings of the ACM SIGCOMM Conference*, New Delhi, India, August 2010.

Re: CAPTCHAs -- Understanding CAPTCHA-Solving from an Economic Context, Marti Motoyama, Kirill Levchenko, Chris Kanich, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the USENIX Security Symposium*, Washington, D.C., August 2010, pages 435-452.

Beyond Heuristics: Learning to Classify Vulnerabilities and Predict Exploits, Mehran Bozorgi, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the ACM SIGKDD Conference*, Washington D.C., July 2010, pages 105-114.

Measuring Online Service Availability Using Twitter, Marti Motoyama, Brendan Meeder, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of ACM Workshop on Online Social Networks (WOSN)*, Boston, MA, June 2010, pages 13:1-13:9.

SleepServer: A Software-Only Approach for Reducing the Energy Consumption of PCs within Enterprise Environments, Yuvraj Agarwal, Stefan Savage, and Rajesh Gupta, *Proceedings of the USENIX Annual Technical Conference*, Boston, MA, June 2010.

Experimental Security Analysis of a Modern Automobile, Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage, *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 2010.

Neon: System Support for Derived Data Management, Qing Zhang, John McCullough, Justin Ma, Nabil Schear, Michael Vrable, Amin Vahdat, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, Pittsburgh, PA, March 2010, pages 63-74.

Cloudflare, Inc. Exhibit 1003 Page 1143

Botnet Judo: Fighting Spam with Itself, Andreas Pitsillidis, Kirill Levchenko, Christian Kreibich, Chris Kanich, Geoffrey M. Voelker, Vern Paxson, Nicholas Weaver, and Stefan Savage, *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2010.

2009

Cumulus: Filesystem Backup to the Cloud, Michael Vrable, Stefan Savage, and Geoffrey M. Voelker, *ACM Transaction on Storage* 5(4):1-28, December 2009.

Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds, Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Chicago, IL, November 2009. (**Test of Time Award**).

When Private Keys are Public: Results from the 2008 Debian OpenSSL Debacle, Scott Yilek, Eric Rescorla, Hovav Shacham, Brandon Enright, and Stefan Savage, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Chicago, IL, November 2009.

Spamalytics: An Empirical Analysis of Spam Marketing Conversion, Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, Vern Paxson, and Stefan Savage, *Communications of the Association for Computing Machinery* 52(9):99-107, September 2009.

Cumulus: filesystem backup to the Cloud, Michael Vrable, Stefan Savage, and Geoffrey M. Voelker, *USENIX ;login:* 34(4):7-13, August 2009.

Identifying Suspicious URLs: An Application of Large-Scale Online Learning, Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the International Conference on Machine Learning*, Montreal, Quebec, June 2009, pages 681-688.

Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs, Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the ACM SIGKDD Conference*, Paris, France, June 2009, pages 1245-1254.

Spamcraft: An Inside Look at Spam Campaign Orchestration, Christian Kreibich, Chris Kanich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, Vern Paxson, and Stefan Savage, *Proceedings of the USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, Boston, MA, April 2009, pages 4:1-4:9.

Difference Engine, Diwaker Gupta, Sangmin Lee, Michael Vrable, Stefan Savage, Alex C. Snoeren, George Varghese, Geoffrey M. Voelker, and Amin Vahdat, *USENIX ;login:* 34(2):24-31, April 2009.

Detecting Malicious Packet Losses, Alper Mizrak, Stefan Savage, and Keith Marzullo, *IEEE Transactions on Parallel and Distributed Systems* 20(2), February 2009.

Cumulus: Filesystem Backup to the Cloud, Michael Vrable, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the 7th USENIX Conference on File and Storage Technologies (FAST)*, San Francisco, CA, February 2009, pages 225-238.

2008

Difference Engine: Harnessing Memory Redundancy in Virtual Machines, Diwaker Gupta, Sangmin Lee, Michael Vrable, Stefan Savage, Alex C. Snoeren, George Varghese, Geoffrey M. Voelker, and Amin Vahdat, *Proceedings of the 8th ACM/USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, San Diego, CA, December 2008, pages 309-322. (**Award paper**).

Spamalytics: an Empirical Analysis of Spam Marketing Conversion, Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Vern Paxson, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Alexandria, VA, October 2008, pages 3-14.

Reconsidering Physical Key Secrecy: Teleduplication via Optical Decoding, Benjamin Laxton, Kai Wang, and Stefan Savage, *Proceedings of the ACM Conference on Computer and Communications*

Cloudflare, Inc. Exhibit 1003 Page 1144

Security (CCS), Alexandria, VA, October 2008.

When Good Instructions Go Bad: Generalizing Return-oriented Programming to the SPARC, Erik Buchanan, Ryan Roemer, Hovav Shacham, and Stefan Savage, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Alexandria, VA, October 2008.

XL: An Efficient Network Routing Algorithm, Kirill Levchenko, Geoffrey M. Voelker, Ramamohan Paturi, and Stefan Savage, *Proceedings of the ACM SIGCOMM Conference*, Seattle, WA, August 2008.

Storm: When Researchers Collide, Brandon Enright, Geoff Voelker, Stefan Savage, Chris Kanich, and Kirill Levchenko, *USENIX ;login:* 33(4):6-13, August 2008.

On the Spam Campaign Trail, Christian Kreibich, Chris Kanich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, Vern Paxson, and Stefan Savage, *Proceedings of the USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, San Franciso, CA, April 2008.

The Heisenbot Uncertainty Problem: Challenges in Separating Bots from Chaff, Chris Kanich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, San Franciso, CA, April 2008.

Detecting Compromised Routers via Packet Forwarding Behavior, Alper Mizrak, Stefan Savage, and Keith Marzullo, *IEEE Network* 22(2), March 2008.

2007

An Inquiry into the Nature and Causes of the Wealth of Internet Miscreants, Jason Franklin, Vern Paxson, Adrian Perrig, and Stefan Savage, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Alexandria, VA, October 2007.

Slicing Spam with Occam's Razor, Chris Fleizach, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of Conference on Email and Anti-Spam (CEAS)*, Mountain View, CA, August 2007.

Automating Cross-Layer Diagnosis of Enterprise Wireless Networks, Yu-Chung Cheng, Mikhail Afanasyev, Patrick Verkaik, Péter Benkö, Jennifer Chiang, Alex C. Snoeren, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the ACM SIGCOMM Conference*, Kyoto, Japan, August 2007.

Spamscatter: Characterizing Internet Scam Hosting Infrastructure, David S. Anderson, Chris Fleizach, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the USENIX Security Symposium*, Boston, MA, August 2007.

2006

Maximizing Data Locality in Distributed Systems, Fan Chung, Ronald Graham, Ranjita Bhagwan, Geoffrey M. Voelker, and Stefan Savage, *Journal of Computer and System Sciences* 72(8), December 2006.

Automated Protocol Inference: Unexpected Means of Identifying Protocols, Justin Ma, Kirill Levchenko, Cristian Kriebich, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Rio de Janeiro, Brazil, October 2006.

Finding Diversity in Remote Code Injection Exploits, Justin Ma, John Dunagan, Helen J. Wang, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the ACM Internet Measurement Conference (IMC)*, Rio de Janeiro, Brazil, October 2006.

Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis, Yu-Chung Cheng, John Bellardo, Péter Benkö, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the ACM SIGCOMM Conference*, Pisa, Italy, September 2006, pages 39-50.

Fatih: Detecting and Isolating Malicious Routers via Traffic Validation, Alper Mizrak, Yu-Chung Cheng, Keith Marzullo, and Stefan Savage, *IEEE Transactions on Dependable and Secure Computing* 3(3), July 2006.

Cloudflare, Inc. Exhibit 1003 Page 1145

Inferring Internet Denial-of-Service Activity, David Moore, Colleen Shannon, Doug Brown, Geoffrey M. Voelker, and Stefan Savage, *ACM Transactions on Computer Systems* 24(2):115-139, May 2006.

2005

Opportunistic Measurement: Extracting Insight from Spurious Traffic, Martin Casado, Tal Garfinkel, Weidong Cui, Vern Paxson, and Stefan Savage, *Proceedings of the 4th ACM Workshop on Hot Topics in Networks (HotNets-IV)*, College Park, MD, November 2005.

Self-stopping Worms, Justin Ma, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the ACM Workshop on Rapid Malcode (WORM)*, Washington D.C., November 2005, pages 12-21.

Scalability, Fidelity and Containment in the Potemkin Virtual Honeyfarm, Michael Vrable, Justin Ma, Jay Chen, David Moore, Erik VandeKieft, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the 20th ACM Symposium on Operating Systems Principles (SOSP)*, Brighton, UK, October 2005, pages 148-162.

Fatih: Detecting and Isolating Malicious Routers, Alper Mizrak, Yu-Chung Cheng, Keith Marzullo, and Stefan Savage, *Proceedings of the IEEE Conference on Dependable Systems and Networks (DSN)*, Yokohama, Japan, June 2005, pages 538-547. (**Award paper**).

SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks, Ishwar Ramani and Stefan Savage, *Proceedings of the IEEE Infocom Conference*, Miami, FL, March 2005.

2004

Automated Worm Fingerprinting, Sumeet Singh, Cristian Estan, George Varghese, and Stefan Savage, *Proceedings of the 6th ACM/USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, San Francisco, CA, December 2004, pages 45-60.

Fault-Tolerant Forwarding in the Face of Malicious Routers, Alper Mizrak, Keith Marzullo, and Stefan Savage, *Proceedings of the International Workshop on the Future Directions in Distributed Computing (FuDiCo)*, Bertinoro, Italy, June 2004.

Monkey See, Monkey Do: A Tool for TCP Tracing and Replaying, Yu-Chung Cheng, Urs Hoelzle, Neal Cardwell, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the USENIX Annual Technical Conference*, Boston, MA, June 2004, pages 87-98.

TotalRecall: System Support for Automated Availability Management, Ranjita Bhagwan, Kiran Tati, Yu-Chung Cheng, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the 1st ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, March 2004, pages 337-350.

2003

In Search of Path Diversity in ISP Networks, Renata Teixeira, Keith Marzullo, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the USENIX/ACM Internet Measurement Conference (IMC)*, Miami, FL, October 2003, pages 313-318.

Automatically Inferring Patterns of Resource Consumption in Network Traffic, Cristian Estan, Stefan Savage, and George Varghese, *Proceedings of the ACM SIGCOMM Conference*, Karlsruhe, Germany, August 2003, pages 137-148.

802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions, John Bellardo and Stefan Savage, *Proceedings of the USENIX Security Symposium*, Washington, D.C., August 2003, pages 15-28.

Cloudflare, Inc. Exhibit 1003 Page 1146

Inside the Slammer Worm, David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver, *IEEE Security and Privacy* 1(4):33-39, July 2003.

Structured Superpeers: Leveraging Heterogeneity to Provide Constant-Time Lookup, Alper Mizrak, Yu-Chung Cheng, Vineet Kumar, and Stefan Savage, *Proceedings of the 4th IEEE Workshop on Internet Applications*, San Jose, CA, June 2003, pages 104-111.

The Phoenix Recovery System: Rebuilding from the Ashes of an Internet Catastrophe, Flavio Junqueira, Ranjita Bhagwan, Keith Marzullo, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the 9th USENIX Workshop on Hot Topics in Operating Systems (HotOS-IX)*, Lihue, HI, May 2003, pages 73-78.

Internet Quarantine: Requirements for Containing Self-Propagating Code, David Moore, Colleen Shannon, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the IEEE Infocom Conference*, San Francisco, CA, April 2003, pages 1901-1910.

Understanding Availability, Ranjita Bhagwan, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the International Workshop on Peer To Peer Systems (IPTPS)*, Berkeley, CA, February 2003, pages 256-267.

The Spread of the Sapphire/Slammer Worm, David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver, CAIDA Report, January 2003.

2002

Measuring Packet Reordering, John Bellardo and Stefan Savage, *Proceedings of the ACM/USENIX Internet Measurement Workshop (IMW)*, Marseille, France, November 2002, pages 97-105.

Automated Measurement of High-Volume Traffic Clusters, Cristian Estan, Stefan Savage, and George Varghese, *Proceedings of the ACM/USENIX Internet Measurement Workshop (IMW)*, Marseille, France, November 2002, pages 77-78.

Replication Strategies for Highly Available Peer-to-Peer Storage, Ranjita Bhagwan, David Moore, Stefan Savage, and Geoffrey M. Voelker, *Proceedings of the International Workshop on the Future Directions in Distributed Computing (FuDiCo)*, Bertinoro, Italy, June 2002.

2001

Robust Congestion Signaling, David Ely, Neil Spring, David Wetherall, Stefan Savage, and Tom Anderson, *Proceedings of the 9th International Conference on Network Protocols (ICNP)*, Riverside, CA, November 2001, pages 332-341.

Inferring Internet Denial of Service Activity, David Moore, Geoffrey M. Voelker, and Stefan Savage, *Proceedings of the USENIX Security Symposium*, Washington, D.C., August 2001, pages 9-22. (**Best paper and Test of Time Awards**).

Network Support for IP Traceback, Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson, *IEEE/ACM Transactions on Networking* 9(3):226-237, June 2001.

Alpine: A User-Level Infrastructure for Network Protocol Development, David Ely, Stefan Savage, and David Wetherall, *Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems (USITS)*, San Francisco, CA, March 2001, pages 171-183.

2000

Practical Network Support for IP Traceback, Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson, *Proceedings of the ACM SIGCOMM Conference*, Stockholm, Sweden, August 2000, pages 295-306.

Cloudflare, Inc. Exhibit 1003 Page 1147

Modeling TCP Latency, Neal Cardwell, Stefan Savage, and Tom Anderson, *Proceedings of IEEE Infocom Conference*, Tel-Aviv, Israel, March 2000, pages 1742-1751.

Understanding the Performance of TCP Pacing, Amit Aggarwal, Stefan Savage, and Tom Anderson, *Proceedings of IEEE Infocom Conference*, Tel-Aviv, Israel, March 2000, pages 1157-1165.

1999

TCP Congestion Control with a Misbehaving Receiver, Stefan Savage, Neal Cardwell, David Wetherall, and Tom Anderson, *ACM SIGCOMM Computer Communication Review* 29(5):71-78, October 1999.

Sting: a TCP-based Network Measurement Tool, Stefan Savage, *Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems (USITS)*, Boulder, CO, October 1999, pages 71-79. (**Best student paper**).

The End-to-end Effects of Internet Path Selection, Stefan Savage, Andy Collins, Eric Hoffman, John Snell, and Tom Anderson, *Proceedings of the ACM SIGCOMM Conference*, Cambridge, MA, September 1999, pages 289-299.

The Case for Informed Transport Protocols, Stefan Savage, Neal Cardwell, and Tom Anderson, *Proceedings of the 7th IEEE Workshop on Hot Topics in Operating Systems (HotOS-VII)*, Rio Rico, AZ, March 1999, pages 58-63.

Detour: a Case for Informed Internet Routing and Transport, Stefan Savage, Tom Anderson, Amit Aggarwal, David Becker, Neal Cardwell, Andy Collins, Eric Hoffman, John Snell, Amin Vahdat, Geoffrey M. Voelker, and John Zahorjan, *IEEE Micro* 19(1):50-59, January 1999.

1997

Eraser: A Dynamic Race Detector for Multi-Threaded Programs, Stefan Savage, Michael Burrows, Greg Nelson, Patrick Sobalvarro, and Thomas Anderson, *ACM Transactions on Computer Systems* 15(4):391-411, November 1997.

Eraser: A Dynamic Race Detector for Multi-Threaded Programs, Stefan Savage, Michael Burrows, Greg Nelson, PAtrick Sobalvarro, and Thomas Anderson, *Proceedings of the 16th ACM Symposium on Operating Systems Principles (SOSP)*, Saint-Malo, France, October 1997, pages 27-37. (**Award paper**).

1996

Language Support for Extensible Operating Systems, Wilson Hsieh, Marc Fiuczynski, Charles Garrett, Stefan Savage, David Becker, and Brian Bershad, *Proceedings of the 1st Workshop on Compiler Support for System Software*, Tucson, AZ, February 1996, pages 127-133.

Writing an Operating System with Modula-3, Emin gun Sirer, Stefan Savage, Przemyslaw Pardyak, Greg P. DeFouw, Mary Ann Alapat, and Brian Bershad, *Proceedings of the 1st Workshop on Compiler Support for System Software*, Tucson, AZ, February 1996, pages 134-140.

AFRAID -- A Frequently Redundany Array of Independent Disks, Stefan Savage and John Wilkes, *Proceedings of the USENIX Annual Technical Conference*, San Diego, CA, January 1996, pages 27-39. (**Best student paper**).

1995

Cloudflare, Inc. Exhibit 1003 Page 1148

Extensibility, Safety and Performance in the SPIN Operating System, Brian Bershad, Stefan Savage, Przemyslaw Pardyak, Emin Gun Sirer, Marc Fiuczynski, David Becker, Craig Chamgers, and Susan Eggers, *Proceedings of the 15th ACM Symposium on Operating Systems Principles (SOSP)*, Copper Mountain, CO, December 1995, pages 267-284.

Protection is a Software Issue, Brian Bershad, Stefan Savage, Przemyslaw Pardyak, David Becker, Marc Fiuczynski, and Emin gun Sirer, *Proceedings of the 5th Workshop on Hot Topics in Operating Systems*, Orcas Island, WA, May 1995, pages 62-65.

1994

Processor Capacity Reserves: Operating System Support for Multimedia Applications, Clifford Mercer, Stefan Savage, and Hideyuki Tokuda, *Proceedings of the IEEE Interational Conference on Multimedia Computing and Systems (ICMCS)*, Boston, MA, May 1994, pages 90-99.

1993

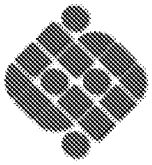
Processor Capacity Reserves: An Abstraction for Managing Processor Usage, Clifford Mercer, Stefan Savage, and Hideyuki Tokuda, *Proceedings of the 4th Workshop on Workstation Operating Systems*, Napa, CA, October 1993, pages 129-134.

Real-time Mach Timers: Exporting Time to the User, Stefan Savage and Hideyuki Tokuda, *Proceedings of the 3rd USENIX Mach Symposium*, Sante Fe, NM, April 1993, pages 221-232.

Cloudflare, Inc. Exhibit 1003 Page 1149

Declaration of Dr. Stefan Savage

ATTACHMENT D
(EXHIBIT 1023)



Information for:

Search SEARCH

acm sigcomm

Welcome

ACM Statement

In the United States, and throughout many regions around the world, current events have brought attention to the urgent need for equality and respect for all individuals. ACM reaffirms its commitment to fostering diversity and inclusion within the global computer science community.

As an international organization of nearly 100,000 professionals, ACM respects the essential worth of all people and firmly believes that diversity strengthens the scientific enterprise and society as a whole. ACM's Code of Professional Ethics embodies these principles, and our Diversity and Inclusion Council works to translate them into effective action.

ACM is committed to creating an environment that welcomes new ideas and perspectives, and where hostility or other harmful behaviors are not tolerated. As an organization, we stand with those who promote inclusivity.

SIGCOMM is ACM's professional forum for discussing communications and computer networks.

SIGCOMM members include scientists, engineers, educators and students. They study all aspects of computer communications and networks: analysis, technical design, engineering, measurement and management. Our members are particularly interested in the systems engineering and architectural questions surrounding computer communication.

Special Statement on Freedom of Mobility

SIGCOMM supports a wide variety of activities in the field:

Professional meetings

- The annual SIGCOMM Conference, our flagship conference
- SIG-sponsored or co-sponsored conferences and workshops: CoNEXT, IMC, Hotnets, Sensys, ICN, ANCS, SOSR and ANRW
- Conferences and workshops in cooperation with SIGCOMM
- The SIGCOMM Artifacts Evaluation Committee

Publications

- Computer Communication Review, our newsletter
- IEEE/ACM Transactions on Networking

Recognition and support for computer networking professionals

- Awards: the SIGCOMM Award, the Test of Time Award, the Rising Star Award, the Best Paper and Best Student Paper Awards, the Doctoral Dissertation Award, and the SIGCOMM Networking Systems Award.
- Travel Grants
- A list of all our grants

Here are links to more information about SIGCOMM:

- Membership

Cloudflare, Inc. Exhibit 1003 Page 1151

Annual reports
Blog
Officers

If you are a computer networking professional, please consider joining ACM SIGCOMM: Join online here.

Recent News

January 2021 SIGCOMM Newsletter
January 31, 2021

Contents
SIG elections
ACM SIGCOMM 2021, Call for Workshops
APNet 2021, Call for Papers

SIG elections
SIGCOMM will be holding elections for the positions of Chair, Vice-Chair, and Treasurer. To see the slate of candidates, please visit the 2021 ACM SIG Elections site: https://www.acm.org/elections/sigs/2021-candidate-slate

A description of the voting procedure and schedule can be found at https://www.acm.org/elections/sigs/poi_proc with a detailed schedule available here. The election site is expected to go live in April with an email notification with voting instructions sent to all SIG members. Elections will close by June 7, with results announced by June 14 and hand-over to the new elected officers happening on July 1st.

Note also that in accordance with the ACM SIG Bylaws, additional candidates may be placed on the ballot by petition. All candidates must be ACM Professional Members, as well as members of the SIG. Anyone interested in petitioning must inform ACM Headquarters, Pat Ryan (ryanp@hq.acm.org) and the Secretary of the SIG of their intent to petition by 16 March 2021. Petitions must be submitted to ACM Headquarters for verification by 2 April 2021.

ACM SIGCOMM 2021, Call for Workshops

(full Call for Workshops may be found at https://conferences.sigcomm.org/sigcomm/2021/cf-workshops.html)

SIGCOMM 2021 plans to hold multiple one-day/half-day virtual (online) workshops in conjunction with the conference, to be held between August 24 and August 26, 2021. Workshops will take place on Monday, August 23 and Friday, August 27.We invite you to submit workshop proposals on any topic related to the broad set of research and application areas covered by the SIGCOMM conference, including topics in network research, practice, and education. Topics that represent new research areas and directions are particularly welcome. The deadline for receiving proposals is February 5, 2021.

APNet 2021, Call for Papers
(Full CFP may be found at https://conferences.sigcomm.org/events/apnet2021/cfp.html)

The 5th Asia-Pacific Workshop on Networking (APNet'21) aims to bring together the very best researchers in computer networking and systems across the Asia-Pacific region and around the globe to a live forum discussing innovative ideas at their early stages. The mission of APNet is that promising but not-yet-mature ideas can receive timely feedback from the community and experienced researchers, leading them into publications at major conferences such as

Cloudflare, Inc. Exhibit 1003 Page 1152

SIGCOMM, NSDI, SOSP, OSDI, MobiCom, and CoNEXT.

Abstract registration deadline: March 12, 2021
Paper submission deadline: March 19, 2021
Notification date: April 30, 2021 (tentative)

December 2020 SIGCOMM Newsletter
December 31, 2020

Contents
ACM SIGCOMM 2021, Call for Workshops
ACM SIGCOMM 2021, Call for Papers

ACM SIGCOMM 2021, Call for Workshops

(full Call for Workshops may be found at https://conferences.sigcomm.org/sigcomm/2021/cf-workshops.html)

SIGCOMM 2021 plans to hold multiple one-day/half-day virtual (online) workshops in conjunction with the conference, to be held between August 24 and August 26, 2021. Workshops will take place on Monday, August 23 and Friday, August 27.We invite you to submit workshop proposals on any topic related to the broad set of research and application areas covered by the SIGCOMM conference, including topics in network research, practice, and education. Topics that represent new research areas and directions are particularly welcome. The deadline for receiving proposals is February 5, 2021.

ACM SIGCOMM 2021, Call for Papers

(full CFP may be found at http://conferences.sigcomm.org/sigcomm/2021/cfp.html)

The ACM SIGCOMM 2021 conference seeks papers describing significant research contributions to the field of communication networks and networked systems. SIGCOMM takes a broad view of networking research, which includes all types of computer networks, wired and wireless networking technologies, networked systems, as well as aspects of computer network design, analysis, experimentation, and more. SIGCOMM 2021 will also accept submissions to an experience track in addition to the regular research track, describing the design, analysis, and evaluation of techniques in commercial or otherwise widely-used deployments. In addition to the main conference, SIGCOMM 2021 will have a series of co-located workshops, tutorials, poster and demo sessions, and conference best paper and SIGCOMM awards.

Abstract registration deadline: Jan 20, 2021, 16:59 EST
Paper submission deadline: Jan 27, 2021, 16:59 EST
Notification date: May 3, 2021 (tentative)

November 2020 SIGCOMM Newsletter
November 30, 2020

Contents
ACM SIGCOMM 2021, Call for Papers
ACM History Committee—Call for Proposals 2021
Deadline for the April 2021 issue of ACM SIGCOMM CCR: December 1

Cloudflare, Inc. Exhibit 1003 Page 1153

ACM CoNEXT 2021 -- Call for Organizing Committee

ACM SIGCOMM 2021
Call for Papers

(full CFP may be found at http://conferences.sigcomm.org/sigcomm/2021/cfp.html)

The ACM SIGCOMM 2021 conference seeks papers describing significant research contributions to the field of communication networks and networked systems. SIGCOMM takes a broad view of networking research, which includes all types of computer networks, wired and wireless networking technologies, networked systems, as well as aspects of computer network design, analysis, experimentation, and more. SIGCOMM 2021 will also accept submissions to an experience track in addition to the regular research track, describing the design, analysis, and evaluation of techniques in commercial or otherwise widely-used deployments. In addition to the main conference, SIGCOMM 2021 will have a series of co-located workshops, tutorials, poster and demo sessions, and conference best paper and SIGCOMM awards.

Abstract registration deadline: Jan 20, 2021, 16:59 EST
Paper submission deadline: Jan 27, 2021, 16:59 EST
Notification date: May 3, 2021 (tentative)

ACM History Committee — Call for Proposals 2021

The ACM History Committee runs this program, which has resulted over 35 fellowships awarded since the start of this fellowship program in 2009. Many of these awards have gone to scholars (graduate students to senior scholars) in the SIGs to support preservation of a SIG's history.

The submission deadline for proposals is February 15, 2021.

See https://history.acm.org/wp-content/uploads/2020/11/ACM-fellowship_CFP_2021-final-1.pdf for proposal submission requirements and further details.

Deadline for the April 2021 issue of ACM SIGCOMM CCR: December 1

The ACM SIGCOMM Computer Communication Review (CCR) newsletter is published every quarter. The deadline for the submission of technical papers, education papers and editorials for the April 2021 issue of CCR is December 1, 2020.

You can find submission details at https://ccronline.sigcomm.org/instructions-for-submission/

ACM CoNEXT 2021 -- Call for Organizing Committee

ACM CoNEXT is soliciting proposals from organization committees to host a virtual CoNEXT 2021 conference. As the SIG has opted for all 2021 conferences to be virtual, "site" proposals are simpler as they need not include information typically involved with holding physical events, i.e., venue, hotels, banquet, etc. Below you find the required details. Organization Committee proposals should be less than a single page, in pdf format. Check the details in the online call. Deadline January 15 2021.

Cloudflare, Inc. Exhibit 1003 Page 1154

October 2020 SIGCOMM Newsletter
October 31, 2020

Contents
Call for Participation: HotNets 2020 (November 4-6, 2020)
Call for nominations: IRTF Applied Networking Research Prize (ANRP) 2021
October 2020 issue of CCR

Call for Participation: HotNets 2020 (November 4-6, 2020)

In light of continued effects from the COVID-19 pandemic, HotNets 2020 (November 4-6, 2020) will be a fully virtual event. We will take this opportunity to deploy and try out several new features designed to encourage interaction and discussion at HotNets.

- In contrast to the invite-only format in previous years, we opened up registration for HotNets 2020 to all members of the community.
- We will provide two types of access to the workshop: live interactive zoom sessions for registered attendees, and concurrent streaming YouTube broadcasts for other viewers.
- In addition to Zoom/YouTube, we will host parallel discussions and Q&A on Slack, which is also open to the general public.

Finally, Hotnets 2020 will be a 3-day event (3.5 hours per day)

November 4th, 2020 (Wed): Start: 8:45AM (US Central Time), End: 12:30PM (US Central Time)
November 5th, 2020 (Thur): Start: 9:00AM (US Central Time), End: 12:30PM (US Central Time)
November 6th, 2020 (Fri): Start: 9:00AM (US Central Time), End: 12:30PM (US Central Time)

We hope that this combination of tools and platforms will facilitate more lively discussions on new ideas and future research topics.

Registered attendees will receive zoom session information via email, while information on YouTube broadcasts and Slack will be posted soon to the HotNets'20 website (https://conferences.sigcomm.org/hotnets/2020/program.html). The full program is already posted on this site. Please stay tuned.

We look forward to seeing you all at HotNets 2020.
HotNets Organization Committee

Call for nominations: Applied Networking Research Prize (ANRP) 2021

Submit nominations for the ANRP 2021 until 22 November 2020

The Applied Networking Research Prize is awarded to recognise the best recent results in applied networking, interesting new research ideas of potential relevance to the Internet standards community, and upcoming people that are likely to have an impact on Internet standards and technologies, with a particular focus on cases where these people or ideas would not otherwise get much exposure or be able to participate in the discussion.

We encourage nominations of researchers with relevant research results, interesting ideas, and new perspectives. The award provides a cash prize, and will offer the winner the opportunity to present and discuss their work with the engineers, network operators, policy makers, and scientists that participate in the Internet Engineering Task Force (IETF) and its research arm, the Internet Research Task Force (IRTF). Both self- and third-party nominations for this prize are encouraged.

More details, and how to nominate, at https://irtf.org/anrp/

Cloudflare, Inc. Exhibit 1003 Page 1155

October 2020 issue of CCR

The online version of the October 2020 issue of CCR is available at https://ccronline.sigcomm.org/2020/ccr-october-2020-2020/the-october-2020-issue/.

September 2020 SIGCOMM Newsletter
September 30, 2020

Contents
Call for participation, HotNets 2020
SIGCOMM Rising Star Award Winner of 2020

Call for participation, HotNets 2020

Registration for HotNets 2020 is now open. Please visit http://conferences.sigcomm.org/hotnets/2020/register.html to register. Note that seats are limited and will be filled up on a first-come, first-served basis.

SIGCOMM Rising Star Award Winner of 2020

The SIGCOMM Rising star award of 2020 goes to Keith Winstein (Stanford University).
The award is in recognition of outstanding research contributions in the areas of computer-generated robust networking algorithms such as congestion control. His innovative and out-of-the-box approaches, supporting the networking community through open datasets and open source software, have impacted academia, industry and society. The rising star award committee comprised Sharad Agarwal (MSR, chair), Morley Mao (University of Michigan), Matt Roughan (University of Adelaide), and Hakim Weatherspoon (Cornell University).



All SIGCOMM News

Contact us to send feedback about this page.

Cloudflare, Inc. Exhibit 1003 Page 1156

ATTACHMENT E
(EXHIBIT 1017)

ACM SIGCOMM 2000

Monday, Aug 28

Tutorial M1 09.00-16.30	**Voice over IP** Henning Schulzrinne, Columbia University	Vapensalen
Tutorial M2 09.00-16.30	**Analysis of Network and Protocol Behavior** Shawn Ostermann, Ohio University	Spegelsalen
12.00-13.00	**Lunch**	Vinterträdgården

Tuesday, Aug 29

Tutorial T1 09.00-16.30	**Network Security Protocols** Radia Perlman, Sun Microsystems	Spegelsalen
Tutorial T2 09.00-16.30	**Distributed Control and Resource Pricing** R. Gibbens, University of Cambridge, UK, P. Key, Microsoft Research	Vapensalen
12.00-13.00	**Lunch**	Vinterträdgården
18.00-19.30	**Opening Reception** Boats leave for City Hall at 17.00 in front of Grand Hôtel.	Stadshuset (Stockholm City Hall)

Wednesday, Aug 30

Opening Session and SIGCOMM Award Ceremony — Vinterträdgården
09.00-10.15 *Mutual Cloning between Network Paradigms* — André Danthine

10.15-10.45 **Break** — Vapensalen, Spegelsalen, Strömsalongerna

Session 1 — **Multicast and Anycast** — Vinterträdgården
10.45-12.15 Chair: Jon Crowcroft

A Framework for Scalable Global IP-Anycast (GIA) — Dina Katabi, John Wroclawski (Massachusetts Institute of Technology)

pgmcc: A TCP-friendly Single-Rate Multicast Congestion Control Scheme — Luigi Rizzo (Universita di Pisa)

Fault Isolation in Multicast Trees — Anoop Reddy, Ramesh Govindan, Deborah Estrin (USC/Information Sciences Institute)

12.15-13.30 **Lunch** — Vapensalen, Spegelsalen

Session 2 — **Control Mechanisms** — Vinterträdgården
13.30-15.00 Chair: Henning Schulzrinne

Equation-Based Congestion Control for Unicast Applications — Sally Floyd, Mark Handley (AT&T Center for Internet Research at ICSI), Jitendra Padhye (University of Massachusetts, Amherst), Joerg Widmer (AT&T Center for Internet Research at ICSI)

Endpoint Admission Control: Architectural Issues and Performance — Lee Breslau (AT&T Research), Edward Knightly (Rice University), Scott Shenker (AT&T Center for Internet Research at ICSI), Ion Stoica, Hui Zhang (CMU)

Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services — Zhi-Li Zhang, Zhenhai Duan (University of Minnesota), Lixin Gao (Smith College), Yiwei Thomas Hou (Fujitsu Labs)

15.00-15.30 **Break** — Vapensalen, Spegelsalen, Strömsalongerna

Session 3 — **World Wide Web** — Vinterträdgården
15.30-17.00 Chair: Erich Nahum

A Protocol-Independent Technique for Eliminating Redundant Network Traffic — Neil T. Spring, David Wetherall (University of Washington)

On Network-Aware Clustering of Web Clients — Balachander Krishnamurthy (AT&T Research), Jia Wang (Cornell University)

The Content and Access Dynamics of a Busy Web Site: Findings and Implications — Venkata N. Padmanabhan (Microsoft Research), Lili Qiu (Cornell University)

Outrageous Opinions Session — Spegelsalen, Strömsalongerna
17.15-18.30 Chair: Gary Delp

Cloudflare, Inc. Exhibit 1003 Page 1158

Thursday, Aug 31

Session 4 **Performance Analysis and Modeling** Vinterträdgården
09.00-10.30 Chair: Keith Ross

Critical Path Analysis of TCP Transactions — Paul Barford, Mark Crovella (Boston University)

Tuning RED for Web Traffic — Mikkel Christiansen, Kevin Jeffay, David Ott, F. Donelson Smith (University of North Carolina at Chapel Hill)

A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED — Vishal Misra, Wei-Bo Gong, Don Towsley (University of Massachusetts, Amherst)

10.30-11.00 **Break** Vapensalen, Spegelsalen, Strömsalongerna

Session 5 **Routing Stability and Convergence** Vinterträdgården
11.00-12.00 Chair: Roch Guerin

Routing Stability in Congested Networks: Experimentation and Analysis — Aman Shaikh (University of California, Santa Cruz), Lampros Kalampoukas, Rohit Dube (Bell Labs, Lucent Technologies), Anujan Varma (University of California, Santa Cruz)

An Experimental Study of Delayed Internet Routing Convergence — Craig Labovitz (Microsoft Research), Abha Ahuja, Abhijit Abose, Farnam Jahanian (University of Michigan)

12.00-13.30 **Lunch** Vapensalen, Spegelsalen

Session 6 **Routing and Bridging** Vinterträdgården
13.30-14.30 Chair: Julio Escobar

FIRE: Flexible Intra-AS Routing Environment — Craig Partridge (BBN Technologies), Alex C. Snoeren (BBN Technologies & Massachusetts Institute of Technology), Tim Strayer, Beverly Schwartz, Matthew Condell, Isidro Castineyra (BBN Technologies)

SmartBridge: A Scalable Bridge Architecture — Thomas L. Rodeheffer, Chandramohan A. Thekkath (Compaq Systems Research Center), Darrell Anderson (Duke University)

Student Poster Session – Work in Progress Vinterträdgården
14.30-15.30 In parallel with the break

Session 7 **TCP Analysis** Vinterträdgården
15.30-17.00 Chair: Jean-Yves Leboudec

TCP is Max-Plus Linear — Francois Baccelli, Dohy Hong (ENS-INRIA)

A Stochastic Model of TCP/IP with Stationary Random Losses — Eitan Altman, Kostia Avrachenkov, Chadi Barakat (INRIA)

On the Propagation of Long-range Dependency in the Internet — Andras Veres (Ericsson), Zsolt Kenesi, Sandor Molnar (Technical University of Budapest), Gabor Vattay (Eotvos University, Budapest)

17.00-18.00 **Sigcomm Business Meeting** Spegelsalen

18.30-21.00 **Dinner** Vasa Museum
Buses start to leave at 18.00 in front of Grand Hôtel.

Friday, Sep 1

Session 8 **Tracing and Measurement** Vinterträdgården
09.00-11.00 Chair: Bryan Lyles

Deriving Traffic Demands for Operational IP Networks: Methodology and Experience — Anja Feldmann (University of Saarbruecken), Albert Greenberg, Carsten Lund, Nick Reingold, Jennifer Rexford, Fred True (AT&T Research)

Trajectory Sampling for Direct Traffic Observation — Nick Duffield, Matthias Grossglauser (AT&T Research)

Measuring Link Bandwidths Using a Deterministic Model of Packet Delay — Kevin Lai, Mary Baker (Stanford University)

Practical Network Support For IP Traceback — Stefan Savage, David Wetherall, Anna Karlin, Tom Anderson (University of Washington)

11.00-11.30 **Break** Vapensalen, Spegelsalen, Strömsalongerna

Session 9 **Header Processing** Vinterträdgården
11.30-13.00 Chair: Vern Paxson

When the CRC and TCP Checksum Disagree — Jonathan Stone (Stanford University), Craig Partridge (BBN Technologies)

Packet Types: Abstract Specification of Network Protocol Messages — Peter J. McCann, Satish Chandra (Bell Labs, Lucent Technologies)

Memory-Efficient State Lookups with Fast Updates — Sandeep Sikka (Inktomi), George Varghese (UCSD)

13.00-13.30 **Closing Remarks** Vinterträdgården

Cloudflare, Inc. Exhibit 1003 Page 1159

ATTACHMENT F
(EXHIBIT 1018)

http://www.acm.org/sigcomm/sigcomm2000/conf/techprog.htm Go JUL AUG OCT
22 captures 20 Aug 2000 - 3 Jan 2008 1999 2000 2001 About this capture

ACM SIGCOMM 2000
August 28 - September 1, 2000
Grand Hotel, Stockholm, Sweden

Call for papers
Paper Submission
Organization Committee
Program Committee
- Conference program
Registration
Local information
Student travel awards

CONFERENCE PROGRAM

Program at a glance Tutorial program Technical program Abstracts Papers

Preliminary Technical Program

Preliminary technical program including abstracts in PostScript Format .ps or .ps.gz, Portable Document Format .pdf, or in HTML format.

Introduction Session
Wednesday, Aug 30, 9.00-10.15

- **Welcome**
- **SIGCOMM Award Ceremony**
  André Danthine (University of Liege, Belgium)
- **Keynote Speech**
  André Danthine

Session 1: Multicast and Anycast
Wednesday, Aug 30, 10.45-12.15

- **A Framework for Scalable Global IP-Anycast (GIA)** (abstract, paper)
  Dina Katabi, John Wroclawski (Massachusetts Institute of Technology)
- **pgmcc: A TCP-friendly Single-Rate Multicast Congestion Control Scheme** (abstract, paper)
  Luigi Rizzo (Universita di Pisa)
- **Fault Isolation in Multicast Trees** (abstract, paper)
  Anoop Reddy, Ramesh Govindan, Deborah Estrin (USC/Information Sciences Institute)

Session 2: Control Mechanisms
Wednesday, Aug 30, 13.30-15.00

- **Equation-Based Congestion Control for Unicast Applications** (abstract, paper)
  Sally Floyd, Mark Handley (AT&T; Center for Internet Research at ICSI), *Jitendra Padhye* (University of Massachusetts, Amherst), *Joerg Widmer* (AT&T; Center for Internet Research at ICSI)
- **Endpoint Admission Control: Architectural Issues and Performance** (abstract, paper)
  Lee Breslau (AT&T; Research), *Edward Knightly* (Rice University), *Scott Shenker* (AT&T; Center for Internet Research at ICSI), *Ion Stoica, Hui Zhang* (CMU)
- **Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services** (abstract, paper)
  Zhi-Li Zhang, Zhenhai Duan (University of Minnesota), *Lixin Gao* (Smith College), *Yiwei Thomas Hou* (Fujitsu Labs)

Cloudflare, Inc. Exhibit 1003 Page 1161

http://www.acm.org/sigcomm/sigcomm2000/conf/techprog.htm Go JUL AUG OCT
22 captures 20 Aug 2000 - 3 Jan 2008 20 1999 2000 2001 About this capture

Traffic (abstract, paper)
Neil T. Spring, David Wetherall (University of Washington)

- **On Network-Aware Clustering of Web Clients** (abstract, paper)
  Balachander Krishnamurthy (AT&T; Research), *Jia Wang* (Cornell University)
- **The Content and Access Dynamics of a Busy Web Site: Findings and Implications** (abstract, paper)
  Venkata N. Padmanabhan (Microsoft Research), *Lili Qiu* (Cornell University)

Session 4: Performance Analysis and Modeling
Thursday, Aug 31, 9.00-10.30

- **Critical Path Analysis of TCP Transactions** (abstract, paper)
  Paul Barford, Mark Crovella (Boston University)
- **Tuning RED for Web Traffic** (abstract, paper)
  Mikkel Christiansen, Kevin Jeffay, David Ott, F. Donelson Smith (University of North Carolina at Chapel Hill)
- **A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED** (abstract, paper)
  Vishal Misra, Wei-Bo Gong, Don Towsley (University of Massachusetts, Amherst)

Session 5: Routing Stability and Convergence
Thursday, Aug 31, 11.00-12.00

- **Routing Stability in Congested Networks: Experimentation and Analysis** (abstract, paper)
  Aman Shaikh (University of California, Santa Cruz), *Lampros Kalampoukas, Rohit Dube* (Bell Labs, Lucent Technologies), *Anujan Varma* (University of California, Santa Cruz)
- **An Experimental Study of Delayed Internet Routing Convergence** (abstract, paper)
  Craig Labovitz (Microsoft Research), *Abha Ahuja, Abhijit Abose, Farnam Jahanian* (University of Michigan)

Session 6: Routing and Bridging
Thursday, Aug 31, 13.30-14.30

- **FIRE: Flexible Intra-AS Routing Environment** (abstract, paper)
  Craig Partridge (BBN Technologies), *Alex C. Snoeren* (BBN Technologies & Massachusetts Institute of Technology), *Tim Strayer, Beverly Schwartz, Matthew Condell, Isidro Castineyra* (BBN Technologies)
- **SmartBridge: A Scalable Bridge Architecture** (abstract, paper)
  Thomas L. Rodeheffer, Chandramohan A. Thekkath (Compaq Systems Research Center), *Darrell Anderson* (Duke University)

Student Poster Session - Work in Progress
Thursday, Aug 31, 14.30-15.30

Cloudflare, Inc. Exhibit 1003 Page 1162

http://www.acm.org/sigcomm/sigcomm2000/conf/techprog.htm Go JUL AUG OCT
22 captures 20 Aug 2000 - 3 Jan 2008 1999 2000 2001 About this capture

Thursday, Aug 31, 15.30-17.00

- **TCP is Max-Plus Linear** (abstract, paper)
  Francois Baccelli, Dohy Hong (ENS-INRIA)
- **A Stochastic Model of TCP/IP with Stationary Random Losses** (abstract, paper)
  Eitan Altman, Kostia Avrachenkov, Chadi Barakat (INRIA)
- **On the Propagation of Long-range Dependency in the Internet** (abstract, paper)
  Andras Veres (Ericsson), *Zsolt Kenesi, Sandor Molnar* (Technical University of Budapest), *Gabor Vattay* (Eotvos University, Budapest)

Session 8: Tracing and Measurement
Friday, Sep 1, 9.00-11.00

- **Deriving Traffic Demands for Operational IP Networks: Methodology and Experience** (abstract, paper)
  Anja Feldmann (University of Saarbruecken), *Albert Greenberg, Carsten Lund, Nick Reingold, Jennifer Rexford, Fred True* (AT&T; Research)
- **Trajectory Sampling for Traffic Measurement** (abstract, paper)
  Nick Duffield, Matthias Grossglauser (AT&T; Research)
- **Measuring Link Bandwidths Using a Deterministic Model of Packet Delay** (abstract, paper)
  Kevin Lai, Mary Baker (Stanford University)
- **Practical Network Support For IP Traceback** (abstract, paper)
  Stefan Savage, David Wetherall, Anna Karlin, Tom Anderson (University of Washington)

Session 9: Header Processing
Friday, Sep 1, 11.30-13.00

- **When the CRC and TCP Checksum Disagree** (abstract, paper)
  Jonathan Stone (Stanford University), *Craig Partridge* (BBN Technologies)
- **Packet Types: Abstract Specification of Network Protocol Messages** (abstract, paper)
  Peter J. McCann, Satish Chandra (Bell Labs, Lucent Technologies)
- **Terabit IP Lookups** (abstract, paper)
  Sandeep Sikka (Inktomi), *George Varghese* (UCSD)

Cloudflare, Inc. Exhibit 1003 Page 1163

ATTACHMENT G

Practical Network Support for IP Traceback

Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson

Department of Computer Science and Engineering
University of Washington
Seattle, WA, USA

Abstract

This paper describes a technique for tracing anonymous packet flooding attacks in the Internet back towards their source. This work is motivated by the increased frequency and sophistication of denial-of-service attacks and by the difficulty in tracing packets with incorrect, or "spoofed", source addresses. In this paper we describe a general purpose traceback mechanism based on probabilistic packet marking in the network. Our approach allows a victim to identify the network path(s) traversed by attack traffic without requiring interactive operational support from Internet Service Providers (ISPs). Moreover, this traceback can be performed "post-mortem" – after an attack has completed. We present an implementation of this technology that is incrementally deployable, (mostly) backwards compatible and can be efficiently implemented using conventional technology.

1. INTRODUCTION

Denial-of-service attacks consume the resources of a remote host or network, thereby denying or degrading service to legitimate users. Such attacks are among the hardest security problems to address because they are simple to implement, difficult to prevent, and very difficult to trace. In the last several years, Internet denial-of-service attacks have increased in frequency, severity and sophistication. Howard reports that between the years of 1989 and 1995, the number of such attacks reported to the Computer Emergency Response Team (CERT) increased by 50 percent per year [25]. More recently, a 1999 CSI/FBI survey reports that 32 percent of respondents detected denial-of-service attacks directed against their sites [16]. Even more worrying, recent reports indicate that attackers have developed tools to coordinate distributed attacks from many separate sites [14].

Unfortunately, mechanisms for dealing with denial-of-service have not advanced at the same pace. Most work in this area has focused on *tolerating* attacks by mitigating their effects on the victim [38, 2, 26, 29, 9]. This approach can provide an effective stop-gap measure, but does not eliminate the problem nor does it discourage attackers. The other option, and the focus of this paper, is to trace attacks back towards their origin – ideally stopping an attacker at the source.

A perfect solution to this problem is complicated by the potential use of indirection to "launder" the true causal origin of an attack. For example, an attack may consist of packets sent from many different slave machines, themselves under the control of a remote master machine. Such indirection may be achieved either explicitly (by compromising the individual slave hosts directly) or implicitly (by sending false requests to the slaves on behalf of the victim – a so-called *reflector*). More challenging still, the true origin and identity of the attacker can be similarly concealed through chains of false computer accounts, call forwarding, and so forth. Consequently, we regard a complete solution – particularly one able to address the forensic needs of law enforcement – as an open problem.

Instead, we address the more limited operational goal of simply identifying the machines that *directly* generate attack traffic and the network path this traffic subsequently follows. We call this the *traceback problem* and it is motivated by the operational need to control and contain attacks. In this setting, even incomplete or approximate information is valuable because the efficacy of measures such as packet filtering improve as they are applied further from the victim and closer to the source.

However, even for our restricted problem, determining the source generating attack traffic is surprisingly difficult due to the stateless nature of Internet routing. Attackers routinely disguise their location using incorrect, or "spoofed", IP source addresses. As these packets traverse the Internet their true origin is lost and a victim is left with little useful information. While there are several ad hoc traceback techniques in use, they all have significant drawbacks that limit their practical utility in the current Internet.

In this paper we present a new approach to the traceback problem that addresses the needs of both victims and network operators. Our solution is to probabilistically mark packets with partial path information as they arrive at routers. This approach exploits the observation that attacks generally comprise large numbers of packets. While each marked packet represents only a "sample" of the path it has traversed, by combining a modest number of such packets a victim can reconstruct the entire path. This allows victims to locate the approximate source of attack traffic without requiring the assistance of outside network operators. Moreover, this determination can be made even after an attack has completed. Both facets of our solution represent substantial improvements over existing capabilities for dealing with flooding-style denial-of-service attacks.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGCOMM'00, Stockholm, Sweden.
Copyright 2000 ACM 1-58113-224-7/00/0008...$5.00.

A key practical deployment issue with any modification of Internet routers is to ensure that the mechanisms are efficiently implementable, may be incrementally deployed, and are backwards compatible with the existing infrastructure. We describe a traceback algorithm that adds little or no overhead to the router's critical forwarding path and may be incrementally deployed to allow traceback within the subset of routers supporting our scheme. Further, we demonstrate that we can encode the necessary path information in a way that peacefully co-exists with existing routers, host systems and more than 99% of today's traffic.

The rest of this paper is organized as follows: In Section 2, we describe related work concerning IP spoofing and solutions to the traceback problem. Section 3 outlines our basic approach and section 4 characterizes several abstract algorithms for implementing it. In Section 5 we detail a concrete encoding strategy for our algorithm that can be implemented within the current Internet environment. We also present experimental results demonstrating the effectiveness of our solution. In section 6 we discuss the main limitations and weaknesses of our proposal and potential extensions to address some of them. Finally, we summarize our findings in Section 7.

2. RELATED WORK

It has been long understood that the IP protocol permits anonymous attacks. In his 1985 paper on TCP/IP weaknesses, Morris writes:

> "The weakness in this scheme [the Internet Protocol] is that the source host itself fills in the IP source host id, and there is no provision in ... TCP/IP to discover the true origin of a packet." [31]

In addition to denial-of-service attacks, IP spoofing can be used in conjunction with other vulnerabilities to implement anonymous one-way TCP channels and covert port scanning [31, 3, 24, 44].

There have been several efforts to reduce the anonymity afforded by IP spoofing. Table 1 provides a subjective characterization of each of these approaches in terms of management cost, additional network load, overhead on the router, the ability to trace multiple simultaneous attacks, the ability trace attacks after they have completed, and whether they are preventative or reactive. We also characterize our proposed traceback scheme according to the same criteria. In the remainder of this section we describe each previous approach in more detail.

2.1 Ingress filtering

One way to address the problem of anonymous attacks is to eliminate the ability to forge source addresses. One such approach, frequently called *ingress filtering*, is to configure routers to block packets that arrive with illegitimate source addresses [20]. This requires a router with sufficient power to examine the source address of every packet and sufficient knowledge to distinguish between legitimate and illegitimate addresses. Consequently, ingress filtering is most feasible in customer networks or at the border of Internet Service Providers (ISP) where address ownership is relatively unambiguous and traffic load is low. As traffic is aggregated from multiple ISPs into transit networks, there is no longer enough information to unambiguously determine if a packet arriving on a particular interface has a "legal" source address. Moreover, on many deployed router architectures the overhead of ingress filter becomes prohibitive on high-speed links.

The principal problem with ingress filtering is that its effectiveness depends on widespread, if not universal, deployment. Unfortunately, a significant fraction of ISPs, perhaps the majority, do not implement this service – either because they are uninformed or have been discouraged by the administrative burden[1], potential router overhead and complications with existing services that depend on source address spoofing (e.g. some versions of Mobile IP [33] and some hybrid satellite communications architectures). A secondary problem is that even if ingress filtering were universally deployed at the customer-to-ISP level, attackers could still forge addresses from the hundreds or thousands of hosts within a valid customer network [14].

It is clear that wider use of ingress filtering would dramatically improve the Internet's robustness to denial-of-service attacks. At the same time it is prudent to assume that such a system will never be fullproof – and therefore traceback technologies will continue to be important.

2.2 Link testing

Most existing traceback techniques start from the router closest to the victim and interactively test its upstream links until they determine which one is used to carry the attacker's traffic. Ideally, this procedure is repeated recursively on the upstream router until the source is reached. This technique assumes that an attack remains active until the completion of a trace and is therefore inappropriate for attacks that are detected after the fact, attacks that occur intermittently, or attacks that modulate their behavior in response to a traceback (it is prudent to assume the attacker is fully informed). Below we describe two varieties of link testing schemes, *input debugging* and *controlled flooding*.

2.2.1 Input debugging

Many routers include a feature called *input debugging*, that allows an operator to filter particular packets on some egress port and determine which ingress port they arrived on. This capability is used to implement a trace as follows: First, the victim must recognize that it is being attacked and develop an *attack signature* that describes a common feature contained in all the attack packets. The victim communicates this signature to a network operator, frequently via telephone, who then installs a corresponding input debugging filter on the victim's upstream egress port. This filter reveals the associated input port, and hence which upstream router originated the traffic. The process is then repeated recursively on the upstream router, until the originating site is reached or the trace leaves the ISP's border (and hence its administrative control over the routers). In the later case, the upstream ISP must be contacted and the procedure repeats itself. While such tracing is frequently performed manually, several ISPs have developed tools to automatically trace attacks across their own networks [41].

The most obvious problem with the input debugging approach, even with automated tools, is its considerable management overhead. Communicating and coordinating with network operators at multiple ISPs requires the time, attention and commitment of both the victim and the remote personnel – many of whom have no direct economic incentive to provide aid. If the appropriate network

[1] Some modern routers ease the administrative burden of ingress filtering by providing functionality to automatically check source addresses against the destination-based routing tables (e.g. `ip verify unicast reverse-path` on Cisco's IOS). This approach is only valid if the route to and from the customer is symmetric – generally at the border of single-homed stub networks.

	Management overhead	Network overhead	Router overhead	Distributed capability	Post-mortem capability	Preventative/ reactive
Ingress filtering	Moderate	Low	Moderate	N/A	N/A	Preventative
Link testing						
Input debugging	High	Low	High	Good	Poor	Reactive
Controlled flooding	Low	High	Low	Poor	Poor	Reactive
Logging	High	Low	High	Excellent	Excellent	Reactive
ICMP Traceback	Low	Low	Low	Good	Excellent	Reactive
Marking	Low	Low	Low	Good	Excellent	Reactive

Table 1: Qualitative comparison of existing schemes for combating anonymous attacks and the probabilistic marking approach we propose.

operators are not available, if they are unwilling to assist, or if they do not have the appropriate technical skills and capabilities, then a traceback may be slow or impossible to complete [21].

2.2.2 *Controlled flooding*

Burch and Cheswick have developed a link testing traceback technique that does not require any support from network operators [6]. We call this technique *controlled flooding* because it tests links by flooding them with large bursts of traffic and observing how this perturbs traffic from the attacker. Using a pre-generated "map" of Internet topology, the victim coerces selected hosts along the upstream route into iteratively flooding each incoming link on the router closest to the victim. Since router buffers are shared, packets traveling across the loaded link – including any sent by the attacker – have an increased probability of being dropped. By observing changes in the rate of packets received from the attacker, the victim can therefore infer which link they arrived from. As with other link testing schemes, the basic procedure is then applied recursively on the next upstream router until the source is reached.

While the scheme is both ingenious and pragmatic, it has several drawbacks and limitations. Most problematic among these is that controlled flooding is itself a denial-of-service attack – exploiting vulnerabilities in unsuspecting hosts to achieve its ends. This drawback alone makes it unsuitable for routine use. Also, controlled flooding requires the victim to have a good topological map of large sections of the Internet in addition to an associated list of "willing" flooding hosts. As Burch and Cheswick note, controlled flooding is also poorly suited for tracing distributed denial-of-service attacks because the link-testing mechanism is inherently noisy and it can be difficult to discern the set of paths being exploited when multiple upstream links are contributing to the attack. Finally, like all link-testing schemes, controlled flooding is only effective at tracing an on-going attack and cannot be used "post-mortem".

2.3 Logging

An approach suggested in [37] and [41] is to log packets at key routers and then use data mining techniques to determine the path that the packets traversed. This scheme has the useful property that it can trace an attack long after the attack has completed. However, it also has obvious drawbacks, including potentially enormous resource requirements (possibly addressed by sampling) and a large scale inter-provider database integration problem. We are unaware of any commercial organizations using a fully operational traceback approach based on logging[2].

[2]Historically, the T3-NFSNET *did* log network-to-network traffic statistics and these were used on at least one occasion to trace IP spoofing attacks to an upstream provider [43].

2.4 ICMP Traceback

Since the first writing of this paper, a new traceback proposal bas emerged based on the use of explicit router-generated ICMP traceback messages [4]. The principle idea in this scheme is for every router to sample, with low probability (e.g., 1/20,000), one of the packets it is forwarding and copy the contents into a special ICMP traceback message including information about the adjacent routers along the path to the destination. During a flooding-style attack, the victim host can then use these messages to reconstruct a path back to the attacker. This scheme has many benefits compared to previous work and is in many ways similar to the packet marking approach we have taken. However, there are several disadvantages in the current design that complicate its use. Among these: ICMP traffic is increasingly differentiated and may be filtered or rate limited differently from normal traffic, the ICMP Traceback message relies on an input debugging capability (i.e. the ability to associate a packet with the input port and/or MAC address on which it arrived) that is not available in some router architectures, if only some of the routers participate it seems difficult to positively "connect" traceback messages from participating routers separated by a non-participating router, and finally, it requires a key distribution infrastructure to deal with the problem of attackers sending false ICMP Traceback messages. That said, we believe that the scheme is promising and that hybrid approaches combining it with some of the algorithms we propose are likely to be quite effective.

3. OVERVIEW

Burch and Cheswick mention the possibility of tracing flooding attacks by "marking" packets, either probabilistically or deterministically, with the addresses of the routers they traverse [6]. The victim uses the information in the marked packets to trace an attack back to its source. This approach has not been previously explored in any depth, but has many potential advantages. It does not require interactive cooperation with ISPs and therefore avoids the high management overhead of input debugging. Unlike controlled flooding, it does not require significant additional network traffic and can potentially be used to track multiple attacks. Moreover, like logging, packet marking can be used to trace attacks "post-mortem" – long after the attack has stopped. Finally, we have found that marking algorithms can be implemented without incurring any significant overhead on network routers. The remainder of this paper focuses on fully exploring and characterizing this approach.

3.1 Definitions

Figure 1 depicts the network as seen from a victim V. For the purposes of this paper, V may be a single host under attack, or a network border device such as a firewall or intrusion detection system that represents many such hosts. Every potential *attack origin* A_i is a leaf in a tree rooted at V and every router R_i is an internal

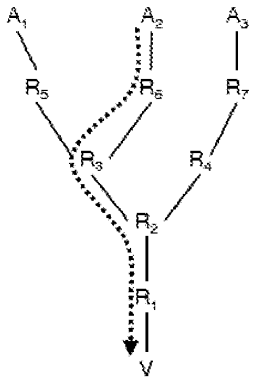


Figure 1: Network as seen from the victim of an attack, V. Routers are represented by R_i, and potential attackers by A_i. The dotted line represents a particular *attack path* between an attacker and the victim.

node along a path between some A_i and V. The *attack path* from A_i is the unique ordered list of routers between A_i and V. For instance, if an attack originates from A_2 then to reach V it must first traverse the path R_6, R_3, R_2, and R_1 – as shown by the dotted line in Figure 1.

The *exact traceback* problem is to determine the attack path and the associated attack origin for each attacker. However, solving this problem is complicated by several practical limitations. The exact attack origin may never be revealed (even MAC source addresses may be spoofed) and a wily attacker may send false signals to "invent" additional routers in the traceback path. We address these issues in section 6, but for now we restrict our discussion to solving a more limited problem. We define the *approximate traceback* problem as finding a candidate attack path for each attacker that contains the true attack path as a suffix. We call this the *valid suffix* of the candidate path. For example, $(R_5, R_6, R_3, R_2, R_1)$ is a valid approximate solution to Figure 1 because it contains the true attack path as a suffix. We say a solution to this problem is *robust* if an attacker cannot prevent the victim from discovering candidate paths containing the valid suffix.

All marking algorithms have two components: a *marking procedure* executed by routers in the network and a *path reconstruction procedure* implemented by the victim. A router "marks" one or more packets by augmenting them with additional information about the path they are traveling. The victim attempts to reconstruct the attack path using only the information in these marked packets. The *convergence time* of an algorithm is the number of packets that the victim must observe to reconstruct the attack path.

3.2 Basic assumptions

The design space of possible marking algorithms is large, and to place our work in context we identify the assumptions that motivate and constrain our design:

- an attacker may generate any packet,
- multiple attackers may conspire,
- attackers may be aware they are being traced,
- packets may be lost or reordered,
- attackers send numerous packets,
- the route between attacker and victim is fairly stable,
- routers are both CPU and memory limited, and
- routers are not widely compromised.

The first four assumptions represent conservative assessments of the abilities of the modern attackers and limitations of the network. Designing a traceback system for the Internet environment is extremely challenging because there is very little that can be trusted. In particular, the attacker's ability to create arbitrary packets significantly constrains potential solutions. When a router receives a packet, it has no way to tell whether that packet has been marked by an upstream router or if the attacker simply has forged this information. In fact, the only invariant that we can depend on is that a packet from the attacker must traverse all of the routers between it and the victim.

The remaining assumptions reflect the basis for our design and deserve additional discussion. First, denial-of-service attacks are only effective so long as they occupy the resources of the victim. Consequently, most attacks are comprised of thousands or millions of packets. Our approach relies on this property because we mark each packet with only a small piece of path state and the victim must observe many such packets to reconstruct the complete path back the the attacker. If many attacks emerge that require only a single packet to disable a host (e.g. ping-of-death [11]), then this assumption may not hold (although we note that even these attacks require multiple packets to *keep* a machine down).

Second, measurement evidence suggests that while Internet routes do change, it is extremely rare for packets to follow many different paths over the short time-scales of a traceback operation (seconds in our system) [32]. This assumption greatly simplifies the role of the victim, since it can therefore limit its consideration to a single primary path for each attacker. If the Internet evolves to allow significant degrees of multi-path routing then this assumption may not hold.

Third, while there have been considerable improvements in router implementation technology, link speeds have also increased dramatically. Consequently, we assert that any viable implementation must have low per-packet overhead and must not require per-flow state. Significantly simpler schemes than ours can be implemented if we assume that routers are not resource constrained.

Finally, since a compromised router can effectively eliminate any information provided by upstream routers, it is effectively indistinguishable from an attacker. In such circumstances, the security violation at the router must be addressed first, before any further traceback is attempted. In normal circumstances, we believe this is an acceptable design point. However, if non-malicious, but information hiding, routing infrastructures become popular, such as described in [22, 35], then this issue may need to be revisited.

4. BASIC MARKING ALGORITHMS

In this section we describe a series of marking algorithms – starting from the most simple and advancing in complexity. Each algorithm attempts to solve the approximate traceback problem in a manner consistent with our assumptions.

Marking procedure at router R:
for each packet w, append R to w

Path reconstruction procedure at victim v:
for any packet w from attacker
extract path $(R_i..R_j)$ from the suffix of w

Figure 2: Node append algorithm.

4.1 Node append

The simplest marking algorithm – conceptually similar to the IP Record Route option [34] – is to append each node's address to the end of the packet as it travels through the network from attacker to victim (see Figure 2). Consequently, every packet received by the victim arrives with a complete ordered list of the routers it traversed – a built-in attack path.

The node append algorithm is both robust and extremely quick to converge (a single packet), however it has several serious limitations. Principal among these is the infeasibly high router overhead incurred by appending data to packets in flight. Moreover, since the length of the path is not known *a priori*, it is impossible to ensure that there is sufficient unused space in the packet for the complete list. This can lead to unnecessary fragmentation and bad interactions with services such as MTU discovery [30]. This problem cannot be solved by reserving "enough" space, as the attacker can completely fill any such space with false, or misleading, path information.

4.2 Node sampling

To reduce both the router overhead and the per-packet space requirement, we can sample the path one node at a time instead of recording the entire path. A single static "node" field is reserved in the packet header – large enough to hold a single router address (i.e. 32 bits for IPv4). Upon receiving a packet, each router chooses to write its address in the node field with some probability p. After enough packets have been sent, the victim will have received at least one sample for every router in the attack path. As stated in section 3, we assume that the attacker sends enough packets and the route is stable enough that this sampling can converge.

Although it might seem impossible to reconstruct an ordered path given only an unordered collection of node samples, it turns out that with a sufficient number of trials, the order can be deduced from the relative number of samples per node. Since routers are arranged serially, the probability that a packet will be marked by a router and then left unmolested by all downstream routers is a strictly decreasing function of the distance to the victim. If we constrain p to be identical at each router, then the probability of receiving a marked packet from a router d hops away is $p(1-p)^{d-1}$. Since this function is monotonic in the distance from the victim, ranking each router by the number of samples it contributes will tend to produce the accurate attack path. The full algorithm is shown in Figure 3.

Putting aside for the moment the difficulty in changing the IP header to add a 32-bit node field, this algorithm is efficient to implement because it only requires the addition of a write and checksum update to the forwarding path. Current high-speed routers already must perform these operations efficiently to update the *time-to-live* field on each hop. Moreover, if $p > 0.5$ then this algorithm is robust against a single attacker because there is no way for an attacker to insert a "false" router into the path's valid suffix by contributing

Marking procedure at router R:
for each packet w
let x be a random number from [0..1)
if $x < p$ then,
write R into w.node

Path reconstruction procedure at victim v:
let $NodeTbl$ be a table of tuples (node,count)
for each packet w from attacker
$z :=$ lookup w.node in $NodeTbl$
if z != NIL then
increment z.count
else
insert tuple (w.node,1) in $NodeTbl$
sort $NodeTbl$ by count
extract path $(R_i..R_j)$ from ordered node fields in $NodeTbl$

Figure 3: Node sampling algorithm.

more samples than a downstream router, nor to reorder valid routers in the path by contributing more samples than the difference between any two downstream routers.

However, there are also two serious limitations. First, inferring the total router order from the distribution of samples is a slow process. Routers far away from the victim contribute relatively few samples (especially since p must be large) and random variability can easily lead to misordering unless a very large number of samples are observed. For instance, if $d = 15$ and $p = 0.51$, the receiver must receive more than 42,000 packets on average before it receives a *single* sample from the furthest router. To guarantee that the order is correct with 95% certainty requires more than seven times that number.

Second, if there are multiple attackers then multiple routers may exist at the same distance – and hence be sampled with the sample probability. Therefore, this technique is not robust against multiple attackers.

4.3 Edge sampling

A straightforward solution to these problems is to explicitly encode *edges* in the attack path rather than simply individual nodes. To do this, we would need to reserve *two* static address-sized fields, *start* and *end*, in each packet to represent the routers at each end of a link, as well as an additional small field to represent the distance of an edge sample from the victim.

When a router decides to mark a packet, it writes its own address into the start field and writes a zero into the distance field. Otherwise, if the distance field is already zero this indicates that the packet was marked by the previous router. In this case, the router writes its own address into the end field – thereby representing the edge between itself and the previous router. Finally, if the router doesn't mark the packet then it always increments the distance field. This somewhat baroque signaling mechanism allows edge sampling to be incrementally deployed – edges are constructed only between participating routers.

The mandatory increment is *critical* to minimize spoofing by an attacker. When the packet arrives at the victim its distance field represents the number of hops traversed since the edge it contains

Marking procedure at router R:

```
for each packet w
    let x be a random number from [0..1)
    if x < p then
        write R into w.start and 0 into w.distance
    else
        if w.distance = 0 then
            write R into w.end
        increment w.distance
```

Path reconstruction procedure at victim v:

```
let G be a tree with root v
let edges in G be tuples (start,end,distance)
for each packet w from attacker
    if w.distance = 0 then
        insert edge (w.start,v,0) into G
    else
        insert edge (w.start,w.end,w.distance) into G
remove any edge (x,y,d) with d ≠ distance from x to v in G
extract path (R_i..R_j) by enumerating acyclic paths in G
```

Figure 4: Edge sampling algorithm.

was sampled.[3] Any packets written by the attacker will necessarily have a distance greater or equal to the length of the true attack path. Therefore, a single attacker is unable to forge any edge between themselves and the victim (for a distributed attack, of course, this applies only to the closest attacker) and the victim does not have to worry about "chaff" while reconstructing the valid suffix of the attack path. Consequently, since we no longer use the sampling rank approach to distinguish "false" samples, we are free to use arbitrary values for the marking probability p.

The victim uses the edges sampled in these packets to create a graph (much as in Figure 1) leading back to the source, or sources, of attack. The full algorithm is described in Figure 4. Because the probability of receiving a sample is geometrically smaller the further away it is from the victim, the time for this algorithm to converge is dominated by the the time to receive a sample from the furthest router, $\frac{1}{p(1-p)^{d-1}}$ in expectation, for a router d hops away. However, there is a small probability that we will receive a sample from the furthest router, but not from some nearer router. We can bound this effect to a factor of $ln(d)$ by the following argument: We conservatively assume that samples from all of the d routers appear with the same likelihood as the furthest router. Since these probabilities are disjoint, the probability that a given packet will deliver a sample from some router is at least $dp(1-p)^{d-1}$. Finally, as per the well-known *coupon collector* problem, the number of trials required to select one of each of d equi-probable items is $d(ln(d) + O(1))$ [19]. Therefore, the number of packets, X, required for the victim to reconstruct a path of length d has the following bounded expectation:

$$E(X) < \frac{ln(d)}{p(1-p)^{d-1}}$$

For example, if $p = \frac{1}{10}$, and the attack path has a length of 10, then a victim can typically reconstruct this path after receiving 75 packets from the attacker. While this choice of $p = \frac{1}{d}$ is optimal, the convergence time is not overly sensitive to this parameter for the path lengths that occur in the Internet. So long as $p \leq \frac{1}{d}$, the results are generally within a small constant of optimal. In the rest of this paper we will use $p = \frac{1}{25}$ since few paths exceed this length [7, 42, 17]. For comparison, the previous example converges with only 108 packets using $p = \frac{1}{25}$.

This same algorithm can efficiently discern multiple attacks because attackers from different sources produce disjoint edges in the tree structure used during reconstruction. The number of packets needed to reconstruct each path is independent, so the number of packets needed to reconstruct all paths is a linear function of the number of attackers. Finally, edge sampling is also robust. That is it is impossible for any edge closer than the closest attacker to be spoofed, due to the robust distance determination. Conversely, in a distributed attack this also means that it is impossible to trust the contents of any edge *further* away than the closest attacker. As with the ICMP Traceback approach [4], an additional mechanism incorporating a shared secret is required to completely address the problem of attackers spoofing edges.

Of course, a significant practical limitation of this approach is that it requires additional space in the IP packet header and therefore is not backwards compatible. In the next section we discuss a modified version of edge-sampling that addresses this problem, albeit at some cost in performance and a reduction in robustness during large distributed attacks.

5. ENCODING ISSUES

The edge sampling algorithm requires 72 bits of space in every IP packet (two 32-bit IP addresses and 8 bits for distance to represent the theoretical maximum number of hops allowed using IP). It would be possible to directly encode these values into an MPLS label stack [36], to enable traceback within a single homogeneous ISP network. However, our focus is on a heterogeneous environment based purely on IP datagrams. One obvious approach is to store the edge sample data in an IP option, but this is a poor choice for many of the same reasons that the node append algorithm is infeasible – appending additional data to a packet in flight is expensive and there may not be sufficient space to append this data. We could also send this data out-of-band – in a separate packet – but this would add both router and network overhead plus the complexity of a new and incompatible protocol.

Instead, we have developed a modified version of edge sampling that dramatically reduces the space requirement in return for a modest increase in convergence time and a reduction in robustness to multiple attackers. Following an analysis of our algorithm we explore the practical implementation issues and discuss one concrete encoding of this scheme based on overloading the 16-bit IP *identification* field used for fragmentation. Any solution involving such overloading necessarily requires compromises and we stress that our solution reflects only one design point among many potential implementation tradeoffs for this class of algorithm and *does not* necessarily reflect an optimal balance among them.

5.1 Compressed edge fragment sampling

We use three techniques to reduce per-packet storage requirements while preserving robustness. First, we encode each edge in half the space by representing it as the *exclusive-or* (XOR) of the two IP addresses making up the edge, as depicted in Figure 5. When

[3] It is important that distance field is updated using a saturating addition. If the distance field were allowed to wrap, then the attacker could spoof edges close to the victim by sending packets with a distance value close to the maximum.

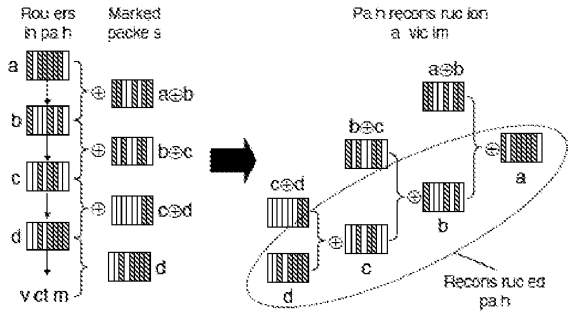


Figure 5: Edge data can be communicated in half the space by sending the XOR of the two nodes (i.e. router IP addresses) making up an edge, rather than sending each node separately. Over time the victim receives the messages d, $c \oplus d$, $b \oplus c$, and $a \oplus b$. By XORing these messages together the original path can be reconstructed.

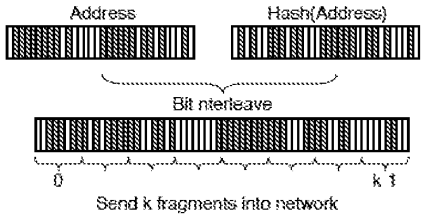


Figure 6: Each router calculates a uniform hash of its IP address once, at startup, using a well-known function. This hash is interleaved with the original IP address (the original address on odd bits, the hash on even bits). The resulting quantity is then broken into k fragments, which the router selects among randomly when marking a packet. Although it is not shown, each of these fragments is further labeled with its offset. The next downstream router uses this offset to select the appropriate fragment to XOR – thereby encoding part of an edge.

some router decides to mark a packet it writes its address, a, into the packet. The following router, b, notices that the distance field is 0 and (assuming it does not mark the packet itself) reads a from the packet, XORs this value with its own address and writes the resulting value, $a \oplus b$, into the packet. We call the resulting value the *edge-id* for the edge between a and b. The edge-ids in the packets received by the victim always contain the XOR of two adjacent routers, except for samples from routers one hop away from the victim, which arrive unmodified. Since $b \oplus a \oplus b = a$, marked packets from the final router can be used to decode the previous edge id, and so on, hop-by-hop until we reach the first router.

Our second modification further reduces our per-packet space requirements by subdividing each edge-id into some number, k, of smaller non-overlapping fragments. When a router decides to mark a packet, it selects one of these fragments at random and stores it in the packet. We use a few additional bits ($log_2 k$) to store the offset of this fragment within the original address – this is necessary to ensure that both fragments making up an edge-id are taken from the same offset. If enough packets are sent by the attacker, the victim will eventually receive all fragments from all edge-ids.

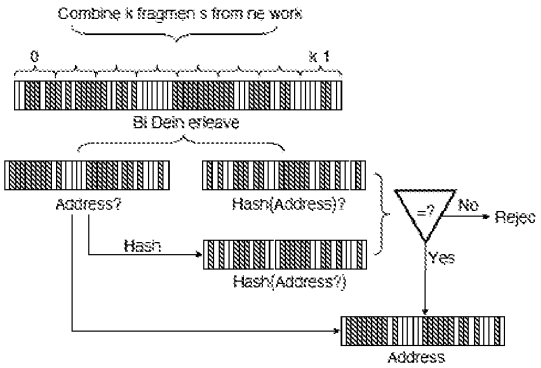


Figure 7: When reconstructing a candidate edge, the victim combines k fragments to produce a bit string. By deinterleaving this string, the address portion and the hash portion are extracted. We recalculate the hash over this address portion using the same hash function used by the router. If the resulting hash is the same as the hash portion extracted, then the address is accepted as valid. This procedure protects against accidentally combining fragments of different edges.

Finally, unlike full IP addresses, edge-id fragments are not unique and multiple fragments from different edge-ids may have the same value. If there are multiple attackers, a victim may receive multiple edge fragments with the same offset and distance. To reduce the probability that we accidentally reconstruct a "false" edge-id by combining fragments from different paths, we add a simple error detection code to our algorithm. We *increase* the size of each router address, and hence each edge-id, by bit-interleaving its IP address with a random hash of itself (depicted in Figure 6). As described earlier, this value is split into fragments, each fragment is selected randomly and stored with an offset, and downstream routers use XOR to combine fragments at the same offset to make up edge-id fragments. The victim constructs *candidate edge-ids* by combining all combinations of fragments at each distance with disjoint offset values. As shown in Figure 7, a candidate edge-id is only accepted if the hash portion matches the data portion for each of its two nodes. As we increase the size of the hash, the probability of a collision is reduced. We provide a describe the full procedure in Figure 8.

The expected number of packets for this algorithm to converge is similar to the edge sampling approach, except now we need k fragments for each edge-id, rather than just one, a total of kd fragments. If we again assume conservatively that each of these fragments is delivered equi-probably with probability $p(1-p)^{d-1}$, the expected number of packets required for path reconstruction is bounded by:

$$E(X) < \frac{k \cdot ln(kd)}{p(1-p)^{d-1}}$$

For example, if there are 8 fragments per edge-id, an attacker is 10 hops away, and $p = \frac{1}{25}$, then a victim can reconstruct the full path after receiving slightly less than 1,300 packets on average. Using techniques similar to those used to show sharp concentration results for the coupon collectors problem, we can further show that the approximate the number of packets required to ensure that a path

Marking procedure at router R:

let R' = BitInterleave(R, Hash(R))
let k be the number of non-overlapping fragments in R'
for each packet w
  let x be a random number from [0..1)
  if $x < p$ then
    let o be a random integer from [0..$k-1$]
    let f be the fragment of R' at offset o
    write f into w.frag
    write 0 into w.distance
    write o into w.offset
  else
    if w.distance = 0 then
      let f be the fragment of R' at offset w.offset
      write $f \oplus w$.frag into w.frag
    increment w.distance

Path reconstruction procedure at victim v:

let $FragTbl$ be a table of tuples (frag,offset,distance)
let G be a tree with root v
let edges in G be tuples (start,end,distance)
let $maxd := 0$
let $last := v$
for each packet w from attacker
  $FragTbl$.Insert(w.frag,w.offset,w.distance)
  if w.distance > $maxd$ then
    $maxd := w$.distance
for $d := 0$ to $maxd$
  for all ordered combinations of fragments at distance d
    construct edge z
    if $d \neq 0$ then
      $z := z \oplus last$
    if Hash(EvenBits(z)) = OddBits(z) then
      insert edge (z,EvenBits(z),d) into G
      $last$:= EvenBits(z);
remove any edge (x,y,d) with $d \neq$ distance from x to v in G
extract path (R_i..R_j) by enumerating acyclic paths in G

Figure 8: Compressed edge fragment sampling algorithm.

can be reconstructed with probability $1 - \frac{1}{c}$ is:

$$\frac{k \cdot ln(kdc)}{p(1-p)^{d-1}}$$

packets. To completely reconstruct the previous path with 95% certainty should require no more than 2150 packets. Many denial-of-service attacks send this many packets in a few seconds.

Finally, we explore the robustness of this algorithm with respect to multiple attackers. For a random hash of length h, the probability of accepting an arbitrarily constructed candidate edge-id is $\frac{1}{2^h}$. In the event that there are m attackers, then at any particular distance d, in the worst case there may be up to m distinct routers.[4] Consequently the probability that any edge-id at distance d is accepted incorrectly is at most:

$$1 - (1 - \frac{1}{2^h})^{m^k}$$

[4]In practice, the number of distinct routers is likely to be smaller for the portion of the path closest to the receiver, since many attackers will still share significant portions of their attack path with one another.

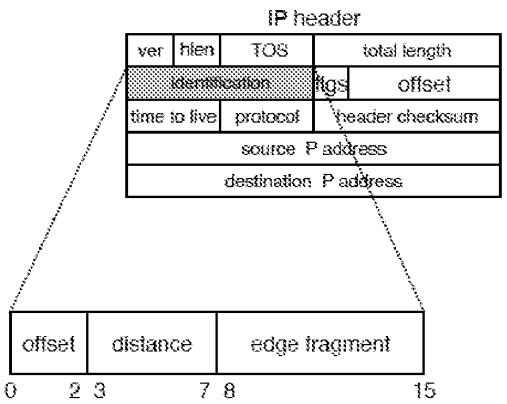


Figure 9: Encoding edge fragments into the IP identification field.

since there are m^k possible combinations of fragments in the worst case. For $h = 32$ and $k = 4$ this means that 100 distinct routers at the same distance (i.e. disjoint attack paths) will be resolved with no errors with a probability of better than 97%. For $h = 32$ and $k = 8$, (the values we use for our implementation) the same certainty can only be provided for 10 distinct routers at the same distance. Our use of the XOR function further complicates reconstruction since all combinations of XOR values must be tried as attack paths diverge. This is somewhat mitigated as the probability of propagating an error from a single edge all the way to the attacker is is extremely small because the resulting edge-id, when XORed with the previous edge-id, must again produce a correct hash.

The most significant drawback to this scheme is the large number of combinations that must be considered as the multiple attack paths diverge. While these combinations can be computed off-line, for large values of k and m even this can become intractable. For example, even with $k = 8$ and $m = 10$, if the separate attack paths diverge such that there are 10 completely independent edges per attacker, this will require roughly a billion combinations to be considered. Consequently, there is a design tension in the size of k – per-packet space overhead is reduced by a larger k, while computational overhead and robustness benefits from a smaller k.

5.2 IP header encoding

To allow for practical deployment requires that we "overload" existing header fields in a manner that will have minimal impact on existing users. This is a difficult task, especially given that even after prodigious effort we require 16 bits of space. Nonetheless, we believe it possible to obtain this space by overloading the 16-bit IP identification field. This field is currently used to differentiate IP fragments that belong to different packets. We describe our proposed encoding below, and then discuss the issues of backwards-compatibility that it raises. However, we note that because the issue of backwards-compatible encoding is largely separate from our traceback algorithms, we could adopt any reasonable encoding that comes to light.

Figure 9 depicts our choice for partitioning the identification field: 3 offset bits to represent 8 possible fragments, 5 bits to represent

the distance, and 8 bits for the edge fragment. We use a 32-bit hash, which doubles the size of each router address to 64 bits. This implies that 8 separate fragments are needed to represent each edge – each fragment indicated by a unique offset value. Finally, 5 bits is sufficient to represent 31 hops, which is more than almost all Internet paths [7, 42, 17].[5]

The observant reader will note that this layout is chosen to allow the highest performance software implementation of our algorithm, which already had a low per-packet router overhead. In the common case, the only modification to the packet is to increment its distance field. Because of its alignment within the packet, this increment precisely offsets the required decrement of the time-to-live field implemented by each router [1]. Consequently, the header checksum *does not need to be altered at all* and the header manipulation overhead could be even lower than in current software-based routers – simply an addition to the distance field, a decrement to the TTL field, and a comparison to check if either has overflowed. In the worst case, our algorithm must read the IP identification field, lookup an edge fragment and XOR it, and fold the write-back into the existing checksum update procedure (a few ALU operations). Of course, for modern ASIC-based routers these optimizations are unnecessary.

As we reuse of the IP identification field, we must address issues of backwards-compatibility for IP fragment traffic. Ultimately, there is no perfect solution to this problem and we are forced to make compromises that disadvantage fragmented traffic. Fortunately, recent measurements suggest that less than 0.25% of packets are fragmented [40, 10]. Moreover, it has long been understood that network-layer fragmentation is detrimental to end-to-end performance [27] so modern network stacks implement automatic MTU discovery to prevent fragmentation regardless of the underlying media [30]. Consequently, we believe that our encoding will interoperate seamlessly with existing protocol implementations in the vast majority of cases.

However, there is a small but real fraction of legitimate traffic that is fragmented, and we wish to ensure that it is not affected by our modifications to the extent that this is possible. Normally if a packet is fragmented, its identification field is copied to each fragment so the receiver can faithfully reassemble the fragments into the original packet. Our marking procedure can violate this property in one of two ways: by writing different values into the identification fields of fragments from the same datagram or by writing the same values into the identification fields of fragments from different datagrams. These two problems present different challenges and have different solutions.

First, a datagram may be fragmented *upstream* from a marking router. If the fragment is subsequently marked and future fragments from the same datagram are not marked consistently then reassembly may fail or data may be corrupted. While the simplest solution to this problem is to simply not mark fragments, an adversary would quickly learn to evade traceback by exploiting this limitation. In fact, some current denial-of-service attacks already use IP fragments to exploit errors in host IP reassembly functions [12]. Instead, we propose an alternative marking mechanism for fragments. We use a separate marking probability, q, for fragments. When we decide to mark a fragment, we prepend a new ICMP "echo reply" header, along with the *full* edge data – truncating the tail of the packet. This ICMP packet is considered "marked" and its distance field is set to zero, thereby guaranteeing that the distance field reflects the number of edges traversed on the way to the victim. The packet is consequently "lost" from the standpoint of the receiver, but the edge information is delivered in a way that does not impact legacy hosts. Because we can use the full edge sampling algorithm, q can be more than an order of magnitude smaller than p and yet achieve the same convergence time. This solution increases the loss rate of fragmented flows somewhat (more substantially for longer paths) but preserves the integrity of the data in these flows.

[5] It is also reasonable to turn off marking on any routers that cannot be directly connected to an attacking host (e.g. core routers). This both reduces the convergence time, and increases the "reach" of the distance field.

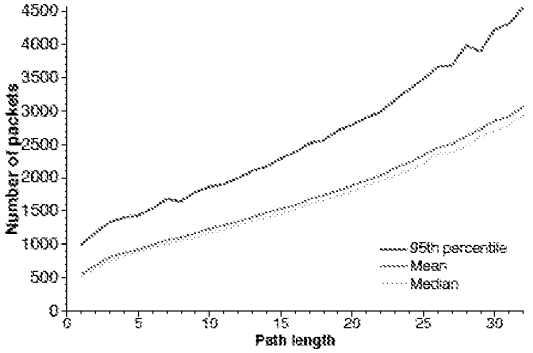


Figure 10: Experimental results for number of packets needed to reconstruct paths of varying lengths. The marking probability, p, is set to $\frac{1}{25}$. Each path length result represents the results of 1,000 independent simulation runs.

A more insidious problem is presented by fragmentation that occurs *downstream* from a marking router. If a marked packet is fragmented, but one of the fragments is lost, then the remaining fragments may linger in the victim's reassembly buffer for an extended period [5]. Future packets marked by the same router can have the same IP identification value and consequently may be incorrectly reassembled with the previous fragments. One possibility is to leave this problem to be dealt with by higher layer checksums. However, not all higher layer protocols employ checksums, and in any case it is dangerous to rely on such checksums because they are typically designed only for low residual error rates. Another solution is to set the *Don't Fragment* flag on every marked packet. Along rare paths that require fragmentation, this solution will degrade communication between hosts not using MTU path discovery, and may filter marked packets if a reduced MTU edge is close to the victim, but it will never lead to data corruption.

5.3 Experience

We have implemented the marking and reconstruction portions of our algorithm and have tested it using a simulator that creates random paths and originates attacks. In Figure 10 we graph the mean, median and 95th percentile for the number of packets required to reconstruct paths of varying lengths over 1,000 random test runs for each length value. We assume a marking probability of $\frac{1}{25}$. Note that while the convergence time is theoretically exponential in the path length, all three lines appear linear due to the finite path length and appropriate choice of marking probability.

We see that most paths can be resolved with between one and two

thousand packets, and even the longest paths can be resolved with a very high likelihood within four thousand packets. To put these numbers in context, most flooding-style denial of service attacks send many hundreds or thousands of packets each second. The analytic bounds we described earlier are conservative, but in our experience they are no more than 30% higher than our experimental results.

6. LIMITATIONS AND FUTURE WORK

There are still a number of limitations and loose ends in our approach. We discuss the most important of these here:

- backwards compatibility,
- distributed attacks,
- path validation, and
- approaches for determining the attack origin.

6.1 Backwards compatibility

The IP header encoding as we have described it has several practical limitations. It negatively impacts users that require fragmented IP datagrams and is currently incompatible with parts of IPsec [28] (the Authentication Header provides cryptographic protection for the identification field and therefore the field cannot be safely modified by routers). These problems are hardly unique to traceback and are inherent limitations that come about from attempting to coexist with or co-opt protocol features that did not anticipate a new use. One way to address this issue, originally proposed by John Hawkinson, is to selectively enable traceback support in response to operational needs. A "request for traceback" from a particular network could be encoded as a BGP attribute in the network's route advertisement. Routers receiving such an advertisement would enable traceback support on packets destined for that network. Since a network requesting such support is presumably already suffering under an attack, any minor service degradation for fragmented or authenticated flows would be acceptable.

Finally, our scheme does not address implementation in IPv6, the proposed successor to IPv4, which does not have an identification field [18]. While we do not attempt to propose a complete encoding here, we believe that the same techniques we have proposed could also be employed within IPv6, perhaps by overloading the 24-bit *flow label* field (without any further modifications this would result in roughly a factor of three increase in the number of packets required to reconstruct a path).

6.2 Distributed attacks

For large distributed attacks, the practical implementation we have described has serious limitations due to the difficulty in correctly grouping fragments together. Consequently, the probability of misattributing an edge, as well as the amount of state needed to evaluate this decision, increases very quickly with the fan-out of an attack. There is significant future work in designing alternative encoding methods that scale their robustness as they receive more data.

6.3 Path validation

Some number of the packets sent by the attacker are unmarked by intervening routers. The victim cannot differentiate between these packets and genuine marked packets. Therefore an attacker could insert "fake" edges by carefully manipulating the identification fields in the packets it sends. While the distance field prevents an attacker from spoofing edges between it and the victim – what we call the *valid suffix* – nothing prevents the attacker from spoofing extra edges past the end of the true attack path.

There are several ways to identify the valid suffix within a path generated by the reconstruction procedure. With minimal knowledge of Internet topology one can differentiate between routers that belong to transit networks (e.g. ISPs) and those which belong to stub networks (e.g. enterprise networks). Generally speaking, a valid path will never enter a stub network and then continue into a transit network. Moreover, simple testing tools such as `traceroute` should enable a victim to determine if two networks do, in fact, connect. More advanced network maps [8, 23] can resolve this issue in an increasing number of cases.

A more general mechanism is to provide each router with a "secret" that is sent along with each marked packet (perhaps in the single unallocated bit in the IP flags field). When the victim wants to validate a router in the path, it contacts the associated network (possibly out of band, via telephone or e-mail) and obtains the secret used by the router at the time of the attack. To guard against replay, the secret can be time-varying and hashed with the packet contents. Since the attacker will not know the router's secret, they will not be able to include the proper bit in their forged edge-id fragments. By eliminating edge-ids for which the secret in their constituent fragments can not be validated, we can prune a candidate attack path to only include the valid suffix.

6.4 Attack origin detection

While our IP-level traceback algorithm could be an important part of the solution for stopping denial-of-service attacks, it is by no means a complete solution. Our algorithm attempts to determine the approximate origin of attack traffic – in particular, the earliest traceback-capable router involved in forwarding attack traffic from the source that directly generated it. As mentioned earlier, there are a number of reasons why this may differ from the true source of the attack: attackers can hide their true identities by "laundering" attacks through third parties, either indirectly (e.g. smurf attacks [13] or DNS reflectors [15]) or directly via compromised "stepping stone" machines or IP-in-IP tunnels. While there is ongoing work on following attackers through intermediate hosts [45, 39], there are still significant challenges in developing a generally applicable and universally deployable solution to this problem. One interesting possibility enabled by the packet marking approach is to extend traceback across "laundering points". For example, identifying marks could be copied from a DNS request packet into the associated DNS reply, thereby allowing the victim to trace the full causal path.

Even in absence of such "laundering", our approach does not reveal the actual host originating the attack. Moreover, since hosts can forge both their IP source address and MAC address the origin of a packet may never be explicitly visible. On shared media such as FDDI rings, this problem can only be solved by explicit testing. However, on point-to-point media, the input port a packet arrives on is frequently enough to determine its true origin. On other media, there may be a MAC address, cell number, channel, or other hint that would help to locate the attack origin. In principle, our algorithm could be modified to report this information by occasionally marking packets with a special edge-id representing a link between the router and the input port on which the packet arrived (or other

"hint" information). We have not explored the design of such a feature in any depth.

Finally, traceback is only effective at finding the source of *attack traffic*, not necessarily the *attacker* themselves. Stopping an attack may be sufficient to eliminate an immediate problem, but long term disincentives may require a legal remedy and therefore the forensic means to determine an attacker's identity. Even with perfect traceback support, unambiguously identifying a sufficiently skilled and paranoid attacker is likely to require cooperation from law enforcement and telecommunications organizations.

7. CONCLUSION

In this paper we have argued that denial-of-service attacks motivate the development of improved traceback capabilities and we have explored traceback algorithms based on packet marking in the network. We have shown that this class of algorithm, best embodied in *edge sampling*, can enable efficient and robust multi-party traceback that can be incrementally deployed and efficiently implemented. As well, we have developed variant algorithms that sacrifice convergence time and robustness for reduced per-packet space requirements. Finally, we have suggested one potential deployment strategy using such an algorithm based on overloading existing IP header fields and we have demonstrated that this implementation is capable of fully tracing an attack after having received only a few thousand packets. We believe our solution represents a valuable first step towards an automated network-wide traceback facility. Several areas remain to be addressed in future work, such as the combination of widely distributed attacks and points of indirection such as reflectors.

Acknowledgments

This paper has benefited from conversations with many different people – far more than can be acknowledged completely here. Still we would like to particularly thank Bill Cheswick and Hal Burch for early access to their work in this area, Sean McCreary and KC Claffy for access to their packet trace data, Alec Wolman for help with jgraph, Przemek Pardyak, Mike Swift, Neil Spring, Gretta Bartels, Robert Grimm and Geoff Voelker for commenting on early drafts of the paper, and constructive feedback from John Hawkinson, Robert Stone, Jeff Mogul, Robert Moskowitz, Greg Minshall, Tony Li, Curtis Villamizar, Steve Corbato, and countless others. Finally, we thank both the anonymous reviewers and our shepherds, Vern Paxson and Craig Partridge, for their efforts.

8. REFERENCES

[1] F. Baker. Requirements for IP Version 4 Routers. RFC 1812, June 1995.

[2] G. Banga, P. Druschel, and J. Mogul. Resource Containers: A New Facility for Resource Management in Server Systems. In *Proceedings of the 1999 USENIX/ACM Symposium on Operating System Design and Implementation*, pages 45–58, Feb. 1999.

[3] S. M. Bellovin. Security Problems in the TCP/IP Protocol Suite. *ACM Computer Communications Review*, 19(2):32–48, Apr. 1989.

[4] S. M. Bellovin. ICMP Traceback Messages. Internet Draft: draft-bellovin-itrace-00.txt, Mar. 2000.

[5] R. Braden. Requirements for Internet Hosts – Communication Layers. RFC 1122, Oct. 1989.

[6] H. Burch and B. Cheswick. Tracing Anonymous Packets to Their Approximate Source. Unpublished paper, Dec. 1999.

[7] R. L. Carter and M. E. Crovella. Dynamic Server Selection Using Dynamic Path Characterization in Wide-Area Networks. In *Proceedings of the 1997 IEEE INFOCOM Conference*, Kobe, Japan, Apr. 1997.

[8] B. Cheswick and H. Burch. Internet Mapping Project. `http://cm.bell-labs.com/who/ches/map/index.html`, 2000.

[9] Cisco Systems. Configuring TCP Intercept (Prevent Denial-of-Service Attacks). Cisco IOS Documentation, Dec. 1997.

[10] K. Claffy and S. McCreary. Sampled Measurements from June 1999 to December 1999 at the AMES Inter-exchange Point. Personal Communication, Jan. 2000.

[11] Computer Emergency Response Team. CERT Advisory CA-96.26 Denial-of-Service Attack via pings. `http://www.cert.org/advisories/CA-96.26.ping.html`, Dec. 1996.

[12] Computer Emergency Response Team. CERT Advisory CA-97.28 IP Denial-of-Service Attacks. `http://www.cert.org/advisories/CA-97.28.smurf.html`, Dec. 1997.

[13] Computer Emergency Response Team. CERT Advisory CA-98.01 "smurf" IP Denial-of-Service Attacks. `http://www.cert.org/advisories/CA-98.01.smurf.html`, Jan. 1998.

[14] Computer Emergency Response Team. CERT Advisory CA-2000-01 Denial-of-Service Developments. `http://www.cert.org/advisories/CA-2000-01.html`, Jan. 2000.

[15] Computer Emergency Response Team. CERT Incident Note IN-2000-04 Denial-of-Service Attacks using Nameservers. `http://www.cert.org/incident_notes/IN-200-04.html`, Apr. 2000.

[16] Computer Security Institute and Federal Bureau of Investigation. 1999 CSI/FBI Computer Crime and Security Survey. Computer Security Institute publication, Mar. 1999.

[17] Cooperative Associationfor Internet Data Analysis. Skitter Analysis. `http://www.caida.org/Tools/Skitter/Summary/`, 2000.

[18] S. Deering. Internet protocol, version 6 (ipv6). RFC 2460, Dec. 1998.

[19] W. Feller. *An Introduction to Probability Theory and Its Applications (2nd edition)*, volume 1. Wiley and Sons, 1966.

[20] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing. RFC 2267, Jan. 1998.

[21] J. Glave. Smurfing Cripples ISPs. Wired Technolgy News: (`http://www.wired.com/news/news/technology/story/9506.html`), Jan. 1998.

[22] I. Goldberg and A. Shostack. Freedom Network 1.0 Architecture and Protocols. Zero-Knowledge Systems White Paper, Nov. 1999.

[23] R. Govindan and H. Tangmunarunkit. Heuristics for Internet Map Discovery. In *Proceedings of the 2000 IEEE INFOCOM Conference*, Tel Aviv, Israel, Mar. 2000.

[24] L. T. Heberlein and M. Bishop. Attack Class: Address Spoofing. In *1996 National Information Systems Security Conference*, pages 371–378, Baltimore, MD, Oct. 1996.

[25] J. D. Howard. *An Analysis of Security Incidents on the Internet*. PhD thesis, Carnegie Mellon University, Aug. 1998.

[26] P. Karn and W. Simpson. Photuris: Session-Key Management Protocol. RFC 2522, Mar. 1999.

[27] C. Kent and J. Mogul. Fragmentation Considered Harmful. In *Proceedings of the 1987 ACM SIGCOMM Conference*, pages 390–401, Stowe, VT, Aug. 1987.

[28] S. Kent and R. Atkinson. Security architecture for the internet protocol. RFC 2401, Nov. 1998.

[29] C. Meadows. A Formal Framework and Evaluation Method for Network Denial of Service. In *Proceedings of the 1999 IEEE Computer Security Foundations Workshop*, Mordano, Italy, June 1999.

[30] J. Mogul and S. Deering. Path MTU Discovery. RFC 1191, Nov. 1990.

[31] R. T. Morris. A Weakness in the 4.2BSD Unix TCP/IP Software. Technical Report Computer Science #117, AT&T Bell Labs, Feb. 1985.

[32] V. Paxson. End-to-End Routing Behavior in the Internet. *IEEE/ACM Transactions on Networking*, 5(5):601–615, Oct. 1997.

[33] C. Perkins. IP Mobility Support. RFC 2002, Oct. 1996.

[34] J. Postel. Internet Protocol. RFC 791, Sept. 1981.

[35] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous Connections and Onion Routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, May 1998.

[36] E. C. Rosen, Y. Rekhter, D. Tappan, D. Farinacci, G. Fedorkow, T. Li, and A. Conta. MPLS Label Stack Encoding. Internet Draft: draft-ietf-mpls-label-encaps-07.txt (expires March 2000), Sept. 1998.

[37] G. Sager. Security Fun with OCxmon and cflowd. Presentation at the Internet 2 Working Group, Nov. 1998.

[38] O. Spatscheck and L. Peterson. Defending Against Denial of Service Attacks in Scout. In *Proceedings of the 1999 USENIX/ACM Symposium on Operating System Design and Implementation*, pages 59–72, Feb. 1999.

[39] S. Staniford-Chen and L. T. Heberlein. Holding Intruders Accountable on the Internet. In *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, pages 39–49, Oakland, CA, May 1995.

[40] I. Stoica and H. Zhang. Providing Guaranteed Services Without Per Flow Management. In *Proceedings of the 1999 ACM SIGCOMM Conference*, pages 81–94, Boston, MA, Aug. 1999.

[41] R. Stone. CenterTrack: An IP Overlay Network for Tracking DoS Floods. In *to appear in Proceedings of thje 2000 USENIX Security Symposium*, Denver, CO, July 2000.

[42] W. Theilmann and K. Rothermel. Dynamic Distance Maps of the Internet. In *Proceedings of the 2000 IEEE INFOCOM Conference*, Tel Aviv, Israel, Mar. 2000.

[43] C. Villamizar. Personal Communication, Feb. 2000.

[44] M. Vivo, E. Carrasco, G. Isern, and G. O. Vivo. A review of port scanning techniques. *ACM Computer Communications Review*, 29(2):41–48, Apr. 1999.

[45] Y. Zhang and V. Paxson. Stepping Stone Detection. In *to appear in Proceedings of thje 2000 USENIX Security Symposium*, Denver, CO, July 2000.

Declaration of Dr. Stefan Savage

ATTACHMENT H



Call for papers
Paper Submission
Organisation Committee
Program Committee
• Conference program
Registration
Local information
Student travel awards

CONFERENCE PROGRAM

Program at a glance Tutorial program Technical program
Final program (PDF) Abstracts Papers Outrageous Opinions
Multicast transmission

Preliminary Technical Program

Preliminary technical program including abstracts in PostScript Format .ps or .ps.gz, Portable Document Format .pdf, or in HTML format.

Introduction Session
Wednesday, Aug 30, 9.00-10.15

- **Welcome**
- **SIGCOMM Award Ceremony**
  André Danthine (University of Liege, Belgium)
- **Mutual Cloning between Network Paradigms**
  André Danthine

Session 1: Multicast and Anycast
Wednesday, Aug 30, 10.45-12.15

- **A Framework for Scalable Global IP-Anycast (GIA)** (abstract, paper)
  Dina Katabi, *John Wroclawski* (Massachusetts Institute of Technology)
- **pgmcc: A TCP-friendly Single-Rate Multicast Congestion Control Scheme** (abstract, paper)
  Luigi Rizzo (Universita di Pisa)
- **Fault Isolation in Multicast Trees** (abstract, paper)
  Anoop Reddy, *Ramesh Govindan*, *Deborah Estrin* (USC/Information Sciences Institute)

Session 2: Control Mechanisms
Wednesday, Aug 30, 13.30-15.00

- **Equation-Based Congestion Control for Unicast Applications** (abstract, paper)
  Sally Floyd, *Mark Handley* (AT&T Center for Internet Research at ICSI), *Jitendra Padhye* (University of Massachusetts, Amherst), *Joerg Widmer* (AT&T Center for Internet Research at ICSI)
- **Endpoint Admission Control: Architectural Issues and Performance** (abstract, paper)
  Lee Breslau (AT&T Research), *Edward Knightly* (Rice University), *Scott Shenker* (AT&T Center for Internet Research at ICSI), *Ion Stoica*, *Hui Zhang* (CMU)
- **Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services** (abstract, paper)
  Zhi-Li Zhang, *Zhenhai Duan* (University of Minnesota), *Lixin Gao* (Smith College), *Yiwei Thomas Hou* (Fujitsu Labs)

Session 3: World Wide Web
Wednesday, Aug 30, 15.30-17.00

- **A Protocol-Independent Technique for Eliminating Redundant Network Traffic** (abstract, paper)
  Neil T. Spring, *David Wetherall* (University of Washington)
- **On Network-Aware Clustering of Web Clients** (abstract, paper)
  Balachander Krishnamurthy (AT&T Research), *Jia Wang* (Cornell University)

Cloudflare, Inc. Exhibit 1003 Page 1178

- **The Content and Access Dynamics of a Busy Web Site: Findings and Implications** (abstract, paper)
  Venkata N. Padmanabhan (Microsoft Research), *Lili Qiu* (Cornell University)

Session 4: Performance Analysis and Modeling
Thursday, Aug 31, 9.00-10.30

- **Critical Path Analysis of TCP Transactions** (abstract, paper)
  Paul Barford, *Mark Crovella* (Boston University)
- **Tuning RED for Web Traffic** (abstract, paper)
  Mikkel Christiansen, *Kevin Jeffay*, *David Ott*, *F. Donelson Smith* (University of North Carolina at Chapel Hill)
- **A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED** (abstract, paper)
  Vishal Misra, *Wei-Bo Gong*, *Don Towsley* (University of Massachusetts, Amherst)

Session 5: Routing Stability and Convergence
Thursday, Aug 31, 11.00-12.00

- **Routing Stability in Congested Networks: Experimentation and Analysis** (abstract, paper)
  Aman Shaikh (University of California, Santa Cruz), *Lampros Kalampoukas*, *Rohit Dube* (Bell Labs, Lucent Technologies), *Anujan Varma* (University of California, Santa Cruz)
- **An Experimental Study of Delayed Internet Routing Convergence** (abstract, paper)
  Craig Labovitz (Microsoft Research), *Abha Ahuja*, *Abhijit Abose*, *Farnam Jahanian* (University of Michigan)

Session 6: Routing and Bridging
Thursday, Aug 31, 13.30-14.30

- **FIRE: Flexible Intra-AS Routing Environment** (abstract, paper)
  Craig Partridge (BBN Technologies), *Alex C. Snoeren* (BBN Technologies & Massachusetts Institute of Technology), *Tim Strayer*, *Beverly Schwartz*, *Matthew Condell*, *Isidro Castineyra* (BBN Technologies)
- **SmartBridge: A Scalable Bridge Architecture** (abstract, paper)
  Thomas L. Rodeheffer, *Chandramohan A. Thekkath* (Compaq Systems Research Center), *Darrell Anderson* (Duke University)

Student Poster Session - Work in Progress
Thursday, Aug 31, 14.30-15.30

- **Student Poster Exhibition**
  Click here for instructions for submission.

Session 7: TCP Analysis
Thursday, Aug 31, 15.30-17.00

- **TCP is Max-Plus Linear** (abstract, paper)
  Francois Baccelli, *Dohy Hong* (ENS-INRIA)
- **A Stochastic Model of TCP/IP with Stationary Random Losses** (abstract, paper)
  Eitan Altman, *Kostia Avrachenkov*, *Chadi Barakat* (INRIA)
- **On the Propagation of Long-range Dependency in the Internet** (abstract, paper)
  Andras Veres (Ericsson), *Zsolt Kenesi*, *Sandor Molnar* (Technical University of Budapest), *Gabor Vattay* (Eotvos University, Budapest)

Cloudflare, Inc. Exhibit 1003 Page 1179

Session 8: Tracing and Measurement
Friday, Sep 1, 9.00-11.00

- **Deriving Traffic Demands for Operational IP Networks: Methodology and Experience** (abstract, paper)
  Anja Feldmann (University of Saarbruecken), *Albert Greenberg*, *Carsten Lund*, *Nick Reingold*, *Jennifer Rexford*, *Fred True* (AT&T Research)
- **Trajectory Sampling for Direct Traffic Observation** (abstract, paper)
  Nick Duffield, *Matthias Grossglauser* (AT&T Research)
- **Measuring Link Bandwidths Using a Deterministic Model of Packet Delay** (abstract, paper)
  Kevin Lai, *Mary Baker* (Stanford University)
- **Practical Network Support For IP Traceback** (abstract, paper)
  Stefan Savage, *David Wetherall*, *Anna Karlin*, *Tom Anderson* (University of Washington)

Session 9: Header Processing
Friday, Sep 1, 11.30-13.00

- **When the CRC and TCP Checksum Disagree** (abstract, paper)
  Jonathan Stone (Stanford University), *Craig Partridge* (BBN Technologies)
- **Packet Types: Abstract Specification of Network Protocol Messages** (abstract, paper)
  Peter J. McCann, *Satish Chandra* (Bell Labs, Lucent Technologies)
- **Memory-Efficient State Lookups with Fast Updates** (abstract, paper)
  Sandeep Sikka (Inktomi), *George Varghese* (UCSD)

Cloudflare, Inc. Exhibit 1003 Page 1180

Declaration of Dr. Stefan Savage

ATTACHMENT I

Association for Computing Machinery

Special Interest Group Data Communication

acm sigcomm

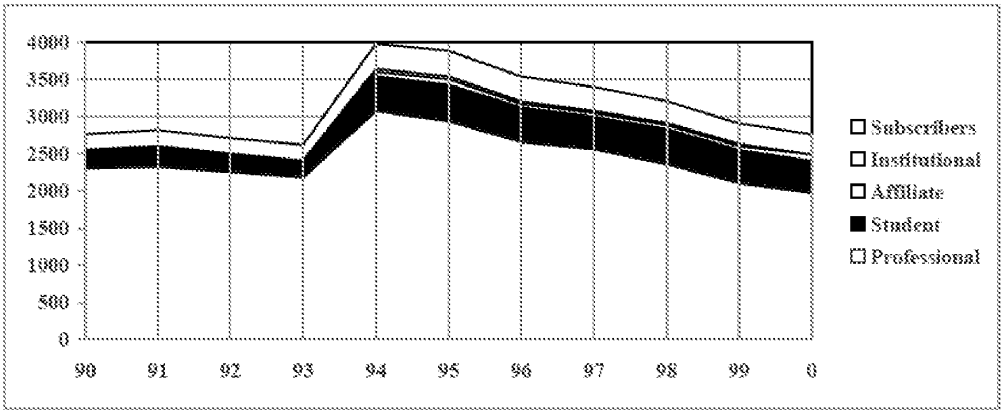
October 22, 2000

Dear Colleagues:

One of the perks of serving as a Sigcomm officer is to have the opportunity to talk with many of you. The Stockholm conference was a superb get together for renewing old friendships, making new ones, seeing an inspiring city, and renewing and refreshing myself technically from the posters, talks, tutorials, arguments, and overheard conversations. The level of the interaction was outrageous! (Please see http://www.acm.org/sigcomm/sigcomm2000/conf/outrageous.html) As Craig's letter has already told you, we will have opportunities to meet again in San Jose, Costa Rica, April 3-5, 2001 and at what is already looking to be the biggest SIGCOMM ever in San Diego, August 27-31, 2001. In the next few sections of this note, I will update you on the state of the SIGCOMM membership, finances, and the SIGCOMM and ACM digital libraries.

Membership:

The graph below shows some history of SIGCOMM membership. A particular feature to note is the large



spike upward in 1994. Anecdotes that explain this spike are very welcome. Please send mail to gdelp@acm.org. We expect that this is a result of a dues structure change, some additional society being rolled into SIGCOMM, or just the fine London venue for the conference, but clearer information would help us duplicate the result.

The great news of the retention information is that the vast majority of you are satisfied multi-year members. Help us get the word out. The numbers that we would like to change are the 789 who left SIGCOMM this year. If Your word of mouth is the most effective means for expanding our benefit to others in the community!

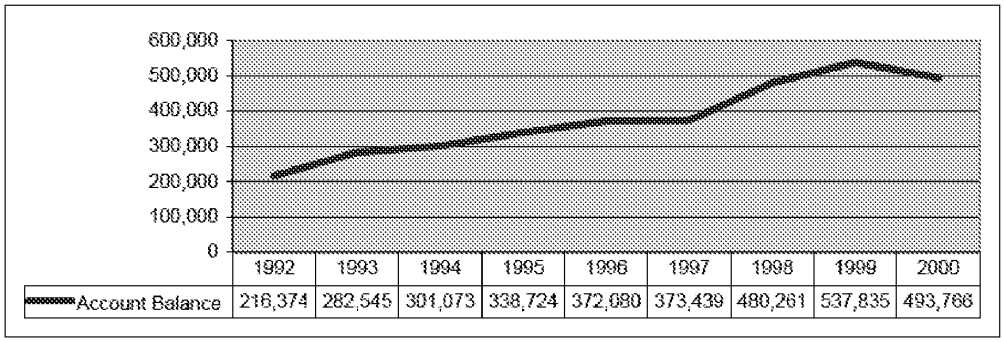
June 2000 retention:	1 year	2 years	>2 years	Total
Active (renewed)	294	762	1,152	2,208
Expiring	210	270	309	789
Total	504	1,032	1,461	2,997
Retention Rate	58%	74%	79%	74%

SIGCOMM Officers
Sigcommex@acm.org

Craig Partridge, *Chair*
Lixia Zhang, *Vice-Chair*
A. Terzis, *Information Services Director*
Gary Delp, *Secretary-Treasurer*
Roch Guerin, *Editor*
Jon Crowcroft, *Past Chair*
Chris Edmonson-Yurkanan, *Conference Advisor*

Finances:

This section is actually worth reading! Some of the stories behind the numbers show that the ACM organization has financially served SIGCOMM very well. SIGCOMM continues as a robust and healthy organization. As you can see from the `00 budget, last year we made a concerted effort to spend down the balance by $55,275. This would have spent our SIG account balance down to $482,560. However, we spent $35K less than budgeted to bring the past SIGCOMM Award winners to Boston for the 10 year celebration of the SIGCOMM Award: Boston was an excellent location. In additions, the SIGCOMM conference, rather than loosing $374.00 as it was budgeted, accrued 34K. 15K of this will be reflected in this year's ('01) budget. The gain was due to both the generosity of the sponsors (which allowed us to substantially add to the conference), and the vigilance of the ACM staff. Erica Johnson, our program director in particular, kept very close watch on the catering actually delivered in Boston, and was able to reduce the conference expenses by $85K. Thank you, Erica! We are very interested in spending additional monies to enhance the conference, but we do not want to waste any of the money we spend. The ACM staff does very well making sure that we get our money's worth from the vendors that we hire.



SIGCOMM has co-sponsored the Virtual Reality Modeling Language Symposium (VRML) for the past few years. The audience for this conference is no longer active, and SIGCOMM no longer a sponsor. To give credit to ACM process improvements and to welcome Ann Marie Ferrara to the ACM staff as coordinated venue negotiator, we provide some history. Previous years of VRML lead the organizers to believe that the attendance would be over 350. Before Ann Marie's arrival, the VRML conference organizers contracted with a Marriott hotel in Monterey, CA. Part of the contract (all conference contracts) was a guarantee for the rental of rooms and minimum catering numbers. In November, the SIGCOMM executive committee was told of potential low attendance; registration was less than 15% of the guaranteed number. We choose between canceling VRML and paying cancellation charges that would have been ~$80K for SIGCOMM's portion, or continuing with the conference with a maximum risk of $120K. ACM staff assisted in renegotiating the contracts, trimming the budget appropriate to the new conference size, and supporting the conference in a number of other ways. The great news is that the SIGCOMM loss for this conference was limited to $37K. Thanks to ACM. (It is worth noting that the profit for VRML over the previous years goes a long way to covering this final shortfall).

ACM fiscal years end on June 30 of each year, and include the conferences of the previous summer. Stockholm numbers are not yet in for the current year, but the Vancouver and Boston conferences are included in FY 99 and 2000 respectively. In the charts below, you can see the actuals of the previous years with the budget of the current year.

Income:	FYE 99 Actual	`00 Actual	`00 Variance	`00 Budget	FYE 01 Budget
SIG Dues	44,518	40,240	(1,015)	41,256	41,150
Non member subs	10,615	9,831	(1,268)	11,100	10,360
Other Subscription Income	16,522	21,488	10,168	11,320	4,750
Paper Sales	5,431	4,537	191	4,346	4,346
Conference revenue	315,057	358,871	64,716	294,155	196,600
Interest income	26,180	24,886	7,445	17,441	24,132
Mailing list – expedited credits	1,941	1,676	576	1,100	1,100

Total Income:	420,264	461,529	80,811	380,718	282,482

Expenses:	FYE 99 Actual	`00 Actual	`00 Variance	`00 Budget	FYE 01 Budget
Staff Travel	1,719	7,398	(7,398)	0	12,000
Volunteer Travel	7,141	10,184	35,816	46,000	
Temporary Help/Communications	110	66	(66)	0	3,400
Mailing & Handling	1,907	636	(636)	0	1,715
Production	31,112	28,168	12,832	41,000	42,650
Distribution	23,185	21,725	(1,701)	20,024	10,109
Storage	1,590	1,546	(1,546)	0	0
Other Pub Services	10,786	6,421	(5,571)	850	0
Grants, Projects, & Awards	37,090	19,424	(4,424)	15,000	28,000
Meetings & Spec Functions	0	23,322	(23,322)	0	0
Large Conf Exp	209,475	363,823	(71,854)	291,969	194,909
Allocations	43,201	51,241	(7,455)	43,786	34,862
Sigboard Allocation To Sigs	1,397	2,247	2,666	4,913	1,682
Service Chg To Conferences	(7,118)	(32,017)	4,468	(27,549)	(17,694)
*Total Expenses	362,540	505,597	(69,604)	435,993	311,633
Net Surplus/(Deficit)===>	57,724	(44,068)		(55,275)	(29,351)

Digital libraries:

We have a long tradition of making SIGCOMM information widely accessible. We were one of the first societies to put full newsletter and conference text on-line, and have been adamant about making sure that this access is free to all. ACM SIGCOMM provides archival web storage, in most cases in searchable form to the output of the SIG. We know that this adds to the utility of publishing in SIGCOMM CCR and affiliated conferences. IEEE/ACM Transactions on Networking is openly accessible as well. You, as members, are making it possible for your work to reach a much wider audience!

You can find this treasure trove of wisdom through a number of links, including:
http://www.acm.org/pubs/contents/proceedings/series/comm
http://www.acm.org/sigcomm/ccr/index.html

As a part of the ACM Digital Library project, in '00 and continuing in '01, SIGCOMM has funded the scanning and posting of the back issues of SIGCOMM's publications. All of the SIGCOMM sponsored conference proceedings are already on line, and is ACM/IEEE Transactions on Networking. The issues of our CCR newsletter that were not submitted digitally will be scanned and posted by the end of '01. Every effort is being made to accelerate this task. The SIGCOMM section of the ACM Digital Library is located at http://www.acm.org/dl/proc_bysig_list.html#SIGCOMM

Happy hunting!
Sincerely,

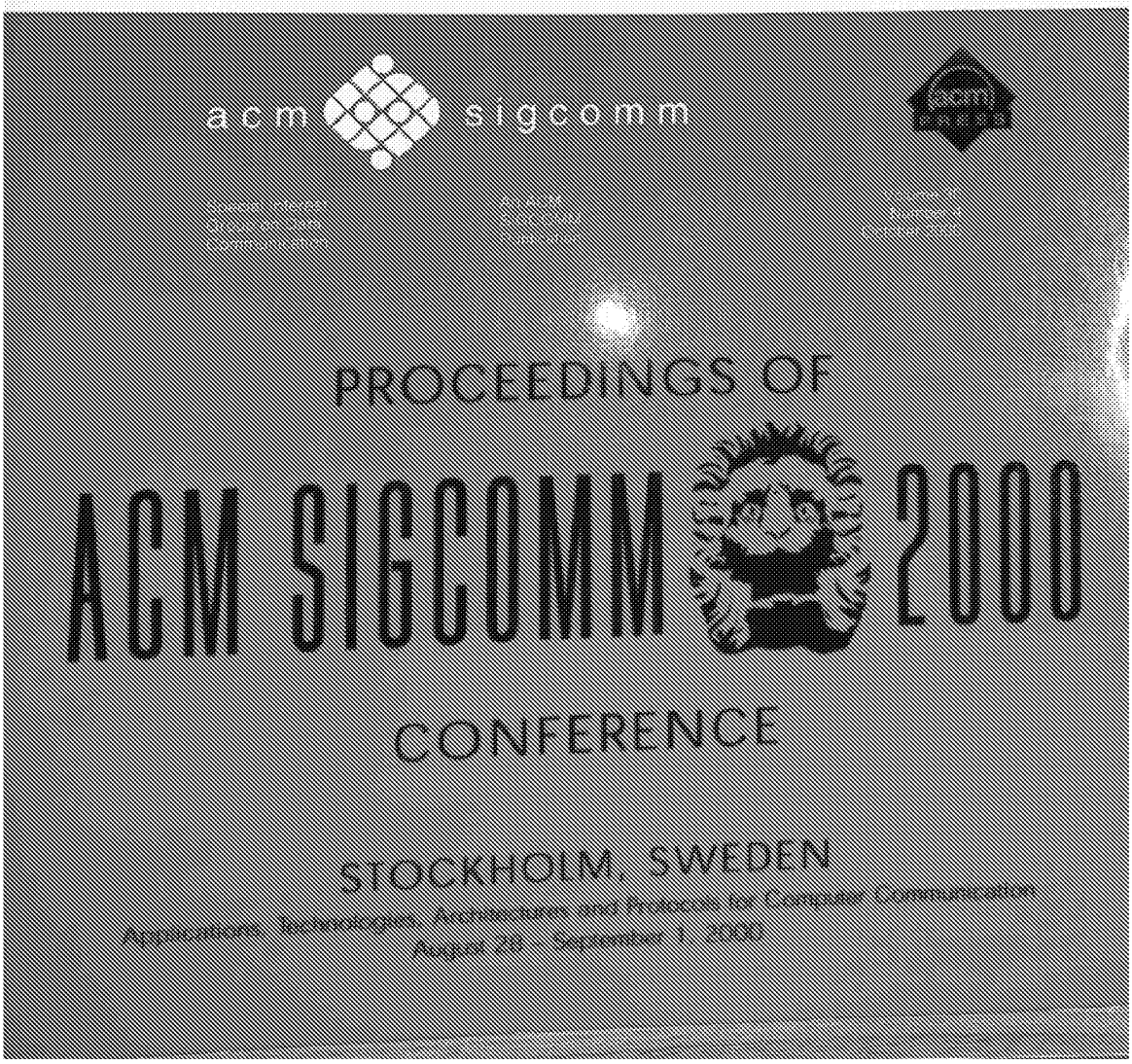


Gary S. Delp SIGCOMM Secretary – Treasurer

p.s. As a reminder, your elected officers serve (listed at the bottom of the first page) serve as volunteers. We are professionals working in the communications field just as you are. We rely on the professional support staff of the Association for Computing Machinery (ACM), directly upon our Program Director, Erica Johnson, but also on all the rest, to provide long term stability, financial management, printing and archival services, conference services, and many, many other things. I take this opportunity as a volunteer to salute our headquarters staff for long term and forward looking service to the SIG!

ATTACHMENT J
(EXHIBIT 1021)

Computer Communication Review



Cloudflare, Inc. Exhibit 1003 Page 1186

Computer Communication Review

a publication of the acm special interest group on data communication

Chairman:

Craig Partridge
BBN Technologies
10 Moulton St
Cambridge MA 02138
+1-510-558-8675 (work)
+1-510-558-8675 (fax)
craig@bbn.com

Editor:

Roch Guerin
University of Pennsylvania
Dept. Elec. Eng., Rm. 367 GRW
200 South 33rd Street
Philadelphia, PA 19104
+1-215-898-9351 (work)
+1-215-573-2068 (fax)
guerin@ee.upenn.edu

Vice Chairman:

Lixia Zhang
Computer Science Dept.
4531G Boelter Hall, UCLA
Los Angeles, CA 90095-1596
+1-310-825-2695 (work)
+1-310-825-2273 (fax)
lixia@cs.ucla.edu

Information Services Director:

Andreas Terzis
4677 Boelter Hall
Computer Science Dept, UCLA
Los Angeles, CA 90095
+1-310-267-2190 (work)
+1-310-825-2273 (fax)
terzis@cs.ucla.edu

Secretary-Treasurer:

Gary S. Delp
IBM Corporation, 030-3/A21 Dept 52J
3605 Highway 52 North
Rochester, MN 55901
+1-507-253-0088 (work)
+1-507-253-4451 (fax)
gdelp@us.ibm.com

SIG Program Director:

Erica Johnson
ACM
1515 Broadway, 17th Floor
New York, NY 10036
+1-212-626-0611 (work)
+1-212-302-5826 (fax)
johnson_e@acm.org

Area Editors:

Vaduvur Bharghavan
Coordinated Science Lab
University of Illinois
1308 W. Main Street
Urbana, IL 61801
bharghav@crhc.uiuc.edu

Ramesh Govindan
USC/ISI
4676 Admiralty Way
Suite 1001
Marina del Rey, CA 90292
govindan@isi.edu

Balachander Krishnamurthy
AT&T Labs – Research
D-229180
Park Avenue
Florham Park, NJ 07932
bala@research.att.com

Klara Nahrstedt
Dept. Computer Science
University of Illinois
1304 W. Springfield Av.
Urbana, IL 61801
klara@nahrstedt.cs.uiuc.edu

Executive Committee:

Craig Partridge
Lixia Zhang
Gary S. Delp
Jon Crowcroft
Chris Edmondson-Yurkanan
Andreas Terzis
Roch Guerin

Award Committee Chairman:

David C. Wood
NATO C3 Agency
P.O. Box 174
2501 CD The Hague, The Netherlands
+31-70-314-2420 (work)
+31-70-314-2176 (fax)
woodd@nc3a.nato.int

Conference Coordinator:

Chris Edmondson-Yurkanan
Computer Sciences, TAY 2.124 (C0500)
The University of Texas at Austin
Austin, TX 78712-1188
+1-512-471-9546 (Work)
Chris@cs.utexas.edu

SIGCOMM Mission Statement:

SIGCOMM is ACM's professional forum for the discussion of topics in the field of communications and computer networks, including technical design and engineering, regulation and operations, and the social implications of computer networking.

The SIG's members are particularly interested in the systems engineering and architectural questions of communication.

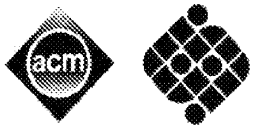
Feedback on our mission is always welcome. Please email suggestions to any of the members of the SIGCOMM Executive Committee.

Advertising: ACM accepts recruitment advertising under the basic premise that the advertising employer does not discriminate on the basis of age, color, race, religion, gender, sexual preference, or national origin. ACM recognizes, however, that laws on such matters vary from country to country and contain exceptions, inconsistencies or contradictions. This is as true of the laws of the United States of America as it is of other countries. Thus ACM policy requires each advertising employer to state explicitly in the advertisement any employment restrictions that may apply with respect to age, color, race, religion, gender, sexual preference, or national origin. Observance of the legal retirement age in the employer's country is not considered discrimination under this policy.
For advertising information, please contact Lynne Lancaster (Advertising Manager) at ACM, 1515 Broadway, New York, NY 10036 USA, tel: +1 212 869 7440, fax: +1 212 869 0481

COMPUTER COMMUNICATION REVIEW is a publication of the ACM Special Interest Group on Data Communication. Its scope of interest includes: data communication systems for computers; data communication technology for computers; reliability, security and integrity of data in data communication systems; problems of interfacing communication systems and computer systems; computer communication system modelling and analysis.

Items attributed to persons will ordinarily be interpreted as personal rather than organizational opinions. Technical papers appearing in Computer Communication Review are informally reviewed.

COMPUTER COMMUNICATION REVIEW (ISSN 0146-4833) is published five times a year (January, April, July, and two issues in October) by the ACM, Inc., 1515 Broadway, New York, NY 10036. Annual subscription cost of $14.44 is included in the member dues of $50.00 (for students, the cost is included in $24.00); the non-member annual subscription is $37.00.
Periodicals postage paid at New York, NY 10001, and at additional mailing offices.
POSTMASTER: Send address changes to ACM COMPUTER COMMUNICATION REVIEW, 1515 Broadway, New York, NY 10036.



PROCEEDINGS OF

ACM SIGCOMM 2000



CONFERENCE

Applications, Technologies, Architectures and Protocols for Computer Communication

http://www.acm.org/sigcomm/

STOCKHOLM, SWEDEN

August 28 – September 1, 2000







Cloudflare, Inc. Exhibit 1003 Page 1188

The Association for Computing Machinery
1515 Broadway
New York, New York 10036

Copyright 2000 by the Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of portions of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyright for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permission to republish from: Publications Dept. ACM, Inc. Fax +1 (212) 869-0481 or <permissions@acm.org>.

For other copying of articles that carry a code at the bottom of the first or last page, copying is permitted provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

Notice to Past Authors of ACM-Published Articles

ACM intends to create a complete electronic archive of all articles and/or other material previously published by ACM. If you have written a work that has been previously published by ACM in any journal or conference proceedings prior to 1978, or any SIG Newsletter at any time, and you do NOT want this work to appear in the ACM Digital Library, please inform permissions@acm.org, stating the title of the work, the author(s), and where and when published.

ACM ISBN: 1-58113-224-7

Additional copies may be ordered prepaid from:

ACM Order Department
PO Box 11405
New York, NY 10286-1405

Phone: 1-800-342-6626
(US and Canada)
+1-212-626-0500
(all other countries)
Fax: +1-212-944-1318
E-mail: acmhelp@acm.org

ACM Order Number 533000
Printed in the USA

Cloudflare, Inc. Exhibit 1003 Page 1189

Conference Chairmen's Message

Welcome to Stockholm, Sweden and the ACM SIGCOMM 2000 Conference! We have an outstanding technical program and tutorials, representing the best in the whole field of Computer Communication.

Stockholm, often called "Beauty on Water" welcomes you with a reception in the City Hall where the Nobel prize dinner is held. A sightseeing cruise to the City Hall will provide you with a taste of the Stockholm water ways of today. The conference dinner in the Vasa museum, by the historic warship Vasa that sank within Stockholm harbor in 1628, will give you a glimpse of the water ways of the past. Stockholm is today vibrant with new small and large Internet-related companies and is known as The City of Wireless Internet.

The lion logo for this year's Conference is picked from a detail of the Vasa ship - an impressively decorated cannon port. Such a port should impose strength and scare the enemy. By that time, the lion was used in Sweden and elsewhere to represent strength. We thank Michel Östlund from Hera for the logo design and for associating SIGCOMM with strength.

This year's Conference has a new event: a poster session on work-in-progress that will take place during an extended break on Thursday. New for the participants this year is a temporary Radio LAN coverage of the conference rooms, which will enable ubiquitous Internet access. The successful Outrageous Opinions session will appear again, this time guided by Gary Delp. He says: "Nothing is so permanent as a temporary addition. The Outrageous Opinions session is a true prognostigatory enrichment to the SIGCOMM Conference".

A great deal of effort, by many individual volunteers, has gone into the technical, tutorial and social program to ensure that this year's conference consistently maintained the high SIGCOMM standard. We would like to thank the many individuals that made it possible. First of all, the Technical program committee, led by program chairs Christophe Diot and Jim Kurose, mastered a new record number of submitted paper with an unmatched thorough review process, and a careful selection of the best papers in the field. Our Local Arrangement chairs Bengt Ahlgren and Christian Tschudin prepared the conference venue with Internet access and handled the forever changing conference web. Joe Touch did the publication announcements and Andreas Terzis maintained our ACM web site. A special thanks goes to Stockholm Convention Bureau for their professional handling of the conference logistics.

A very special thanks goes to Chris Edmondson-Yurkanan. She is a true goldmine of past experiences of arranging SIGCOMM conferences. Our SIG Program director Erica Johnson made sure that all decisions were running smoothly through the ACM headquarters.

ACM SIGCOMM is encouraging student participation by providing a Student Travel Grant program and by a reduced registration fee for students. For this year we have received generous support for this program by Nokia Corporation and the US National Science Foundation. We are thankful to the selection committee and its chair Brian Levine for the fund raising and the selection of students for the grants.

Finally, we would like to express our appreciation for our main supporters, Ericsson, Sprint and Telia. Their generous donations have been a significant help in organizing the Conference. Besides these main supporters we have received technical support for both wireless and fixed access to the Internet from Lucent Technologies, Telia, NetInsight and Marratech.

We hope that you will have a pleasant time in Stockholm!

Per Gunningberg and **Steve Pink**
General Chairs
ACM SIGCOMM 2000 Conference

Cloudflare, Inc. Exhibit 1003 Page 1190

A Message from the Program Committee Chairs

The process of putting together the SIGCOMM technical program is always a difficult one. With a large number of submissions, and SIGCOMM's tradition as a single-track conference, only a small fraction of the submitted papers can be accepted. This year we received 238 submissions (a record!) and have a full program of 26 high quality papers. We hope you'll find the program to be full of exciting and innovative work.

Manuscripts submitted to SIGCOMM2000 underwent a rigorous review process. In the first phase of the review process, each paper was reviewed by three Program Committee (PC) members and was discussed electronically. In the second phase, PC members and outside referees provided an additional 150 or so reviews of highly-rated papers or papers with reviewer disagreements in the first round. In all, well over 800 reviews were received; all papers received at least three reviews and many received four or more. The culmination of the review process was the PC meeting - a lively, day-long discussion of the submitted papers and reviews. This was attended by nearly the entire PC. Throughout the review process and the PC meeting, all reviews and discussion were double-blind -- the authors did not know the identity of the referees, and the referees did not know the identities of the authors.

The result of this hard work is a program that addresses both fundamental issues in Internet networking (TCP analysis, control mechanisms, performance analysis and modeling, algorithms), but that also includes hot topics such as multicast, web performance, tracing and measurement. With a large number of submitted manuscripts, many strong submissions could not be accepted. SIGCOMM's traditional single-track format is aimed at providing ample presentation and discussion time among all conference attendees of many (but certainly not all) of the top networking research papers from the year. In order to provide a vehicle for showcasing additional recent work, we have introduced a work-in-progress poster session this year, targeted especially at students.

We gratefully acknowledge the help of the many people whose hard work has resulted in this program. Our first thanks go to the authors, who gave us the privilege of considering their manuscripts; without them and their innovative work, there would be no conference. For many years, SIGCOMM has been known for the generally high quality, depth and carefulness of its reviews. This year is no exception. Program Committee members volunteer huge amounts of time in providing reviews and helping to shape the program; few conferences require more hard work or dedication by PC members. The entire technical community owes them a great debt of thanks. Experts from outside the PC also volunteered time in providing reviews; they are acknowledged on page x. Finally, we thank the UMass student crew (Jim Padhye, Michael Bradshaw, Dan Rubenstein, and Jonathan Shapiro) who helped with the electronic submission, printing and management of a large number of submissions.

We hope you will enjoy the program as much as we enjoyed our (sometimes full time) job as PC co-chairs. It's an excellent program, and we can't imagine it being presented in a more beautiful city than Stockholm.

Christophe Diot and **Jim Kurose**
Program Committee Chairs
ACM SIGCOMM 2000 Conference

Cloudflare, Inc. Exhibit 1003 Page 1191

ACM SIGCOMM 2000

TABLE OF CONTENTS

Organization Committee ix

Program Committee ix

Reviewers x

Keynote Speech xi
André Danthine (University of Liège, Belgium)

Session 1: Multicast and Anycast

- **A Framework for Scalable Global IP-Anycast (GIA)** 3
  Dina Katabi, John Wroclawski (Massachusetts Institute of Technology)
- **pgmcc: a TCP-friendly single-rate multicast congestion control scheme** 17
  Luigi Rizzo (Universita di Pisa)
- **Fault Isolation in Multicast Trees** 29
  Anoop Reddy, Ramesh Govindan, Deborah Estrin (USC/Information Sciences Institute)

Session 2: Control Mechanisms

- **Equation-Based Congestion Control for Unicast Applications** 43
  Sally Floyd, Mark Handley (AT&T Center for Internet Research at ICSI), *Jitendra Padhye* (University of Massachusetts at Amherst), *Jörg Widmer* (International Computer Science Institute)
- **Endpoint Admission Control: Architectural Issues and Performance** 57
  Lee Breslau (AT&T Labs), *Edward W. Knightly* (Rice University), *Scott Shenker* (International Computer Science Institute), *Ion Stoica, Hui Zhang* (Carnegie Mellon University)
- **Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services** 71
  Zhi-Li Zhang, Zhenhai Duan (University of Minnesota), *Lixin Gao* (Smith College), *Yiwei Thomas Hou* (Fujitsu Labs)

Session 3: World Wide Web

- **A Protocol-Independent Technique for Eliminating Redundant Network Traffic** 87
  Neil T. Spring, David Wetherall (University of Washington)
- **On Network-Aware Clustering of Web Clients** 97
  Balachander Krishnamurthy (AT&T Labs), *Jia Wang* (Cornell University)
- **The Content and Access Dynamics of a Busy Web Site: Findings and Implications** 111
  Venkata N. Padmanabhan (Microsoft Research), *Lili Qiu* (Cornell University)

Cloudflare, Inc. Exhibit 1003 Page 1192

Session 4: Performance Analysis and Modeling

- **Critical Path Analysis of TCP Transactions** 1
  Paul Barford, Mark Crovella (Boston University)
- **Tuning RED for Web Traffic** 13
  Mikkel Christiansen, Kevin Jeffay, David Ott, F. Donelson Smith (University of North Carolina at Chapel Hill)
- **A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED** 151
  Vishal Misra, Wei-Bo Gong, Don Towsley (University of Massachusetts at Amherst)

Session 5: Routing Stability and Convergence

- **Routing Stability in Congested Networks: Experimentation and Analysis** 163
  Aman Shaik, Anujan Varma (University of California at Santa Cruz), *Lampros Kalampoukas, Rohit Dube* (Bell Labs)
- **Delayed Internet Routing Convergence** 175
  Craig Labovitz (Microsoft Research), *Abha Ahuja, Abhijit Bose, Farnam Jahanian* (University of Michigan)

Session 6: Routing and Bridging

- **FIRE: Flexible Intra-AS Routing Environment** 191
  Craig Partridge (BBN Technologies), *Alex C. Snoeren* (BBN Technologies & Massachusetts Institute of Technology), *Tim Strayer, Beverly Schwartz, Matthew Condell, Isidro Castineyra* (BBN Technologies)
- **SmartBridge: A Scalable Bridge Architecture** 205
  Thomas L. Rodeheffer, Chandramohan A. Thekkath (Compaq Computer Corporation), *Darrell Anderson* (Duke University)

Session 7: TCP Analysis

- **TCP is Max-Plus Linear** 219
  François Baccelli, Dohy Hong (INRIA-ENS)
- **A Stochastic Model of TCP/IP with Stationary Random Losses** 231
  Eitan Altman, Konstantin Avrachenkov, Chadi Barakat (INRIA)
- **On the Propagation of Long-Range Dependence in the Internet** 243
  A. Veres (Ericsson Research), *Zs. Kenesi, S. Molnár* (Budapest University of Technology and Economics), *G. Vattay* (Eötvös University)

Session 8: Tracing and Measurement

- **Deriving Traffic Demands for Operational IP Networks: Methodology and Experience** 257
  Anja Feldmann (University of Saarbruecken), *Albert Greenberg, Carsten Lund, Nick Reingold, Jennifer Rexford, Fred True* (AT&T Labs)
- **Trajectory Sampling for Traffic Measurement** 271
  N. G. Duffield, M. Grossglauser (AT&T Labs)
- **Measuring Link Bandwidths Using a Deterministic Model of Packet Delay** 283
  Kevin Lai, Mary Baker (Stanford University)
- **Practical Network Support For IP Traceback** 295
  Stefan Savage, David Wetherall, Anna Karlin, Tom Anderson (University of Washington)

Cloudflare, Inc. Exhibit 1003 Page 1193

Session 9: Header Processing

- **When the CRC and TCP Checksum Disagree** 309
  Jonathan Stone (Stanford Distributed Systems Group), *Craig Partridge* (BBN Technologies)
- **Packet Types: Abstract Specification of Network Protocol Messages** 321
  Peter J. McCann, Satish Chandra (Bell Labs)
- **Memory-Efficient State Lookups with Fast Updates** 335
  Sandeep Sikka (Washington University), *George Varghese* (UCSD)

Author Index 348

Cloudflare, Inc. Exhibit 1003 Page 1194

Practical Network Support for IP Traceback

Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson

Department of Computer Science and Engineering
University of Washington
Seattle, WA, USA

Abstract

This paper describes a technique for tracing anonymous packet flooding attacks in the Internet back towards their source. This work is motivated by the increased frequency and sophistication of denial-of-service attacks and by the difficulty in tracing packets with incorrect, or "spoofed", source addresses. In this paper we describe a general purpose traceback mechanism based on probabilistic packet marking in the network. Our approach allows a victim to identify the network path(s) traversed by attack traffic without requiring interactive operational support from Internet Service Providers (ISPs). Moreover, this traceback can be performed "post-mortem" - after an attack has completed. We present an implementation of this technology that is incrementally deployable, (mostly) backwards compatible and can be efficiently implemented using conventional technology.

1. INTRODUCTION

Denial-of-service attacks consume the resources of a remote host or network, thereby denying or degrading service to legitimate users. Such attacks are among the hardest security problems to address because they are simple to implement, difficult to prevent, and very difficult to trace. In the last several years, Internet denial-of-service attacks have increased in frequency, severity and sophistication. Howard reports that between the years of 1989 and 1995, the number of such attacks reported to the Computer Emergency Response Team (CERT) increased by 80 percent per year [23]. More recently, a 1999 CSI/FBI survey reports that 32 percent of respondents detected denial-of-service attacks directed against their sites [18]. Even more worrying, recent reports indicate that attackers have developed tools to coordinate distributed attacks from many separate sites [14].

Unfortunately, mechanisms for dealing with denial-of-service have not advanced at the same pace. Most work in this area has focused on *tolerating* attacks by mitigating their effects on the victim [38, 2, 26, 29, 9]. This approach can provide an effective stop-gap measure, but does not eliminate the problem nor does it discourage attackers. The other option, and the focus of this paper, is to trace attacks back towards their origin - ideally stopping an attacker at the source.

A perfect solution to this problem is complicated by the potential use of indirection to "launder" the true causal origin of an attack. For example, an attack may consist of packets sent from many different slave machines, themselves under the control of a remote master machine. Such indirection may be achieved either explicitly (by compromising the individual slave hosts directly) or implicitly (by sending false requests to the slaves on behalf of the victim - a so-called *reflector*). More challenging still, the true origin and identity of the attacker can be similarly concealed through chains of false computer accounts, call forwarding, and so forth. Consequently, we regard a complete solution - particularly one able to address the forensic needs of law enforcement - as an open problem.

Instead, we address the more limited operational goal of simply identifying the machines that *directly* generate attack traffic and the network path this traffic subsequently follows. We call this the *traceback problem* and it is motivated by the operational need to control and contain attacks. In this setting, even incomplete or approximate information is valuable because the efficacy of measures such as packet filtering improve as they are applied further from the victim and closer to the source.

However, even for our restricted problem, determining the source generating attack traffic is surprisingly difficult due to the stateless nature of Internet routing. Attackers routinely disguise their location using incorrect, or "spoofed", IP source addresses. As these packets traverse the Internet their true origin is lost and a victim is left with little useful information. While there are several ad hoc traceback techniques in use, they all have significant drawbacks that limit their practical utility in the current Internet.

In this paper we present a new approach to the traceback problem that addresses the needs of both victims and network operators. Our solution is to probabilistically mark packets with partial path information as they arrive at routers. This approach exploits the observation that attacks generally comprise large numbers of packets. While each marked packet represents only a "sample" of the path it has traversed, by combining a modest number of such packets a victim can reconstruct the entire path. This allows victims to locate the approximate source of attack traffic without requiring the assistance of outside network operators. Moreover, this determination can be made even after an attack has completed. Both facets of our solution represent substantial improvements over existing capabilities for dealing with flooding-style denial-of-service attacks.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGCOMM'00, Stockholm, Sweden.
Copyright 2000 ACM 1-58113-224-7/00/0008...$5.00.

Cloudflare, Inc. Exhibit 1003 Page 1195

A key practical deployment issue with any modification of Internet routers is to ensure that the mechanisms are efficiently implementable, may be incrementally deployed, and are backwards compatible with the existing infrastructure. We describe a traceback algorithm that adds little or no overhead to the router's critical forwarding path and may be incrementally deployed to allow traceback within the subset of routers supporting our scheme. Further, we demonstrate that we can encode the necessary path information in a way that peacefully co-exists with existing routers, host systems and more than 99% of today's traffic.

The rest of this paper is organized as follows: In Section 2, we describe related work concerning IP spoofing and solutions to the traceback problem. Section 3 outlines our basic approach and section 4 characterizes several abstract algorithms for implementing it. In Section 5 we detail a concrete encoding strategy for our algorithm that can be implemented within the current Internet environment. We also present experimental results demonstrating the effectiveness of our solution. In section 6 we discuss the main limitations and weaknesses of our proposal and potential extensions to address some of them. Finally, we summarize our findings in Section 7.

2. RELATED WORK

It has been long understood that the IP protocol permits anonymous attacks. In his 1985 paper on TCP/IP weaknesses, Morris writes:

> "The weakness in this scheme [the Internet Protocol] is that the source host itself fills in the IP source host id, and there is no provision in ... TCP/IP to discover the true origin of a packet." [31]

In addition to denial-of-service attacks, IP spoofing can be used in conjunction with other vulnerabilities to implement anonymous one-way TCP channels and covert port scanning [31, 3, 24, 44].

There have been several efforts to reduce the anonymity afforded by IP spoofing. Table 1 provides a subjective characterization of each of these approaches in terms of management cost, additional network load, overhead on the router, the ability to trace multiple simultaneous attacks, the ability trace attacks after they have completed, and whether they are preventative or reactive. We also characterize our proposed traceback scheme according to the same criteria. In the remainder of this section we describe each previous approach in more detail.

2.1 Ingress filtering

One way to address the problem of anonymous attacks is to eliminate the ability to forge source addresses. One such approach, frequently called *ingress filtering*, is to configure routers to block packets that arrive with illegitimate source addresses [20]. This requires a router with sufficient power to examine the source address of every packet and sufficient knowledge to distinguish between legitimate and illegitimate addresses. Consequently, ingress filtering is most feasible in customer networks or at the border of Internet Service Providers (ISP) where address ownership is relatively unambiguous and traffic load is low. As traffic is aggregated from multiple ISPs into transit networks, there is no longer enough information to unambiguously determine if a packet arriving on a particular interface has a "legal" source address. Moreover, on many deployed router architectures the overhead of ingress filter becomes prohibitive on high-speed links.

The principal problem with ingress filtering is that its effectiveness depends on widespread, if not universal, deployment. Unfortunately, a significant fraction of ISPs, perhaps the majority, do not implement this service – either because they are uninformed or have been discouraged by the administrative burden, potential router overhead and complications with existing services that depend on source address spoofing (e.g. some versions of Mobile IP [33] and some hybrid satellite communications architectures). A secondary problem is that even if ingress filtering were universally deployed at the customer-to-ISP level, attackers could still forge addresses from the hundreds or thousands of hosts within a valid customer network [14].

It is clear that wider use of ingress filtering would dramatically improve the Internet's robustness to denial-of-service attacks. At the same time it is prudent to assume that such a system will never be fullproof – and therefore traceback technologies will continue to be important.

2.2 Link testing

Most existing traceback techniques start from the router closest to the victim and interactively test its upstream links until they determine which one is used to carry the attacker's traffic. Ideally, this procedure is repeated recursively on the upstream router until the source is reached. This technique assumes that an attack remains active until the completion of a trace and is therefore inappropriate for attacks that are detected after the fact, attacks that occur intermittently, or attacks that modulate their behavior in response to a traceback (it is prudent to assume the attacker is fully informed). Below we describe two varieties of link testing schemes, *input debugging* and *controlled flooding*.

2.2.1 *Input debugging*

Many routers include a feature called *input debugging*, that allows an operator to filter particular packets on some egress port and determine which ingress port they arrived on. This capability is used to implement a trace as follows: First, the victim must recognize that it is being attacked and develop an *attack signature* that describes a common feature contained in all the attack packets. The victim communicates this signature to a network operator, frequently via telephone, who then installs a corresponding input debugging filter on the victim's upstream egress port. This filter reveals the associated input port, and hence which upstream router originated the traffic. The process is then repeated recursively on the upstream router, until the originating site is reached or the trace leaves the ISP's border (and hence its administrative control over the routers). In the later case, the upstream ISP must be contacted and the procedure repeats itself. While such tracing is frequently performed manually, several ISPs have developed tools to automatically trace attacks across their own networks [41].

The most obvious problem with the input debugging approach, even with automated tools, is its considerable management overhead. Communicating and coordinating with network operators in multiple ISPs requires the time, attention and commitment of both the victim and the remote personnel – many of whom have no direct economic incentive to provide aid. If the appropriate network

[1]Some modern routers ease the administrative burden of ingress filtering by providing functionality to automatically check source addresses against the destination-based routing tables (e.g. the `verify unicast reverse-path` on Cisco's IOS). This approach is only valid if the route to and from the customer has a symmetric metric – generally at the border of single-homed stub networks.

Cloudflare, Inc. Exhibit 1003 Page 1196

	Management overhead	Network overhead	Router overhead	Distributed capability	Post-mortem capability	Preventative/ reactive
Ingress filtering	Moderate	Low	Moderate	N/A	N/A	Preventative
Link testing	High	Low	High	Good	Poor	Reactive
Input debugging	Low	High	Low	Poor	Poor	Reactive
Controlled flooding	High	Low	High	Excellent	Excellent	Reactive
Logging	Low	Low	Low	Good	Excellent	Reactive
ICMP Traceback	Low	Low	Low	Good	Excellent	Reactive
Marking						

...: Qualitative comparison of existing schemes for combating anonymous attacks and the probabilistic marking approach we ...

...ers are not available, if they are unwilling to assist, or if they ...k have the appropriate technical skills and capabilities, then a ...back may be slow or impossible to complete [21].

...2 *Controlled flooding*

...ch and Cheswick have developed a link testing traceback tech- ...que that does not require any support from network operators [6]. ...y call this technique *controlled flooding* because it tests links by ...ding them with large bursts of traffic and observing how this ...orts traffic from the attacker. Using a pre-generated "map" ...Internet topology, the victim coerces selected hosts along the ...stream route into iteratively flooding each incoming link on the ...ter closest to the victim. Since router buffers are shared, packets ...oding across the loaded link - including any sent by the attacker ...ve an increased probability of being dropped. By observing ...anges in the rate of packets received from the attacker, the victim ...n therefore infer which link they arrived from. As with other link ...sting schemes, the basic procedure is then applied recursively on ...e next upstream router until the source is reached.

While the scheme is both ingenious and pragmatic, it has several ...rawbacks and limitations. Most problematic among these is that ...ontrolled flooding is itself a denial-of-service attack – exploiting ...ulnerabilities in unsuspecting hosts to achieve its ends. This draw- ...ack alone makes it unsuitable for routine use. Also, controlled ...ooding requires the victim to have a good topological map of large ...ections of the Internet in addition to an associated list of "willing" ...ooding hosts. As Burch and Cheswick note, controlled flooding ...s also poorly suited for tracing distributed denial-of-service attacks ...ecause the link-testing mechanism is inherently noisy and it can ...e difficult to discern the set of paths being exploited when mul- ...iple upstream links are contributing to the attack. Finally, like all ...ink-testing schemes, controlled flooding is only effective at tracing ...n on-going attack and cannot be used "post-mortem".

...3 Logging

...n approach suggested in [37] and [41] is to log packets at key ...outers and then use data mining techniques to determine the path ...hat the packets traversed. This scheme has the useful property that ...t can trace an attack long after the attack has completed. However, ...t also has obvious drawbacks, including potentially enormous re- ...ource requirements (possibly addressed by sampling) and a large ...cale inter-provider database integration problem. We are unaware ...f any commercial organizations using a fully operational trace- ...back approach based on logging[2].

[2]Historically, the T3-NFSNET *did* log network-to-network traffic statistics and these were used on at least one occasion to trace IP spoofing attacks to an upstream provider [43].

2.4 ICMP Traceback

Since the first writing of this paper, a new traceback proposal bas emerged based on the use of explicit router-generated ICMP trace- back messages [4]. The principle idea in this scheme is for every router to sample, with low probability (e.g., 1/20,000), one of the packets it is forwarding and copy the contents into a special ICMP traceback message including information about the adjacent routers along the path to the destination. During a flooding-style attack, the victim host can then use these messages to reconstruct a path back to the attacker. This scheme has many benefits compared to previous work and is in many ways similar to the packet marking approach we have taken. However, there are several disadvantages in the current design that complicate its use. Among these: ICMP traffic is increasingly differentiated and may be filtered or rate lim- ited differently from normal traffic, the ICMP Traceback message relies on an input debugging capability (i.e. the ability to asso- ciate a packet with the input port and/or MAC address on which it arrived) that is not available in some router architectures, if only some of the routers participate it seems difficult to positively "con- nect" traceback messages from participating routers separated by a non-participating router, and finally, it requires a key distribution infrastructure to deal with the problem of attackers sending false ICMP Traceback messages. That said, we believe that the scheme is promising and that hybrid approaches combining it with some of the algorithms we propose are likely to be quite effective.

3. OVERVIEW

Burch and Cheswick mention the possibility of tracing flooding at- tacks by "marking" packets, either probabilistically or deterministi- cally, with the addresses of the routers they traverse [6]. The victim uses the information in the marked packets to trace an attack back to its source. This approach has not been previously explored in any depth, but has many potential advantages. It does not require inter- active cooperation with ISPs and therefore avoids the high manage- ment overhead of input debugging. Unlike controlled flooding, it does not require significant additional network traffic and can po- tentially be used to track multiple attacks. Moreover, like logging, packet marking can be used to trace attacks "post-mortem" - long after the attack has stopped. Finally, we have found that marking algorithms can be implemented without incurring any significant overhead on network routers. The remainder of this paper focuses on fully exploring and characterizing this approach.

3.1 Definitions

Figure 1 depicts the network as seen from a victim V. For the purposes of this paper, V may be a single host under attack, or a network border device such as a firewall or intrusion detection sys- tem that represents many such hosts. Every potential *attack origin* A_i is a leaf in a tree rooted at V and every router R_i is an internal

Cloudflare, Inc. Exhibit 1003 Page 1197

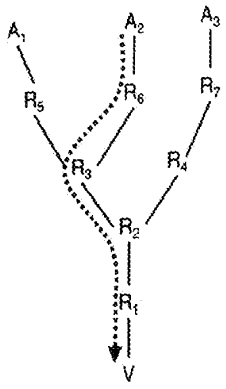


Figure 1: Network as seen from the victim of an attack, V. Routers are represented by R_i, and potential attackers by A_i. The dotted line represents a particular *attack path* between an attacker and the victim.

node along a path between some A_i and V. The *attack path* from A_i is the unique ordered list of routers between A_i and V. For instance, if an attack originates from A_2 then to reach V it must first traverse the path R_6, R_3, R_2, and R_1 - as shown by the dotted line in Figure 1.

The *exact traceback* problem is to determine the attack path and the associated attack origin for each attacker. However, solving this problem is complicated by several practical limitations. The exact attack origin may never be revealed (even MAC source addresses may be spoofed) and a wily attacker may send false signals to "invent" additional routers in the traceback path. We address these issues in section 6, but for now we restrict our discussion to solving a more limited problem. We define the *approximate traceback* problem as finding a candidate attack path for each attacker that contains the true attack path as a suffix. We call this the *valid suffix* of the candidate path. For example, $(R_5, R_6, R_3, R_2, R_1)$ is a valid approximate solution to Figure 1 because it contains the true attack path as a suffix. We say a solution to this problem is *robust* if an attacker cannot prevent the victim from discovering candidate paths containing the valid suffix.

All marking algorithms have two components: a *marking procedure* executed by routers in the network and a *path reconstruction procedure* implemented by the victim. A router "marks" one or more packets by augmenting them with additional information about the path they are traveling. The victim attempts to reconstruct the attack path using only the information in these marked packets. The *convergence time* of an algorithm is the number of packets that the victim must observe to reconstruct the attack path.

3.2 Basic assumptions

The design space of possible marking algorithms is large, and to place our work in context we identify the assumptions that motivate and constrain our design:

- an attacker may generate any packet,
- multiple attackers may conspire,
- attackers may be aware they are being traced,
- packets may be lost or reordered,
- attackers send numerous packets,
- the route between attacker and victim is fairly stable,
- routers are both CPU and memory limited, and
- routers are not widely compromised.

The first four assumptions represent conservative assessments of the abilities of the modern attackers and limitations of the network. Designing a traceback system for the Internet environment is extremely challenging because there is very little that can be trusted. In particular, the attacker's ability to create arbitrary packets significantly constrains potential solutions. When a router receives a packet, it has no way to tell whether that packet has been marked by an upstream router or if the attacker simply has forged this information. In fact, the only invariant that we can depend on is that a packet from the attacker must traverse all of the routers between it and the victim.

The remaining assumptions reflect the basis for our design and deserve additional discussion. First, denial-of-service attacks are only effective so long as they occupy the resources of the victim. Consequently, most attacks are comprised of thousands or millions of packets. Our approach relies on this property because we mark each packet with only a small piece of path state and the victim must observe many such packets to reconstruct the complete path back the the attacker. If many attacks emerge that require only a single packet to disable a host (e.g. ping-of-death [11]), then this assumption may not hold (although we note that even these attacks require multiple packets to *keep* a machine down).

Second, measurement evidence suggests that while Internet routes do change, it is extremely rare for packets to follow many different paths over the short time-scales of a traceback operation (seconds in our system) [32]. This assumption greatly simplifies the role of the victim, since it can therefore limit its consideration to a single primary path for each attacker. If the Internet evolves to allow significant degrees of multi-path routing then this assumption may not hold.

Third, while there have been considerable improvements in router implementation technology, link speeds have also increased dramatically. Consequently, we assert that any viable implementation must have low per-packet overhead and must not require per-flow state. Significantly simpler schemes than ours can be implemented if we assume that routers are not resource constrained.

Finally, since a compromised router can effectively eliminate any information provided by upstream routers, it is effectively indistinguishable from an attacker. In such circumstances, the security violation at the router must be addressed first, before any further traceback is attempted. In normal circumstances, we believe this is an acceptable design point. However, if non-malicious, but information hiding, routing infrastructures become popular, such as described in [22, 35], then this issue may need to be revisited.

4. BASIC MARKING ALGORITHMS

In this section we describe a series of marking algorithms - starting from the most simple and advancing in complexity. Each algorithm attempts to solve the approximate traceback problem in a manner consistent with our assumptions.

Cloudflare, Inc. Exhibit 1003 Page 1198

... procedure at router R:
... each packet w, append R to w
... construction procedure at victim v:
... every packet w from attacker
... extract path $(R_i..R_j)$ from the suffix of w

Figure 2: Node append algorithm.

Node append

...roblest marking algorithm – conceptually similar to the IP ...ord Route option [34] – is to append each node's address to the ...e packet as it travels through the network from attacker to ...(see Figure 2). Consequently, every packet received by the ...arrives with a complete ordered list of the routers it traversed ...th an attack path.

...node append algorithm is both robust and extremely quick to ...verge (a single packet), however it has several serious limita... Principal among these is the infeasibly high router overhead ...ured by appending data to packets in flight. Moreover, since ...length of the path is not known *a priori*, it is impossible to ...sure that there is sufficient unused space in the packet for the ...mplete list. This can lead to unnecessary fragmentation and bad ...eractions with services such as MTU discovery [30]. This prob... cannot be solved by reserving "enough" space, as the attacker ...n completely fill any such space with false, or misleading, path ...formation.

4.2 Node sampling

To reduce both the router overhead and the per-packet space requirement, we can sample the path one node at a time instead of recording the entire path. A single static "node" field is reserved in the packet header – large enough to hold a single router address (i.e. 32 bits for IPv4). Upon receiving a packet, each router chooses to write its address in the node field with some probability p. After enough packets have been sent, the victim will have received at least one sample for every router in the attack path. As stated in section 3, we assume that the attacker sends enough packets and the route is stable enough that this sampling can converge.

Although it might seem impossible to reconstruct an ordered path given only an unordered collection of node samples, it turns out that with a sufficient number of trials, the order can be deduced from the relative number of samples per node. Since routers are arranged serially, the probability that a packet will be marked by a router and then left unmolested by all downstream routers is a strictly decreasing function of the distance to the victim. If we constrain p to be identical at each router, then the probability of receiving a marked packet from a router d hops away is $p(1-p)^{d-1}$. Since this function is monotonic in the distance from the victim, ranking each router by the number of samples it contributes will tend to produce the accurate attack path. The full algorithm is shown in Figure 3.

Putting aside for the moment the difficulty in changing the IP header to add a 32-bit node field, this algorithm is efficient to implement because it only requires the addition of a write and checksum update to the forwarding path. Current high-speed routers already must perform these operations efficiently to update the *time-to-live* field on each hop. Moreover, if $p > 0.5$ then this algorithm is robust against a single attacker because there is no way for an attacker to insert a "false" router into the path's valid suffix by contributing

Marking procedure at router R:
for each packet w
let x be a random number from [0..1)
if $x < p$ then,
write R into w.node

Path reconstruction procedure at victim v:
let $NodeTbl$ be a table of tuples (node,count)
for each packet w from attacker
z := lookup w.node in $NodeTbl$
if z != NIL then
increment z.count
else
insert tuple (w.node,1) in $NodeTbl$
sort $NodeTbl$ by count
extract path $(R_i..R_j)$ from ordered node fields in $NodeTbl$

Figure 3: Node sampling algorithm.

more samples than a downstream router, nor to reorder valid routers in the path by contributing more samples than the difference between any two downstream routers.

However, there are also two serious limitations. First, inferring the total router order from the distribution of samples is a slow process. Routers far away from the victim contribute relatively few samples (especially since p must be large) and random variability can easily lead to misordering unless a very large number of samples are observed. For instance, if $d = 15$ and $p = 0.51$, the receiver must receive more than 42,000 packets on average before it receives a *single* sample from the furthest router. To guarantee that the order is correct with 95% certainty requires more than seven times that number.

Second, if there are multiple attackers then multiple routers may exist at the same distance – and hence be sampled with the same probability. Therefore, this technique is not robust against multiple attackers.

4.3 Edge sampling

A straightforward solution to these problems is to explicitly encode *edges* in the attack path rather than simply individual nodes. To do this, we would need to reserve *two* static address-sized fields, *start* and *end*, in each packet to represent the routers at each end of a link, as well as an additional small field to represent the distance of an edge sample from the victim.

When a router decides to mark a packet, it writes its own address into the start field and writes a zero into the distance field. Otherwise, if the distance field is already zero this indicates that the packet was marked by the previous router. In this case, the router writes its own address into the end field – thereby representing the edge between itself and the previous router. Finally, if the router doesn't mark the packet then it always increments the distance field. This somewhat baroque signaling mechanism allows edge sampling to be incrementally deployed – edges are constructed only between participating routers.

The mandatory increment is *critical* to minimize spoofing by an attacker. When the packet arrives at the victim its distance field represents the number of hops traversed since the edge it contains

Cloudflare, Inc. Exhibit 1003 Page 1199

Marking procedure at router R:

```
for each packet w
  let x be a random number from [0..1)
  if x < p then
    write R into w.start and 0 into w.distance
  else
    if w.distance = 0 then
      write R into w.end
    increment w.distance
```

Path reconstruction procedure at victim v:

```
let G be a tree with root v
let edges in G be tuples (start,end,distance)
for each packet w from attacker
  if w.distance = 0 then
    insert edge (w.start,v,0) into G
  else
    insert edge (w.start,w.end,w.distance) into G
remove any edge (x,y,d) with d ≠ distance from x to v in G
extract path (R_i..R_j) by enumerating acyclic paths in G
```

Figure 4: Edge sampling algorithm.

was sampled.[3] Any packets written by the attacker will necessarily have a distance greater or equal to the length of the true attack path. Therefore, a single attacker is unable to forge any edge between themselves and the victim (for a distributed attack, of course, this applies only to the closest attacker) and the victim does not have to worry about "chaff" while reconstructing the valid suffix of the attack path. Consequently, since we no longer use the sampling rank approach to distinguish "false" samples, we are free to use arbitrary values for the marking probability p.

The victim uses the edges sampled in these packets to create a graph (much as in Figure 1) leading back to the source, or sources, of attack. The full algorithm is described in Figure 4. Because the probability of receiving a sample is geometrically smaller the further away it is from the victim, the time for this algorithm to converge is dominated by the the time to receive a sample from the furthest router, $\frac{1}{p(1-p)^{d-1}}$ in expectation, for a router d hops away. However, there is a small probability that we will receive a sample from the furthest router, but not from some nearer router. We can bound this effect to a factor of $ln(d)$ by the following argument: We conservatively assume that samples from all of the d routers appear with the same likelihood as the furthest router. Since these probabilities are disjoint, the probability that a given packet will deliver a sample from some router is at least $dp(1-p)^{d-1}$. Finally, as per the well-known *coupon collector* problem, the number of trials required to select one of each of d equi-probable items is $d(ln(d) + O(1))$ [19]. Therefore, the number of packets, X, required for the victim to reconstruct a path of length d has the following bounded expectation:

$$E(X) < \frac{ln(d)}{p(1-p)^{d-1}}$$

For example, if $p = \frac{1}{10}$, and the attack path has a length of 10, then a victim can typically reconstruct this path after receiving 75 packets from the attacker. While this choice of $p \approx \frac{1}{d}$ is optimal, the convergence time is not overly sensitive to this parameter for the path lengths that occur in the Internet. So long as $p \leq \frac{1}{d}$, the results are generally within a small constant of optimal. In the rest of this paper we will use $p = \frac{1}{25}$ since few paths exceed this length [7, 42, 17]. For comparison, the previous example converges with only 108 packets using $p = \frac{1}{25}$.

This same algorithm can efficiently discern multiple attacks because attackers from different sources produce disjoint edges in the tree structure used during reconstruction. The number of packets needed to reconstruct each path is independent, so the number of packets needed to reconstruct all paths is a linear function of the number of attackers. Finally, edge sampling is also robust. That is it is impossible for any edge closer than the closest attacker to be spoofed, due to the robust distance determination. Conversely, in a distributed attack this also means that it is impossible to trace the contents of any edge *further* away than the closest attacker. As with the ICMP Traceback approach [4], an additional mechanism incorporating a shared secret is required to completely address the problem of attackers spoofing edges.

Of course, a significant practical limitation of this approach is that it requires additional space in the IP packet header and therefore is not backwards compatible. In the next section we discuss a modified version of edge-sampling that addresses this problem, albeit at some cost in performance and a reduction in robustness during large distributed attacks.

5. ENCODING ISSUES

The edge sampling algorithm requires 72 bits of space in every IP packet (two 32-bit IP addresses and 8 bits for distance to represent the theoretical maximum number of hops allowed using IP). It would be possible to directly encode these values into an MPLS label stack [36], to enable traceback within a single homogeneous ISP network. However, our focus is on a heterogeneous environment based purely on IP datagrams. One obvious approach is to store the edge sample data in an IP option, but this is a poor choice for many of the same reasons that the node append algorithm is infeasible – appending additional data to a packet in flight is expensive and there may not be sufficient space to append this data. We could also send this data out-of-band – in a separate packet – but this would add both router and network overhead plus the complexity of a new and incompatible protocol.

Instead, we have developed a modified version of edge sampling that dramatically reduces the space requirement in return for a modest increase in convergence time and a reduction in robustness to multiple attackers. Following an analysis of our algorithm we explore the practical implementation issues and discuss one concrete encoding of this scheme based on overloading the 16-bit IP *identification* field used for fragmentation. Any solution involving such overloading necessarily requires compromises and we stress that our solution reflects only one design point among many potential implementation tradeoffs for this class of algorithm and does not necessarily reflect an optimal balance among them.

5.1 Compressed edge fragment sampling

We use three techniques to reduce per-packet storage requirements while preserving robustness. First, we encode each edge in half the space by representing it as the *exclusive-or* (XOR) of the two IP addresses making up the edge, as depicted in Figure 5. When

[3] It is important that distance field is updated using a saturating addition. If the distance field were allowed to wrap, then the attacker could spoof edges close to the victim by sending packets with a distance value close to the maximum.

Cloudflare, Inc. Exhibit 1003 Page 1200

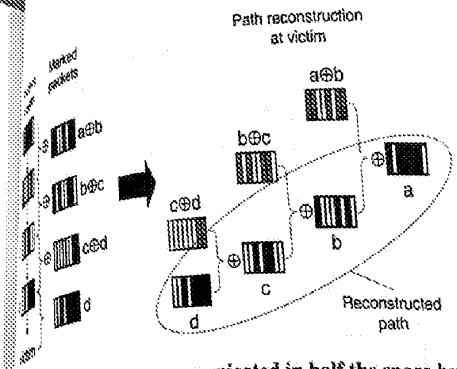


Figure 5: Edge data can be communicated in half the space by using the XOR of the two nodes (i.e. router IP addresses) making up an edge, rather than sending each node separately. In this case the victim receives the messages d, $c \oplus d$, $b \oplus c$, and $a \oplus b$. By XORing these messages together the original path can be reconstructed.

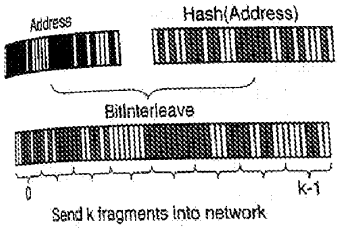


Figure 6: Each router calculates a uniform hash of its IP address once, at startup, using a well-known function. This hash is interleaved with the original IP address (the original address on odd bits, the hash on even bits). The resulting quantity is then broken into k fragments, which the router selects among randomly when marking a packet. Although it is not shown, each of these fragments is further labeled with its offset. The next downstream router uses this offset to select the appropriate fragment to XOR – thereby encoding part of an edge.

some router decides to mark a packet it writes its address, a, into the packet. The following router, b, notices that the distance field is 0 and (assuming it does not mark the packet itself) reads a from the packet, XORs this value with its own address and writes the resulting value, $a \oplus b$, into the packet. We call the resulting value the *edge-id* for the edge between a and b. The edge-ids in the packets received by the victim always contain the XOR of two adjacent routers, except for samples from routers one hop away from the victim, which arrive unmodified. Since $b \oplus a \oplus b = a$, marked packets from the final router can be used to decode the previous edge id, and so on, hop-by-hop until we reach the first router.

Our second modification further reduces our per-packet space requirements by subdividing each edge-id into some number, k, of smaller non-overlapping fragments. When a router decides to mark a packet, it selects one of these fragments at random and stores it in the packet. We use a few additional bits ($log_2 k$) to store the offset of this fragment within the original address – this is necessary to ensure that both fragments making up an edge-id are taken from the same offset. If enough packets are sent by the attacker, the victim will eventually receive all fragments from all edge-ids.

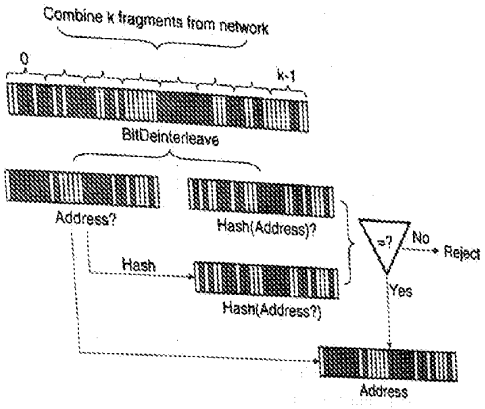


Figure 7: When reconstructing a candidate edge, the victim combines k fragments to produce a bit string. By deinterleaving this string, the address portion and the hash portion are extracted. We recalculate the hash over this address portion using the same hash function used by the router. If the resulting hash is the same as the hash portion extracted, then the address is accepted as valid. This procedure protects against accidentally combining fragments of different edges.

Finally, unlike full IP addresses, edge-id fragments are not unique and multiple fragments from different edge-ids may have the same value. If there are multiple attackers, a victim may receive multiple edge fragments with the same offset and distance. To reduce the probability that we accidentally reconstruct a "false" edge-id by combining fragments from different paths, we add a simple error detection code to our algorithm. We *increase* the size of each router address, and hence each edge-id, by bit-interleaving its IP address with a random hash of itself (depicted in Figure 6). As described earlier, this value is split into fragments, each fragment is selected randomly and stored with an offset, and downstream routers use XOR to combine fragments at the same offset to make up edge-id fragments. The victim constructs *candidate edge-ids* by combining all combinations of fragments at each distance with disjoint offset values. As shown in Figure 7, a candidate edge-id is only accepted if the hash portion matches the data portion for each of its two nodes. As we increase the size of the hash, the probability of a collision is reduced. We provide a describe the full procedure in Figure 8.

The expected number of packets for this algorithm to converge is similar to the edge sampling approach, except now we need k fragments for each edge-id, rather than just one, a total of kd fragments. If we again assume conservatively that each of these fragments is delivered equi-probably with probability $p(1-p)^{d-1}$, the expected number of packets required for path reconstruction is bounded by:

$$E(X) < \frac{k \cdot ln(kd)}{p(1-p)^{d-1}}$$

For example, if there are 8 fragments per edge-id, an attacker is 10 hops away, and $p = \frac{1}{25}$, then a victim can reconstruct the full path after receiving slightly less than 1,300 packets on average. Using techniques similar to those used to show sharp concentration results for the coupon collectors problem, we can further show that the approximate the number of packets required to ensure that a path

Cloudflare, Inc. Exhibit 1003 Page 1201

```
Marking procedure at router R:
    let R' = BitInterleave(R, Hash(R))
    let k be the number of non-overlapping fragments in R'
    for each packet w
        let x be a random number from [0..1)
        if x < p then
            let o be a random integer from [0..k − 1]
            let f be the fragment of R' at offset o
            write f into w.frag
            write 0 into w.distance
            write o into w.offset
        else
            if w.distance = 0 then
                let f be the fragment of R' at offset w.offset
                write f ⊕ w.frag into w.frag
            increment w.distance

Path reconstruction procedure at victim v:
    let FragTbl be a table of tuples (frag,offset,distance)
    let G be a tree with root v
    let edges in G be tuples (start,end,distance)
    let maxd := 0
    let last := v
    for each packet w from attacker
        FragTbl.Insert(w.frag,w.offset,w.distance)
        if w.distance > maxd then
            maxd := w.distance
    for d := 0 to maxd
        for all ordered combinations of fragments at distance d
            construct edge z
            if d ≠ 0 then
                z := z ⊕ last
            if Hash(EvenBits(z)) = OddBits(z) then
                insert edge (z,EvenBits(z),d) into G
                last := EvenBits(z);
    remove any edge (x,y,d) with d ≠ distance from x to v in G
    extract path (R_i..R_j) by enumerating acyclic paths in G
```

Figure 8: Compressed edge fragment sampling algorithm.

can be reconstructed with probability $1 - \frac{1}{c}$ is:

$$\frac{k \cdot ln(kdc)}{p(1-p)^{d-1}}$$

packets. To completely reconstruct the previous path with 95% certainty should require no more than 2150 packets. Many denial-of-service attacks send this many packets in a few seconds.

Finally, we explore the robustness of this algorithm with respect to multiple attackers. For a random hash of length h, the probability of accepting an arbitrarily constructed candidate edge-id is $\frac{1}{2^h}$. In the event that there are m attackers, then at any particular distance d, in the worst case there may be up to m distinct routers.[4] Consequently the probability that any edge-id at distance d is accepted incorrectly is at most:

$$1 - (1 - \frac{1}{2^h})^{m^k}$$

[4]In practice, the number of distinct routers is likely to be smaller for the portion of the path closest to the receiver, since many attackers will still share significant portions of their attack path with one another.

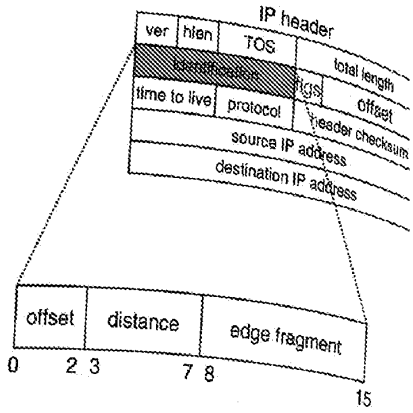


Figure 9: Encoding edge fragments into the IP identific field.

since there are m^k possible combinations of fragments in the v case. For $h = 32$ and $k = 4$ this means that 100 distinct ro at the same distance (i.e. disjoint attack paths) will be reso with no errors with a probability of better than 97%. For $h =$ and $k = 8$, (the values we use for our implementation) the s certainty can only be provided for 10 distinct routers at the s distance. Our use of the XOR function further complicates rec struction since all combinations of XOR values must be trie attack paths diverge. This is somewhat mitigated as the probai ity of propagating an error from a single edge all the way to attacker is is extremely small because the resulting edge-id, wh XORed with the previous edge-id, must again produce a corr hash.

The most significant drawback to this scheme is the large number combinations that must be considered as the multiple attack path diverge. While these combinations can be computed off-line, fo large values of k and m even this can become intractable. For ex ample, even with $k = 8$ and $m = 10$, if the separate attack paths diverge such that there are 10 completely independent edges pe attacker, this will require roughly a billion combinations to be con sidered. Consequently, there is a design tension in the size of k - per-packet space overhead is reduced by a larger k, while compu- tational overhead and robustness benefits from a smaller k.

5.2 IP header encoding

To allow for practical deployment requires that we "overload" ex- isting header fields in a manner that will have minimal impact on existing users. This is a difficult task, especially given that even after prodigious effort we require 16 bits of space. Nonetheless, we believe it possible to obtain this space by overloading the 16-bit IP identification field. This field is currently used to differentiate IP fragments that belong to different packets. We describe our pro- posed encoding below, and then discuss the issues of backwards- compatibility that it raises. However, we note that because the is- sue of backwards-compatible encoding is largely separate from our traceback algorithms, we could adopt any reasonable encoding that comes to light.

Figure 9 depicts our choice for partitioning the identification field: 3 offset bits to represent 8 possible fragments, 5 bits to represent

Cloudflare, Inc. Exhibit 1003 Page 1202

...and 8 bits for the edge fragment. We use a 32-bit ... enables the size of each router address to 64 bits. This ... separate fragments are needed to represent each edge ... indicated by a unique offset value. Finally, 5 bits ... represent 31 hops, which is more than almost all ... [7, 42, 17].[5]

...reader will note that this layout is chosen to allow ...performance software implementation of our algorithm, ...had a low per-packet router overhead. In the com... the only modification to the packet is to increment its ... field. Because of its alignment within the packet, this in... precisely offsets the required decrement of the time-to-live ...mented by each router [1]. Consequently, the header ...does not need to be altered at all and the header manipu... could be even lower than in current software-based ... simply an addition to the distance field, a decrement to the ... and a comparison to check if either has overflowed. In ... case, our algorithm must read the IP identification field, ...an edge fragment and XOR it, and fold the write-back into ...checksum update procedure (a few ALU operations). ...for modern ASIC-based routers these optimizations are ...ssary.

...of the IP identification field, we must address issues of ...-compatibility for IP fragment traffic. Ultimately, there ...perfect solution to this problem and we are forced to make ...ses that disadvantage fragmented traffic. Fortunately, re... measurements suggest that less than 0.25% of packets are ...mented [40, 10]. Moreover, it has long been understood that ...-layer fragmentation is detrimental to end-to-end perfor... [27] so modern network stacks implement automatic MTU ...very to prevent fragmentation regardless of the underlying ...ia [30]. Consequently, we believe that our encoding will inter... seamlessly with existing protocol implementations in the ...majority of cases.

...ever, there is a small but real fraction of legitimate traffic that ...fragmented, and we wish to ensure that it is not affected by ...modifications to the extent that this is possible. Normally if a ...et is fragmented, its identification field is copied to each frag... so the receiver can faithfully reassemble the fragments into ...original packet. Our marking procedure can violate this prop... in one of two ways: by writing different values into the iden...ication fields of fragments from the same datagram or by writing ...same values into the identification fields of fragments from dif...ent datagrams. These two problems present different challenges ...have different solutions.

...t, a datagram may be fragmented *upstream* from a marking ...er. If the fragment is subsequently marked and future fragments ...m the same datagram are not marked consistently then reassem... may fail or data may be corrupted. While the simplest solu...n to this problem is to simply not mark fragments, an adversary ...ld quickly learn to evade traceback by exploiting this limita...n. In fact, some current denial-of-service attacks already use IP ...gments to exploit errors in host IP reassembly functions [12]. In...ead, we propose an alternative marking mechanism for fragments. ...e use a separate marking probability, q, for fragments. When we

[5] It is also reasonable to turn off marking on any routers that cannot be directly connected to an attacking host (e.g. core routers). This both reduces the convergence time, and increases the "reach" of the distance field.

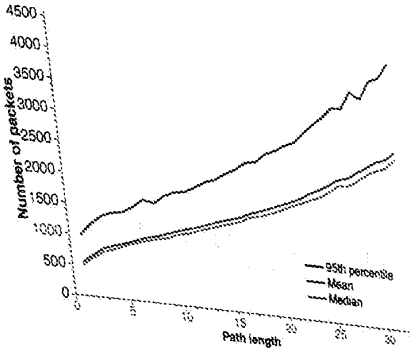


Figure 10: Experimental results for number of packets needed to reconstruct paths of varying lengths. The marking probability, p, is set to $\frac{1}{25}$. Each path length result represents the results of 1,000 independent simulation runs.

decide to mark a fragment, we prepend a new ICMP "echo reply" header, along with the *full* edge data - truncating the tail of the packet. This ICMP packet is considered "marked" and its distance field is set to zero, thereby guaranteeing that the distance field reflects the number of edges traversed on the way to the victim. The packet is consequently "lost" from the standpoint of the receiver, but the edge information is delivered in a way that does not impact legacy hosts. Because we can use the full edge sampling algorithm, q can be more than an order of magnitude smaller than p and yet achieve the same convergence time. This solution increases the loss rate of fragmented flows somewhat (more substantially for longer paths) but preserves the integrity of the data in these flows.

A more insidious problem is presented by fragmentation that occurs *downstream* from a marking router. If a marked packet is fragmented, but one of the fragments is lost, then the remaining fragments may linger in the victim's reassembly buffer for an extended period [5]. Future packets marked by the same router can have the same IP identification value and consequently may be incorrectly reassembled with the previous fragments. One possibility is to leave this problem to be dealt with by higher layer checksums. However, not all higher layer protocols employ checksums, and in any case it is dangerous to rely on such checksums because they are typically designed only for low residual error rates. Another solution is to set the *Don't Fragment* flag on every marked packet. Along rare paths that require fragmentation, this solution will degrade communication between hosts not using MTU path discovery, and may filter marked packets if a reduced MTU edge is close to the victim, but it will never lead to data corruption.

5.3 Experience

We have implemented the marking and reconstruction portions of our algorithm and have tested it using a simulator that creates random paths and originates attacks. In Figure 10 we graph the mean, median and 95th percentile for the number of packets required to reconstruct paths of varying lengths over 1,000 random test runs for each length value. We assume a marking probability of $\frac{1}{25}$. Note that while the convergence time is theoretically exponential in the path length, all three lines appear linear due to the finite path length and appropriate choice of marking probability.

We see that most paths can be resolved with between one and two

Cloudflare, Inc. Exhibit 1003 Page 1203

thousand packets, and even the longest paths can be resolved with a very high likelihood within four thousand packets. To put these numbers in context, most flooding-style denial of service attacks send many hundreds or thousands of packets each second. The analytic bounds we described earlier are conservative, but in our experience they are no more than 30% higher than our experimental results.

6. LIMITATIONS AND FUTURE WORK

There are still a number of limitations and loose ends in our approach. We discuss the most important of these here:

- backwards compatibility,
- distributed attacks,
- path validation, and
- approaches for determining the attack origin.

6.1 Backwards compatibility

The IP header encoding as we have described it has several practical limitations. It negatively impacts users that require fragmented IP datagrams and is currently incompatible with parts of IPsec [28] (the Authentication Header provides cryptographic protection for the identification field and therefore the field cannot be safely modified by routers). These problems are hardly unique to traceback and are inherent limitations that come about from attempting to coexist with or co-opt protocol features that did not anticipate a new use. One way to address this issue, originally proposed by John Hawkinson, is to selectively enable traceback support in response to operational needs. A "request for traceback" from a particular network could be encoded as a BGP attribute in the network's route advertisement. Routers receiving such an advertisement would enable traceback support on packets destined for that network. Since a network requesting such support is presumably already suffering under an attack, any minor service degradation for fragmented or authenticated flows would be acceptable.

Finally, our scheme does not address implementation in IPv6, the proposed successor to IPv4, which does not have an identification field [18]. While we do not attempt to propose a complete encoding here, we believe that the same techniques we have proposed could also be employed within IPv6, perhaps by overloading the 24-bit *flow label* field (without any further modifications this would result in roughly a factor of three increase in the number of packets required to reconstruct a path).

6.2 Distributed attacks

For large distributed attacks, the practical implementation we have described has serious limitations due to the difficulty in correctly grouping fragments together. Consequently, the probability of misattributing an edge, as well as the amount of state needed to evaluate this decision, increases very quickly with the fan-out of an attack. There is significant future work in designing alternative encoding methods that scale their robustness as they receive more data.

6.3 Path validation

Some number of the packets sent by the attacker are unmarked by intervening routers. The victim cannot differentiate between these packets and genuine marked packets. Therefore an attacker could insert "fake" edges by carefully manipulating the identification fields in the packets it sends. While the distance field prevents an attacker from spoofing edges between it and the victim - what we call the *valid suffix* - nothing prevents the attacker from spoofing extra edges past the end of the true attack path.

There are several ways to identify the valid suffix within a path generated by the reconstruction procedure. With minimal knowledge of Internet topology one can differentiate between routers that belong to transit networks (e.g. ISPs) and those which belong to stub networks (e.g. enterprise networks). Generally speaking, a valid path will never enter a stub network and then continue into a transit network. Moreover, simple testing tools such as `traceroute` should enable a victim to determine if two networks do, in fact, connect. More advanced network maps [8, 23] can resolve this issue in an increasing number of cases.

A more general mechanism is to provide each router with a "secret" that is sent along with each marked packet (perhaps in the single unallocated bit in the IP flags field). When the victim wants to validate a router in the path, it contacts the associated network (possibly out of band, via telephone or e-mail) and obtains the secret used by the router at the time of the attack. To guard against replay, the secret can be time-varying and hashed with the packet contents. Since the attacker will not know the router's secret, they will not be able to include the proper bit in their forged edge-id fragments. By eliminating edge-ids for which the secret in their constituent fragments can not be validated, we can prune a candidate attack path to only include the valid suffix.

6.4 Attack origin detection

While our IP-level traceback algorithm could be an important part of the solution for stopping denial-of-service attacks, it is by no means a complete solution. Our algorithm attempts to determine the approximate origin of attack traffic - in particular, the earliest traceback-capable router involved in forwarding attack traffic from the source that directly generated it. As mentioned earlier, there are a number of reasons why this may differ from the true source of the attack: attackers can hide their true identities by "laundering" attacks through third parties, either indirectly (e.g. smurf attacks [13] or DNS reflectors [15]) or directly via compromised "stepping stone" machines or IP-in-IP tunnels. While there is ongoing work on following attackers through intermediate hosts [45, 39], there are still significant challenges in developing a generally applicable and universally deployable solution to this problem. One interesting possibility enabled by the packet marking approach is to extend traceback across "laundering points". For example, identifying marks could be copied from a DNS request packet into the associated DNS reply, thereby allowing the victim to trace the full causal path.

Even in absence of such "laundering", our approach does not reveal the actual host originating the attack. Moreover, since hosts can forge both their IP source address and MAC address the origin of a packet may never be explicitly visible. On shared media such as FDDI rings, this problem can only be solved by explicit tracing. However, on point-to-point media, the input port a packet arrives on is frequently enough to determine its true origin. On other media, there may be a MAC address, cell number, channel, or other hint that would help to locate the attack origin. In principle, our algorithm could be modified to report this information by occasionally marking packets with a special edge-id representing a link between the router and the input port on which the packet arrived (or other

Cloudflare, Inc. Exhibit 1003 Page 1204

...ational. We have not explored the design of such a ...ocol.

...ack is only effective at finding the source of *attack* ...ssarily the *attacker* themselves. Stopping an attack ...ient to eliminate an immediate problem, but long term ...may require a legal remedy and therefore the forensic ...ermine an attacker's identity. Even with perfect trace- ...unambiguously identifying a sufficiently skilled and ...acker is likely to require cooperation from law enforce- ...telecommunications organizations.

ONCLUSION

...per we have argued that denial-of-service attacks moti- ...development of improved traceback capabilities and we ...sed traceback algorithms based on packet marking in the ... We have shown that this class of algorithm, best embod- ...ge sampling, can enable efficient and robust multi-party ...s that can be incrementally deployed and efficiently im- ...ed. As well, we have developed variant algorithms that ...convergence time and robustness for reduced per-packet ...irements. Finally, we have suggested one potential de- ...ense strategy using such an algorithm based on overloading ...IP header fields and we have demonstrated that this imple- ...ion is capable of fully tracing an attack after having received ...few thousand packets. We believe our solution represents ...sible first step towards an automated network-wide traceback ...s. Several areas remain to be addressed in future work, such ...e combination of widely distributed attacks and points of indi- ...ion such as reflectors.

cknowledgments

...is paper has benefited from conversations with many different ...ple - far more than can be acknowledged completely here. Still ...would like to particularly thank Bill Cheswick and Hal Burch ...early access to their work in this area, Sean McCreary and KC ...Claffy for access to their packet trace data, Alec Wolman for help ...th graph, Przemek Pardyak, Mike Swift, Neil Spring, Gretta ...Bartek, Robert Grimm and Geoff Voelker for commenting on early ...afts of the paper, and constructive feedback from John Hawkin- ...on, Robert Stone, Jeff Mogul, Robert Moskowitz, Greg Minshall, ...ngo Li, Curtis Villamizar, Steve Corbato, and countless others. Fi- ...lly, we thank both the anonymous reviewers and our shepherds, ...Vern Paxson and Craig Partridge, for their efforts.

REFERENCES

[1] F. Baker. Requirements for IP Version 4 Routers. RFC 1812, June 1995.

[2] G. Banga, P. Druschel, and J. Mogul. Resource Containers: A New Facility for Resource Management in Server Systems. In *Proceedings of the 1999 USENIX/ACM Symposium on Operating System Design and Implementation*, pages 45-58, Feb. 1999.

[3] S. M. Bellovin. Security Problems in the TCP/IP Protocol Suite. *ACM Computer Communications Review*, 19(2):32-48, Apr. 1989.

[4] S. M. Bellovin. ICMP Traceback Messages. Internet Draft: draft-bellovin-itrace-00.txt, Mar. 2000.

[5] R. Braden. Requirements for Internet Hosts - Communication Layers. RFC 1122, Oct. 1989.

[6] H. Burch and B. Cheswick. Tracing Anonymous Packets to Their Approximate Source. Unpublished paper, Dec. 1999.

[7] R. L. Carter and M. E. Crovella. Dynamic Server Selection Using Dynamic Path Characterization in Wide-Area Networks. In *Proceedings of the 1997 IEEE INFOCOM Conference*, Kobe, Japan, Apr. 1997.

[8] B. Cheswick and H. Burch. Internet Mapping Project. `http://cm.bell-labs.com/who/ches/map/index.html`, 2000.

[9] Cisco Systems. Configuring TCP Intercept (Prevent Denial-of-Service Attacks). Cisco IOS Documentation, Dec. 1997.

[10] K. Claffy and S. McCreary. Sampled Measurements from June 1999 to December 1999 at the AMES Inter-exchange Point. Personal Communication, Jan. 2000.

[11] Computer Emergency Response Team. CERT Advisory CA-96.26 Denial-of-Service Attack via pings. `http://www.cert.org/advisories/CA-96.26.ping.html`, Dec. 1996.

[12] Computer Emergency Response Team. CERT Advisory CA-97.28 IP Denial-of-Service Attacks. `http://www.cert.org/advisories/CA-97.28.smurf.html`, Dec. 1997.

[13] Computer Emergency Response Team. CERT Advisory CA-98.01 "smurf" IP Denial-of-Service Attacks. `http://www.cert.org/advisories/CA-98.01.smurf.html`, Jan. 1998.

[14] Computer Emergency Response Team. CERT Advisory CA-2000-01 Denial-of-Service Developments. `http://www.cert.org/advisories/CA-2000-01.html`, Jan. 2000.

[15] Computer Emergency Response Team. CERT Incident Note IN-2000-04 Denial-of-Service Attacks using Nameservers. `http://www.cert.org/incident_notes/IN-200-04.html`, Apr. 2000.

[16] Computer Security Institute and Federal Bureau of Investigation. 1999 CSI/FBI Computer Crime and Security Survey. Computer Security Institute publication, Mar. 1999.

[17] Cooperative Associationfor Internet Data Analysis. Skitter Analysis. `http://www.caida.org/Tools/Skitter/Summary/`, 2000.

[18] S. Deering. Internet protocol, version 6 (ipv6). RFC 2460, Dec. 1998.

[19] W. Feller. *An Introduction to Probability Theory and Its Applications (2nd edition)*, volume 1. Wiley and Sons, 1966.

[20] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing. RFC 2267, Jan. 1998.

[21] J. Glave. Smurfing Cripples ISPs. Wired Technolgy News: (`http://www.wired.com/news/news/technology/story/9506.html`), Jan. 1998.

Cloudflare, Inc. Exhibit 1003 Page 1205

[22] I. Goldberg and A. Shostack. Freedom Network 1.0 Architecture and Protocols. Zero-Knowledge Systems White Paper, Nov. 1999.

[23] R. Govindan and H. Tangmunarunkit. Heuristics for Internet Map Discovery. In *Proceedings of the 2000 IEEE INFOCOM Conference*, Tel Aviv, Israel, Mar. 2000.

[24] L. T. Heberlein and M. Bishop. Attack Class: Address Spoofing. In *1996 National Information Systems Security Conference*, pages 371–378, Baltimore, MD, Oct. 1996.

[25] J. D. Howard. *An Analysis of Security Incidents on the Internet*. PhD thesis, Carnegie Mellon University, Aug. 1998.

[26] P. Karn and W. Simpson. Photuris: Session-Key Management Protocol. RFC 2522, Mar. 1999.

[27] C. Kent and J. Mogul. Fragmentation Considered Harmful. In *Proceedings of the 1987 ACM SIGCOMM Conference*, pages 390–401, Stowe, VT, Aug. 1987.

[28] S. Kent and R. Atkinson. Security architecture for the internet protocol. RFC 2401, Nov. 1998.

[29] C. Meadows. A Formal Framework and Evaluation Method for Network Denial of Service. In *Proceedings of the 1999 IEEE Computer Security Foundations Workshop*, Mordano, Italy, June 1999.

[30] J. Mogul and S. Deering. Path MTU Discovery. RFC 1191, Nov. 1990.

[31] R. T. Morris. A Weakness in the 4.2BSD Unix TCP/IP Software. Technical Report Computer Science #117, AT&T Bell Labs, Feb. 1985.

[32] V. Paxson. End-to-End Routing Behavior in the Internet. *IEEE/ACM Transactions on Networking*, 5(5):601–615, Oct. 1997.

[33] C. Perkins. IP Mobility Support. RFC 2002, Oct. 1996.

[34] J. Postel. Internet Protocol. RFC 791, Sept. 1981.

[35] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous Connections and Onion Routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, May 1998.

[36] E. C. Rosen, Y. Rekhter, D. Tappan, D. Farinacci, G. Fedorkow, T. Li, and A. Conta. MPLS Label Stack Encoding. Internet Draft: draft-ietf-mpls-label-encaps-07.txt (expires March 2000), Sept. 1998.

[37] G. Sager. Security Fun with OCxmon and cflowd. Presentation at the Internet 2 Working Group, Nov. 1998.

[38] O. Spatscheck and L. Peterson. Defending Against Denial of Service Attacks in Scout. In *Proceedings of the 1999 USENIX/ACM Symposium on Operating System Design and Implementation*, pages 59–72, Feb. 1999.

[39] S. Staniford-Chen and L. T. Heberlein. Holding Intruders Accountable on the Internet. In *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, pages 39–49, Oakland, CA, May 1995.

[40] I. Stoica and H. Zhang. Providing Guaranteed Services Without Per Flow Management. In *Proceedings of the 1999 ACM SIGCOMM Conference*, pages 81–94, Boston, MA, Aug. 1999.

[41] R. Stone. CenterTrack: An IP Overlay Network for Tracking DoS Floods. In *to appear in Proceedings of thje 2000 USENIX Security Symposium*, Denver, CO, July 2000.

[42] W. Theilmann and K. Rothermel. Dynamic Distance Maps of the Internet. In *Proceedings of the 2000 IEEE INFOCOM Conference*, Tel Aviv, Israel, Mar. 2000.

[43] C. Villamizar. Personal Communication, Feb. 2000.

[44] M. Vivo, E. Carrasco, G. Isern, and G. O. Vivo. A review of port scanning techniques. *ACM Computer Communications Review*, 29(2):41–48, Apr. 1999.

[45] Y. Zhang and V. Paxson. Stepping Stone Detection. In *to appear in Proceedings of thje 2000 USENIX Security Symposium*, Denver, CO, July 2000.

Cloudflare, Inc. Exhibit 1003 Page 1206

Declaration of Dr. Stefan Savage

ATTACHMENT K

Networking Conferences Statistics

This page is an attempt to gather year-over-year statistics for some of the major systems and networking conferences.

For an analysis of this data see Utkarsh Goel, Clemente Izurieta, Mike P. Wittie, "Understanding Factors Influencing the Citation Count of Networking Conference Papers" in ISCA Software Engineering and Data Engineering (SEDE), October 2015.

For FOG and Edge Computing conference stats, see the Fog and Edge Computing Conferences and Journals Page.
For Architecture conference stats (ISCA, Micro, HPCA, ASPLOS), see the Prichard, Scopel, Hill, Sohi, and Wood Excel File.
For Database conference stats, see Peter Aper's Stats Page.
For Computer Security conference stats, see Guofei Gu's Computer Security Conference Ranking and Statistics Page.
For Theory (and some Computational Biology) conference stats, see Florian Sikora's TCS page.

ACNS, AIMS, ANCS, APCC, CAMAD, CCS, CCGrid, CCNC, CloudCom, CloudNet, Cluster, CNS, CNSM, CNSR, COMSNETS, CoNext, CrossCloud, DAIS, DEBS, DRCN, DSN, DSOM, ECRTS, EDOC, EPEW, eScience, ESOCC, EuCNC, Euro-Par, EUNICE, EuroSys, EWSN, FMN, FMOODS, FMOODS/FORTE, FORTE, FORTE/PSTV, Global Internet, Globecom, Grid, HiPC, HotPlanet, HPCC, HPDC, HPSR, I3E, ICASSP, ICBC, ICC, ICCC, ICCCN, ICCPS, ICDCS, ICDIM, ICIN, ICODP, ICNC, ICNP, ICPADS, ICPP, ICQT, ICS, ICUFN, IM, IMC, Infocom, IPCCC, IPDPS, IPOM, IPS-MoMe, IPSN, IPTPS, ISCC, ISPDC, ITC, IWAN, IWCMC, IWPTS, IWTCS, IWQoS, LANC, LANMAN, LANOMS, LCN, MASCOTS, MASS, MedHocNet, MELT, Middleware, MMCN, MMNS, Mobiarch, Mobicom, Mobihoc, MobiOpp, Mobiquitous, Mobisys, Mobiwac, MSN, MSWiM, Multimedia, NCA, NESEA, NetGames, NetSoft, Networking, NEW2AN, NIME, NDSS, NGC, NGMAST, NOF, NOMS, NOSSDAV, NPC, NSDI, NTMS, OPENARCH, OOPSLA, OSDI, P2P, PAM, PerCom, Performance, PEMWN, PIMRC, PSTV, PVM/MPI, PWC, Q2SWinet, QofIS, RTAS, RTSS, SAINT, SASO, SAWN, SBAC-PAD, SCNS, SECON, Security, Security & Privacy, SenSys, Sigcomm, Sigmetrics, SmartGridComm, SOSP, SOSR, Supercomputing, SUSTAINIT, TechSym, Testcom, Testcom/Fates, TMA (workshop), TMA, UBICOMM, USENIX, USITS, VANET, VTC, WiOpt, WiSec, WCITD, WCNC, WCW, WD, WICON, WiMob, WMASH, WMI, WMNC, WONS, WoWMoM, WORM, WPMC, WQoSR, WWIC, and WWW.

If you do want to add information, be sure to send *all* of the following:

- URL of the Conference
- Papers Submitted
- Papers Accepted
- Acceptance Ratio
- Number of Parallel Tracks, and
- Number of Attendees

Send info to *almeroth@cs.ucsb.edu*. Note that I always send a response. If you do not receive a response, try again, your email might have been accidently filtered.

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ACNS	2003	191	32	16.8%	2	~100
	2004	297	36	12.1%	2	~150
	2005	158	35	22.2%	2	~100

Cloudflare, Inc. Exhibit 1003 Page 1208

	2006	218	33	15.1%	2	~100
	2007	260	31	11.9%	2	~60
	2008	131	30	22.9%	2	~80
	2009	150	32	21.3%	2	~100
	2010	178	32	18.0%	2	~90
	2011	172	31	18.0%	2	~90
	2012	192	33	17.2%	2	~120

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
AIMS	2007	54	13	24.1%	1	83
	2008	33	13	39.4%	1	50
	2009	28	12	42.0%	1	45
	2010	27	9	33.3%	1	41
	2011	37	12	32.4%	1	59
	2012	23	10	43.5%	1	45
	2013	32	11	34.4%	1	44
	2014	29	9	31.0%	1	44
	2015	22	7	31.8%	1	52
	2016	22	7	31.8%	1	45
	2017	24	8	33.3%	1	51

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ANCS (IEEE/ACM)	2005	100	23	23.0%	1	???
	2006	97	19	19.6%	1	???
	2007	??	??	??.?%	1	???
	2008	??	??	??.?%	1	???
	2009	73	17	23.3%	1	???
	2010	??	??	??.?%	1	???
	2011	??	??	??.?%	?	???
	2012	64	18	28.1%	1	80
	2016	25 (+25 short)	6 (+6 short)	24.0%	1	50

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
APCC (IEEE)	2015	232 (+30)	115 (+13)	49.6%	???	???
	2016	297	126	42.4%	3	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
CAMAD (IEEE)	2015	126	63	50.0%	2	~80

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
CCS (ACM)	1993	60	27	45.0%	1	???
	1997	64	17	26.6%	1	???
	1998	85	17	20.0%	1	???
	1999	83	16	19.3%	1	???
	2000	131	28	21.4%	1	???

Cloudflare, Inc. Exhibit 1003 Page 1209

	2001	153	27	17.6%	1	???
	2002	153	27	17.6%	1	???
	2006	256	38	14.8%	1	???
	2007	302	55	18.2%	2	???
	2008	280	51	18.2%	2	???
	2009	315	58	18.4%	2	533
	2010	320	55	17.2%	2	???
	2011	429	60	14.0%	2	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
CCGrid (IEEE)	2001	126	48	38.1%	3	162
	2002	108	27	25.0%	3	220
	2003	114	39	34.2%	3	300
	2004	138	38	27.5%	3	200+
	2005	235	76	32.3%	3	300
	2006	257	61	23.7%	3	189
	2007	218	72	33.0%	3	330
	2008	201	65	32.4%	3	260
	2009	271	57	21.0%	3	170
	2010	219	51	23.3%	3	200
	2011	189	55	29.1%	3	150+
	2012	302	83	27.5%	3	220
	2013	257	57	22.2%	3	160
	2014	283	54	19.1%	3	229
	2015	268	69	25.7%	3	250
	2021	230	60	26.1%	3	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
CCNC (IEEE)	2010	320	??	~30%-33%	1	???
	2011	305	95	31.1%	6	355
	2012	200	80	40.0%	6	???
	2013	313	94	30.0%	6	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
CloudCom (IEEE)	2009	~200	54	27.0%	1	??
	2010	207	51	24.6%	1	??
	2011	237	57	24.1%	1	??
	2012	310 (407 total)	55 (+25 short)	17.7%	4	???
	2013	337	60 (+34 short)	17.8%	4	???
	2014	~303	54 (+31 short)	17.8%	5	265
	2015	221	50 (+13 short)	22.6%	5	???
	2016	193	51 (+14 short)	26.4%	7	~200
	2017	136	40 (+5 short)	29.4%	7	~100

Cloudflare, Inc. Exhibit 1003 Page 1210

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
CloudNet (IEEE)	2012	63	25	39.7%	1	91
	2013	75	21	28.0%	1	90
	2015	108	36	33.3%	1-2	~90

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Cluster (IEEE)	2000	144	34	23.6%	1.5	161
	2001	74	35	47.3%	1.5	304
	2002	116	45	38.8%	3	283
	2003	164	48	29.3%	3	176
	2004	150	48	32.0%	2	200
	2005	138	45	32.6%	2	~200
	2006	127	42	33.1%	2	210
	2007	106	42	39.6%	2	195
	2008	92	28	30.4%	2	~200
	2009	100	48	48.0%	3	~210
	2010	107	33	30.8%	3	~200
	2011	140	39	27.8%	2	~300
	2012	201	58	28.8%	4	~300

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
CNS	2015	171	48	28.1%	2	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Resulted from merging Manweek, DSOM, IPOM, and MMNS.						
CNSM	2010	176	27	15.3%	1	180
	2011	164	24	14.6%	1	216
	2012	97	15	15.5%	1	107
	2013	116	21	18.1%	1	179
	2014	105	18	17.1%	1	152
	2015	102	18	17.6%	1	126
	2016	116	17	14.7%	1	130
	2017	136	24	17.6%	1	160
	2018	100	19 (+15 short)	19.0%	1	~150
	2019	127	21	16.5%	1	134
	2020	106	21 (+22 short +31 poster)	19.8%	1	152

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
CNSR	2004	72	41	56.9%	2	???
	2005	70+	52	74.3%	2	???
	2006	81	36	44.4%	2	70++
	2007	102	54	52.9%	2	???
	2008	184	83	45.1%	2	???
	2009	157	76	48.4%	2	???

Cloudflare, Inc. Exhibit 1003 Page 1211

	2010	97	54	55.7%	2	???
	2011	82	41	50.0%	2	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
COMSNETS	2009	230	40	17.4%	2	???
	2010	153	31	20.3%	2	???
	2011	???	??	??.?%	2	???
	2012	165	46	27.9%	2	???
	2013	144	38	26.4%	2	???
	2014	210	37	17.6%	2	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
CoNext	2005	110	19	17.2%	1	100
	2006	86	19	22.1%	1	100
	2007	138	28	20.3%	1	155
	2008	166	29	17.4%	1	154
	2009	170	29	17.1%	1	152
	2010	147	28	19.1%	1	136
	2011	159	30	18.9%	1	???
	2012	176	31	17.6%	1	???
	2013	143 (+83 short)	29 (+15 short)	20.2% (18.1% short)	1	160
	2014	133 (+54 short)	27 (+10 short)	20.3% (18.5% short)	1	???
	2015	136 (+60 short)	32 (+9 short)	23.5% (15.0% short)	1	???
	2016	160 (+39 short)	30 (+5 short)	18.8% (12.8% short)	1	190
	2017	171 (+51 short)	33 (+7 short)	19.3% (13.7% short)	1	230
	2018	142 (+43 short)	28 (+4 short)	19.7% (9.3% short)	1	~180
	2019	149 (+41 short)	23 (+9 short)	15.7% (21.9% short)	1	~150

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
CrossCloud (IEEE)	2014	18	9	50.0%	1	20
	2015	14	7	50.0%	1	29
	2016	15	8	53.3%	1	45
	2017	17	8	47.1%	1	18
	2018	6	5	83.3%	1	12

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
DAIS (IFIP)	1997	31	16 (+6 short)	51.6%	1	75
	1999	88	25 (+7 short)	28.4%	1	120

Cloudflare, Inc. Exhibit 1003 Page 1212

	2001	65	16 (+10 short)	24.6%	1	110
	2003	76 (+12 short)	21 (+5 short)	27.6%	1	100
	2005	44	19 (+3 short)	43.2%	1	24
	2006	78	26	33.3%	1	34
	2007	743	24	32.4%	1	35
	2008	66	24	36.4%	1	35
	2009	33	12	36.4%	1	18
	2010	53	17	32.1%	1	35

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
DEBS (ACM)	2007	??	14	??.?%	1	???
	2008	67	20	29.9%	1	???
	2009	60	16	26.7%	1	82
	2010	64	16	25.0%	2	105
	2011	95	23	24.2%	1	121
	2012	103	17	16.5%	1	88

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
DRCN	1998	??	??	??.?%	2	???
	2000	??	??	??.?%	2	???
	2001	??	??	??.?%	2	???
	2003	??	60	??.?%	2	142
	2005	??	??	??.?%	2	???
	2007	63	50	79.4%	2	97
	2009	85	42	49.4%	2	100
	2011	42	26	61.9%	1	69
	2013	62	25	40.3%	1	81
	2014	45	19	42.2%	1	72
	2015	44	21	47.7%	1	91
	2016	57	20	35.1%	1	84
	2017	33	16	48.5%	1	42
	2018	26	10	38.5%	1	37
	2019	38 (+7 special)	15 (+4 short)(+3 special)	39.5%	1	35
	2020	56	24 (+4 posters)	42.8%	1	82

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
DSN	2000	131	56	42.7%	3	280
	2001	153	53	34.6%	3	297
	2002	213	75	35.2%	3	340
	2003	225	73	32.4%	3	324
	2004	377	83	22.0%	3	357
	2005	319	78	24.5%	3	353

Cloudflare, Inc. Exhibit 1003 Page 1213

	2006	283	57	20.1%	2	280
	2007	307	80	26.4%	2	366
	2008	237	58	24.5%	2	230
	2009	260	64	24.6%	2	270
	2010	273	67	24.5%	2	260
	2011	253	52	20.6%	2	240
	2012	236	51	19.9%	2	238
	2013	220	46	20.9%	2	185
	2014	181	56	29.8%	2	196
	2015	229	50	21.8%	2	160
	2016	259	58	20.5%	2	260
	2017	221	54	24.5%	2	155
	2018	228	62	26.7%	2	250
	2019	252	54	21.4%	2	193
	2020	291	48	16.5%	2	357

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
DSOM (IFIP/IEEE)	1997	40	24	60.0%	1	???
	1998	48	24	50.0%	1	???
	1999	54	20	37.0%	1	80
	2000	65	21	32.3%	1	???
	2001	76	26	34.2%	1	72
	2002	41	16	39.0%	1	41
	2003	86	20	23.3%	1	54
	2004	90	20	22.2%	1	54
	2005	87	23	26.4%	1	80
	2006	85	21	24.7%	1	95
	2007	64	20	31.3%	1	87
	2008	45	14	31.1%	1	90
	2009	50	14	28.0%	1	123
Merged with Manweek, IPOM, and MMNS into CNSM.						

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ECRTS	2002	92	27	29.3%	1	???
	2003	102	32	31.4%	1	???
	2004	115	29	25.2%	1	???
	2005	145	26	17.9%	1	???
	2006	105	25	23.8%	1	???
	2007	90	27	30.0%	1	???
	2008	109	32	29.4%	1	???
	2009	102	26	25.5%	1	???
	2010	112	27	24.1%	1	???

Cloudflare, Inc. Exhibit 1003 Page 1214

	2011	119	25	21.0%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
EDOC (IEEE)	2004	66	24	36.4%	1	91
	2005	173	25	14.4%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
EPEW	2007	53	21	39.6%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
eScience (IEEE)	2005	175	54	30.9%	3	160+
	2006	160	60	37.5%	3	200+
	2007	206	60	29.1%	3	200+
	2008	206	60	29.1%	3	268
	2009	177	51	28.8%	3	248
	2010	142	44	31.0%	3	~210
	2011	110	54	49.1%	3	294
	2012	154	48	31.2%	3	~225
	2013	98	41	41.8%	3	259
	2014	101	38	37.6%	2	210
	2015	123	31	25.2%	2	160
	2016	75	30	40.0%	2	~100

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ESOCC	2014	38	8 (+4 short)	21.1%	1	54

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
EuCNC	2015	226	110	48.7%	6	530
	2016	166	80	48.5%	7	540
	2017	317	146	57.5%	6	600

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Euro-Par	1995	180	50	27.8%	2	???
	1996	383	214	55.9%	5	???
	1997	297	154	51.9%	5	???
	1998	238	129	54.2%	5	???
	1999	343	188	54.8%	5	???
	2000	326	167	51.2%	5	???
	2001	207	111	53.7%	4	???
	2002	265	125	47.0%	4	???
	2003	338	155	46.0%	5	???
	2004	352	124	35.2%	5	???
	2005	388	121	31.2%	5	???
	2006	290	110	37.9%	3	???
	2007	333	89	26.7%	3	???
	2008	264	89	33.7%	4	309

Cloudflare, Inc. Exhibit 1003 Page 1215

	2009	256	85	33.2%	4	273
	2010	256	90	35.1%	4	???
	2011	271	81	29.9%	4	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
EUNICE	2009	71	23	32.4%	1	65
	2010	50	24	48.0%	1	54
	2011	26	16	61.5%	1	50
	2012	48	23	47.9%	1	51

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
EuroSys	2006	144	29	20.1%	1	177
	2007	131	29	22.1%	1	220
	2008	131	24	18.3%	1	???
	2009	148	25	16.9%	1	???
	2010	140	27	19.3%	1	???
	2011	154	24	15.6%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
EWSN	2004	76	24	31.6%	1	???
	2005	>100	35	~35.0%	1	???
	2006	133	21	15.8%	1	???
	2007	164	22	13.4%	1	???
	2008	110	23	20.9%	1	???
	2009	145	23	15.9%	1	~140
	2010	109	21	19.3%	1	???
	2011	87	14	16.1%	1	107
	2012	78	16	25.5%	1	???
	2013	51	12	23.5%	1	???
	2014	50	12	24.0%	1	84

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
FMN	2008	72	18	25.0%	1	30
	2009	64 (+18 short)	16 (+12 short)	25.0%	1	60
	2010	27	9	33.3%	1	30
	2012	28	9	32.1%	1	30

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
FMOODS (IFIP) (joined with FORTE in 2009)	1996	???	19 (+7 short)	??.?%	1	???
	1997	58	20 (+8 short)	34.4%	1	74
	1999	52	19 (+7 short)	36.5%	1	80
	2000	33	16	48.5%	1	37
	2002	31	15	48.4%	1	48
	2003	63	18	28.6%	1	100
	2005	44	19	43.2%	1	31

Cloudflare, Inc. Exhibit 1003 Page 1216

	2006	55	16	29.1%	1	30
	2007	45	17	37.8%	1	28
	2008	35	14	40.0%	1	35

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
FMOODS/FORTE (IFIP)	2009	46	18	39.1%	1	30

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
FORTE (IFIP) (joined with PSTV in 1996) (joined with FMOODS in 2009)	1988	??	18	??.?%	1	118
	1989	??	28	??.?%	1	140
	1990	110	22	20.0%	1	256
	1991	70	29	41.4%	1	83
	1992	116	29	25.0%	1	201
	1993	93	29	31.2%	1	97
	1994	121	27	22.3%	1	???
	1995	80	30	37.5%	1	125
	2001	51	24	47.0%	1	122
	2002	54	22	40.7%	1	70
	2003	54 (+10 WIP)	24	44.4%	1	48
	2004	54	20	37.0%	1	67
	2005	89	33	37.1%	1	109
	2006	78	29	37.2%	1	76
	2007	67	22	32.8%	1	38
	2008	44	20	45.5%	1	70

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
FORTE/PSTV (IFIP)	1996	110	28	25.5%	1	140
	1997	115	29 (+4 short)	27.0%	1	???
	1998	85	26	30.6%	1	132
	1999	78	22 (+5 short)	28.2%	1	130
	2000	63	22	34.9%	1	95

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Global Internet	1999	117	28	23.9%	1	???
	2000	113	28	24.8%	1	???
	2001	74	18	24.3%	1	???
	2002	142	28	19.7%	1	???
	2003	???	??	??.?%	1	???
	2004	???	??	??.?%	1	???
	2005	??	22	??.?%	1	???
	2006	66	22	33.3%	1	47
	2007	42	15	35.7%	1	40
	2008	43	15	34.9%	1	???
	2009	36	12	33.3%	1	???

Cloudflare, Inc. Exhibit 1003 Page 1217

	2010	73	16	21.9%	1	40
	2011	35	14	40.0%	1	20
	2012	27	10	37.0%	1	22
	2013	41	13	31.8%	1	??
	2014	21	9	42.9%	1	20
	2015	29	9	31.0%	1	24
	2016	33	9	27.3%	1	42
	2017	25	9	36.0%	1	30+
	2021	44	18	40.9%	1	30+

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Globecom (IEEE)	2000	1020	357	35%	?	1548
	2001	1525+	718	47%	?	1403
	2002	1980	606	30.6%	?	1403
	2003	2,250	816	36.3%	???	1447
	2004	2,086	792	37.7%	7 symposia	???
	2005	1,638	770	47.7%	7 symposia	???
	2006	2,548	1,027	40.2%	14 symposia	???
	2007	2,576	1,018	39.5%	10 symposia	???
	2008	2,854	1,051	36.8%	11 symposia	???
	2009	3,200	1,104	34.8%	12 symposia	???
	2010	3,688	1,313	35.6%	11 symposia	???
	2011	2,923	1,070	36.6%	11 symposia	???
	2012	2,560	966	37.7%	11 symposia	???
	2013	2,272	841	37.0%	11 symposia	???
	2014	?,???	???	??.?%	?? symposia	???
	2015	2,614	915	35.0%	12 symposia	???
2019	2,560	973	38.0%	12 symposia	2500	

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Grid	2004 (part of SC)	173	40	23.1%	2	390
	2005 (part of SC)	170	32	18.8%	2	450
	2006 (co-located w/ Cluster)	188	38	20.2%	2	140
	2007 (co-located w/ Cluster)	169	37	21.8%	2	143
	2008 (co-located w/ Cluster)	176	35	19.8%	2	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
HiPC	2001	108	29	26.9%	???	220
	2002	145	57	39.3%	???	???
	2003	164	48	29.3%	5	???
	2004	214	48	22.4%	5	???

Cloudflare, Inc. Exhibit 1003 Page 1218

	2005	362	50	13.8%	5	~300
	2006	335	52	15.5%	6	???
	2007	253	52	20.6%	6	???
	2008	317	46	14.5%	6	???
	2009	320	35	10.9%	1	250
	2010	208	40	19.2%	1	280
	2011	206	40	19.4%	1	400
	2012	163	41	25.1%	1	425+
	2013	196	49	25.0%	1	450
	2014	212	49	23.1%	1	375
	2015	201	48	23.9%	1	???
	2016	160	40	25.0%	1	315
	2017	184	42	22.8%	1	433
	2018	151	33	21.9%	1	415
	2019	171	39	22.8%	1	501
	2020	127	33	226.0	1	604

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
HotPlanet (ACM)	2009	13	5	38.4%	1	22

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
HPCC	2008	455	90	19.8%	3	???
	2009	249	54	21.6%	3	???
	2010	304	58	19.1%	3	???
	2011	271	59	21.7%	3	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
HPDC	2003	125	25	20.0%	1	???
	2004	149	24	16.1%	1	???
	2005	133	24	18.0%	1	???
	2006	157	24	15.3%	1	270
	2007	100	20	20.0%	1	???
	2008	103	18	17.5%	1	105
	2009	68	20	29.4%	1	???
	2010	91	23	25.3%	1	???
	2011	170	22	12.9%	1	???
	2012	141	23	16.3%	1	160
	2013	131	20	15.3%	1	150
	2014	130	21 full	16.1%	1	125
	2015	116	19 full	16.4%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
HPSR	2000	194	64	33.0%	3	150
	2001	???	??	??.?%	?	???

Cloudflare, Inc. Exhibit 1003 Page 1219

	2002	110	63	57.2%	3	204
	2003	116	53	45.7%	3	120
	2004	???	??	??.?%	?	???
	2005	218	108	49.5%	3	167
	2006	114	61	53.5%	3	90
	2007	94	56	59.6%	3	80
	2008	??	??	??.?%	?	??
	2009	??	??	??.?%	?	??
	2010	73	34	46.6%	1	???
	2011	93	46	49.4%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
I3E (IFIP)	2015	80	32	40.0%	3	~55
	2016	136	52	38.2%	4	~60

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICASSP (IEEE)	1999	1490	867	58.2%	???	???
	2000	???	???	??.?%	???	???
	2001	1916	975	50.9%	???	???
	2002	1770	1007	56.9%	???	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICBC (IEEE)	2019	153	30 (+5 short, +41 posters, +9 demos)	19.6%	1	203
	2020	203	30 (+14 short, +32 posters, +7 demos)	21.0%	1	242

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICC (IEEE)	2001	1145	628	54.8%	2 (plus 8 symposia)	1390
	2002	1568	655	41.8%	11 symposia	???
	2003	1879	704	37.5%	15	1036
	2004	2946	864	29.0%	18	1501
	2005	2000	692	35.0%	8	???
	2006	2517	982	39.0%	9	???
	2007	~2740	1070	39.0%	9	???
	2008	3135	1102	35.2%	9	???
	2009	~3000	1046	34.9%	11	2020
	2010	2618	1035	39.5%	11	???
	2011	2838	1093	38.5%	12	???
	2012	2819	1043	37.0%	12	???
	2013	2422	948	39.1%	12	???
	2014	2608	990	38.0%	12	???
	2015	3342	1283	38.4%	~17	>1500
	2016	2469	960	38.9%	~13	???

Cloudflare, Inc. Exhibit 1003 Page 1220

	2017	3074	1172	38.1%	~12	???
	2018	2431	970	39.9%	~13	???
	2019	2590	~1010	~39.0%	~13	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IEEE/CIC ICCC	2013	370	151	40.8%	3	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICCCN	2006	237	66	27.8%	3	???
	2007	550	160	29.1%	4	???
	2008	399	104	26.1%	17	???
	2010	189	64	33.9%	8	~100
	2011	452	134	29.6%	12	315
	2012	266	82	30.8%	11	???
	2019	175	50	28.6%	13	~200

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICCPS (ACM/IEEE)	2010	70	20	28.6%	1	???
	2011	79	20	25.3%	1	???
	2012	41	14	34.1%	1	???
	2013	103	24	23.3%	1	107
	See Official Stats					

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICDCS (IEEE)	2001	217	69	31.8%	3	270
	2002	331	61	18.4%	3	?
	2003	406	72	17.7%	3	?
	2004	475	84	17.7%	3	?
	2005	543	75	13.8%	3	?
	2006	540	75	13.8%	3	?
	2007	528	71	13.5%	3	?
	2008	638	102	16.0%	3	?
	2009	455	74	16.3%	3	?
	2010	585	84	14.4%	3	?
	2011	565	87	15.4%	3	~200
	2012	515	71	13.8%	3	???
	2013	365	61	16.7%	3	???
	2014	500	66	13.2%	2	???
	2015	543	70	12.9%	2	???
	2016	398	68	17.1%	2	???
	2017	531	89	16.8%	6	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICDIM (IEEE)	2006	231	84	36.4%	3	135
	2007	219	83	37.9%	3	139

Cloudflare, Inc. Exhibit 1003 Page 1221

	2008	389	152	39.1%	5	182

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICIN	2013	45	22	48.9%	1	69
	2015	88	32	36.4%	1	105
	2016	70	17	24.3%	1	178
	2017	96	31	32.3%	1	110
	2018	85	24 (+13 short)	28.2%	1	125
	2019	67	21	31.3%	1	132
	2020	69	22 (+3 short)	31.8%	1	103

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICODP (IFIP)	1991	60	22	30.0%	1	90
	1993	60	22	30.0%	1	110
	1995	??	30	??.?%	1	150
	1997	??	30	??.?%	1	180

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICNC	2012	467	136	29.1%	4	???
	2013	527	133	25.2%	4	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICPADS (IEEE)	2009	305	91	29.8%	3	???
	2012	294	87	29.6%	3	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICNP (IEEE)	1996	96	31	32.3%	1	???
	1997	81	32	39.5%	1	???
	1998	108	36	33.3%	1	???
	1999	131	36	27.5%	1	???
	2000	109	30	27.5%	1	???
	2001	157	36	22.9%	1	???
	2002	217	32	14.7%	1	???
	2003	230	30	13.0%	1	128
	2004	213	33	15.5%	1	81
	2005	212	36	17.0%	1	130
	2006	232	33	14.2%	1	???
	2007	220	32	14.5%	1	???
	2008	202	33	16.3%	1	???
	2009	198	36	18.2%	1	???
	2010	170	31	18.2%	1	???
	2011	189	31	16.4%	1	???
	2012	131	30	22.9%	1	???
	2013	252	46	18.3%	1	157
	2014	160 (79 short)	32 (15 short)	20.0% (19.0% short)	1	???

Cloudflare, Inc. Exhibit 1003 Page 1222

	2015	??? (?? short)	?? (?? short)	??.?% (??.?% short)	1	???
	2016	230	46 (21 short)	20.0%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICPADS (IEEE)	2009	305	91	29.8%	3	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICPP	2001	139	65	46.7%	1	???
	2002	188	67	35.6%	3	???
	2003	192	69	35.9%	3	???
	2004	190	65	34.2%	3	???
	2005	241	69	28.6%	3	???
	2006	200	64	32.0%	3	???
	2007	???	??	??.?%	3	???
	2008	263	81	30.8%	3	???
	2009	???	??	??.?%	3	???
	2010	???	??	??.?%	3	???
	2011	363	81	22.3%	4	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICQT	2001	19	10	52.6%	1	40
	2002	20	10	50.0%	1	81 (QofIS & ICQT)
	2003	27	8	29.6%	1	72 (ICQT & NGC)
	2004	22	8	36.4%	1	103 (QofIS, ICQT and WQoSR)
	2006	27	9	30.0%	1	28
	2009	26	9	34.6%	1	35
	2011	18	6	33.3%	1	35
	2013	15	6	40.0%	1	15

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICS (ACM)	1994	114	45	39.5%	1	???
	1995	120	49	40.8%	1	???
	1996	116	50	43.1%	1	???
	1997	135	45	33.3%	1	???
	1998	???	??	??.?%	1	???
	1999	180	57	31.7%	1	???
	2000	122	33	27.0%	1	???
	2001	133	45	33.8%	1	???
	2002	144	31	21.5%	1	???
	2003	171	36	21.1%	1	???
	2004	162	35	21.6%	1	????
	2005	152	42	27.6%	1	????

Cloudflare, Inc. Exhibit 1003 Page 1223

	2006	141	37	26.2%	1	????
	2007	123	29	23.6%	1	????
	2008	140	37	26.4%	1	????
	2009	191	47	24.6%	1	????
	2010	180	32	17.8%	1	????
	2011	161	35	21.7%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ICUFN	2009	127	39	30.7%	2	120
	2014	307	124	40.4%	3	???
	2015	446	211	47.3%	4	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IM (IFIP/IEEE)	1989	93	46	49.5%	3	???
	1991	110	50	45.5%	3	???
	1993	100	50	50.0%	3	550
	1995	109	50	45.9%	3	500
	1997	138	56	40.6%	3	???
	1999	161	61	37.8%	3	570
	2001	150	51	34.0%	3	283
	2003	149	41	27.5%	3	154
	2005	226	53	23.5%	3	320
	2007	187	51	27.3%	3	316
	2009	125	41	32.8%	3	224
	2011	160	47	29.4%	2	210
	2013	196	53	27.0%	3	367
	2015	206	56	27.2%	3	350
	2017	154	44	28.6%	3	303
	2019	157	39	24.8%	3	172

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IMC (ACM/USENIX) (originally IMW)	2001	53	14	26.4%	1	60
	2002	93	39	41.9%	1	103
	2003	109	32	29.4%	1	114
	2004	157	39	24.8%	1	109
	2005	148	36	24.3%	1	???
	2006	155	34	21.9%	1	???
	2007	86 (long papers only)	21 (long papers only)	24.4%	1	???
	2008	185	32	17.3%	1	???
	2009	183	41	22.4%	1	???
	2010	110 (+101 short)	24 (+23 short)	22.3%	1	~120
	2011	220 (total)	42 (total)	19.1%	1	~120

Cloudflare, Inc. Exhibit 1003 Page 1224

	2012	183	45	24.6%	1	???
	2013	178	42	23.6%	1	???
	2014	188	43	22.9%	1	150
	2015	169	44	26.0%	1	???
	2016	182	46	25.3%	1	???
	2017	188	42	22.3%	1	???
	2018	174	43	24.7%	1	???
	2019	197	39	19.8%	1	232

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Infocom (IEEE)	1995	470	156	33.2%	4	???
	1996	590	176	29.8%	4	???
	1997	524	167	31.9%	4	???
	1998	800	172	21.5%	4	???
	1999	600	184	30.7%	5	???
	2000	735	192	26.1%	5	550
	2001	830	192	23.1%	5	???
	2002	938	192	20.5%	5	???
	2003	1078	224	20.8%	5	???
	2004	1420	261	18.4%	6	???
	2005	1419	244	17.2%	6	???
	2006	>1400	252	~18%	6	???
	2007	>1400	252	~18%	6	???
	2008	>1160	236	20.3%	6	???
	2009	1435	282	19.7%	6	???
	2010	1575	276	17.5%	6	???
	2011	1823	291	16.0%	6	???
	2012	1547	278	18.0%	6	???
	2013	1613	280	17.4%	6	???
	2014	1650	319	19.3%	6	???
	2015	1640	316	19.3%	6	???
	2016	1644	300	18.2%	6	???
	2017	1395	292	20.9%	8	???
	2018	1606	309	19.2%	8	???
	2019	1464	288	19.7%	8	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IPCCC (IEEE)	2004	???	??	28.0%	3	???
	2005	115	37	32.2%	3	?
	2006	???	??	??.?%	1	?
	2007	119	44	37.0%	3	?
	2008	141	42	29.8%	3	?

Cloudflare, Inc. Exhibit 1003 Page 1225

	2009	145	43	29.7%	2	~60
	2010	127	36	28.4%	1	~50
	2011	129	36	27.9%	1	???
	2012	115	32	27.8%	1	~100
	2013	129	35	27.1%	1	???
	2015	298	81	27.2%	2	???
	2018	228	60	28.9%	2	???
	2019	???	??	29.2%	2	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IPDPS (IEEE)	1996	353	126	35.7%	3	400
	1997	298	112	37.6%	3	429
	1998	324	118	36.4%	3	456
	1999	260	113	43.5%	3	460
	2000	303	107	35.3%	3	402
	2001	276	100	36.2%	3	465
	2002	258	98	38.0%	3	445
	2003	407	119	29.2%	4	550
	2004	447	142	31.8%	4	617
	2005	343	115	33.5%	4	588
	2006	531	125	23.5%	4	542
	2007	419	109	26.0%	4	656
	2008	410	105	25.6%	4	657
	2009	440	101	23.0%	4	535
	2010	527	127	24.1%	4	577
	2011	571	112	19.6%	4	600
	2012	568	118	20.8%	4	621
	2013	490	108	22.0%	4	645
	2014	541	114	21.1%	4	603
	2015	496	109	22.0%	4	484
	2016	496	114	23.0%	4	658
	2017	508	116	22.8%	4	634
	2018	461	113	24.4%	4	495
	2019	372	103	27.7%	4	373
	2020	446	110	24.7%	4	699

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IPOM (IEEE)	2003	55	19	34.5%	1	45
	2004	130	44	33.9%	2	100
	2005	46	21	45.6%	1	50
	2006	45	18	40.0%	1	86
	2007	40	16	40.0%	1	69

Cloudflare, Inc. Exhibit 1003 Page 1226

	2008	30	12	40.0%	1	60
	2009	35	12	34.3%	1	123
Merged with Manweek, DSOM, and MMNS into CNSM.						

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IPS-MoMe	2003	31	26	83.9%	1	32
	2004	37	25	70.3%	1	50
	2005	42	27	64.3%	1	70
	2006	34	18	52.9%	1	63

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IPSN	2004	145	25 (+25 posters)	17.2%	1	???
	2005	???	?? (+?? posters)	??.?%	1	???
	2006	165	25 (+17 posters)	15.2%	1	???
	2007	170	26 (+12 posters)	15.3%	1	???
	2008	126	30	23.8%	1	???
	2009	117	21	17.9%	1	???
	2010	117	20	17.1%	1	???
	2011	122	27	22.1%	1	???
	2012	147	22	15.0%	1	???
	2013	115	24	20.9%	1	???
	2014	111	20	18.0%	1	???
	2015	108	27	25.0%	1	???
	2016	117	21 (+2 short)	17.9%	1	???
	2017	104	19	18.3%	1	???
	2018	82	22 (+2 short)	26.8%	1	???
	2019	91	25	27.5%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IPTPS	2002	99	31	31.3%	1	55
	2003	166	27	16.3%	1	??
	2004	145	27	18.6%	1	66
	2005	124	24	19.4%	1	???
	2006	90	24	26.7%	1	???
	2007	90	21	23.3%	1	49
	2008	68	21	30.1%	1	48
	2009	59	12	20.0%	1	40
	2010	64	13	20.3%	1	34

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ISCC (IEEE)	1995	???	95	??.?%	?	???
	1997	???	106	??.?%	?	???
	1998	???	130	??.?%	?	???
	1999	119	66	55.5%	3	88

Cloudflare, Inc. Exhibit 1003 Page 1227

	2000	224	123	54.9%	3	???
	2001	206	113	54.9%	3	136
	2002	280	154	55.0%	3	177
	2003	455	228	50.1%	4	236
	2004	475	187	39.4%	3	195
	2005	400	147	36.8%	4	195
	2006	348	161	46.3%	4	186
	2007	415	167	40.3%	4	205
	2008	390	180	46.2%	4	???
	2009	???	???	35.9%	?	???
	2010	323	130	40.2%	4	???
	2011	389	131	33.7%	4	???
	2012	316	150	47.5%	4	???
	2013	302	147	48.7%	4	???
	2014	338	157	46.4%	4	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ISPDC	2006	84	45	53.6%	1	35-40

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ITC	2009	174	58	33.3%	2	139
	2010	97	35	36.1%	1	120
	2011	115	38	33.0%	1	131
	2012	74	24	32.4%	1	129
	2013	91	32	35.1%	1	64
	2014	72	27	37.5%	1	103
	2015	77	26	33.7%	1	83
	2016	108	37 (+6 demos)	34.3%	2	117
	2017	91 (+3 demos)	28 (+2 demos)	30.8%	1	71
	2018	46 full (+9 short, +13 special)	18 full (+5 short, +11 special)	39.1%	1	76

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IWAN	2002	53	20	37.8%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IWCMC (IEEE)	2006	550	270	49.1%	3	???
	2007	300	119	39.7%	3	???
	2008	???	???	??.?%	4	???
	2009	>600	285	~47.5%	5	???
	2010	???	???	??.?%	5	???
	2011	???	???	??.?%	5	???
	2012	???	???	??.?%	5	???

Cloudflare, Inc. Exhibit 1003 Page 1228

	2013	630	220	34.9	5	???
	2014	595	202	33.9	5	???
	2015	700	252	36.0	5	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IWPTS (1988-1996) (continued as IWTCS)	1998	??	15	??.?%	1	40
	1989	45	25	55.6%	1	70
	1990	??	??	??.?%	1	???
	1991	??	16	??.?%	1	???
	1992	37	19	51.4%	1	85
	1993	??	16	??.?%	1	???
	1994	??	15 (+3 short)	??.?%	1	???
	1995	??	??	??.?%	1	???
	1996	??	??	??.?%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IWTCS (1997-1999) (continued as Testcom)	1997	47	25	53.2%	1	130
	1998	43	18	41.9%	1	45
	1999	29	21	72.4%	1	107

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
IWQoS (IEEE)	1994(1st)	15	15	100.0%	1	45
	1994(2nd)	15	15	100.0%	1	40
	1995	15	15	100.0%	1	80
	1996	30	15	50.0%	1	90
	1997	106	24	22.6%	1	???
	1998	125	21	16.8%	1	160
	1999	120	21	17.5%	1	130
	2000	84	21	25.0%	1	130
	2001	143	30	21.0%	1	120
	2002	143	27	18.9%	1	60
	2003	???	??	??.?%	?	??
	2004	154	24 (plus 6 short)	19.5%	1	80
	2005	120	23 (plus 17 short)	19.2%	1	80
	2006	150	27 (plus 13 short)	18.0%	1	??
	2007	64	17	20.6%	1	50+
	2008	73	26	35.6%	1	60+
	2009	80	27	33.8%	1	???
	2010	133	33	24.8%	1	???
	2011	80	23	28.8%	1	???
	2012	110	24	21.8%	1	???
	2013	60	16 (plus 10 short)	26.7%	1	???

Cloudflare, Inc. Exhibit 1003 Page 1229

	2014	101	24	23.8%	1	80

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
LANC (ACM)	2003	30	7	23.3%	1	40
	2005	33	12	33.2%	1	45
	2007	37	11	32.4%	1	50
	2009	30	12	36.3%	1	40
	2011	27	9	33.3%	1	40
	2012	24	9	37.5%	1	??

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
LANMAN (IEEE)	1999	63	53	84.1%	1	??
	2001	??	??	??.?%	1	??
	2002	??	??	??.?%	1	??
	2004	109	47	43.1%	1	??
	2005	110	36	32.7%	1	??
	2007	60	30	50.0%	1	63
	2008	36	15	41.7%	1	40
	2010	38	23	60.5%	1	42
	2011	39	21	53.8%	1	41
	2013	32	10(+4 short)	31.3%	1	37
	2014	37	12(+4 short)	32.4%	1	30
	2015	47	18	38.3%	1	??

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
LANOMS (IEEE)	2009	43	12	27.9%	1	42
	2011	30	12	40.0%	1	~40

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
LCN (IEEE)	2000	128	70	54.7%	4	115
	2001	100	49	49.0%	3	80
	2002	140	58	41.0%	3	130
	2003	146	49	33.6%	3	138
	2004	142	42	30.4%	3	144
	2005	212	56	26.4%	3	152
	2006	172	60	34.9%	3	173
	2007	171	63	36.8%	3	154
	2008	164	64	39.0%	3	164
	2009	188	59	31.0%	3	219
	2010	169	51	30.2%	3	215
	2011	181	53	29.3%	3	237
	2012	174	52	29.8%	3	196
	2013	201	53	26.4%	3	205
	2014	143	42	29.4%	2	142

Cloudflare, Inc. Exhibit 1003 Page 1230

	2015	145	44	30.3%	3	174
	2016	231	58	25.1%	3	185
	2017	197	56	28.4%	3	169
	2018	161	45	28.0%	3	155
	2019	135	39	28.9%	3	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
MASCOTS (IEEE/ACM)	2003	???	??	~30.0%	1	???
	2004	172	67	39.0%	2	???
	2005	151	46 (+21 short)	30.5%	2	???
	2006	125	45	36.0%	2	???
	2007	187	62	33.2%	2	???
	2008	94	36	38.3%	1	108
	2009	162	54	33.3%	2	???
	2010	116	36	31.0%	1 & 2	65
	2011	120	42	35.0%	1 & 2	???
	2012	134	49	36.6%	2	76
	2013	???	??	??.?%	2	??
	2014	192	39	20.3%	2	~80

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
MASS (IEEE)	2006	197	49	24.9%	1	???
	2007	265	69	26.0%	3	???
	2008	250	59	23.6%	3	???
	2009	???	??	??.?%	3	???
	2010	181	51	27.5%	3	???
	2011	256	49	19.1%	3	???
	2012	166	50	30.1%	4	???
	2013	186	55 (+16 short)	29.6%	3	???
	2014	181	48	26.5%	2	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Med-Hoc-Net (IFIP)	2008	36	20	55.6%	1	25
	2009	17	9	52.9%	1	40
	2010	78	22	28.2%	1	51
	2011	30	12	40.0%	1	40
	2012	33	15	45.5%	1	30
	2013	56	20	35.7%	1	51
	2014	44	14	31.8%	1	36
	2015	33	18	54.5%	1	24
	2016	38	17	44.7%	1	22
	2017	41	15	36.6%	1	43

Cloudflare, Inc. Exhibit 1003 Page 1231

	2018	39	17	43.6%	1	30

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
MELT (ACM)	2008	38	15	39.5%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Middleware (IFIP/ACM)	1998	150	28	18.7%	1	160
	2000	70	21	30.0%	1	180
	2001	116	20 (+10 short)	17.2%	1	105
	2003	158	25 (+5 WIP)	15.8%	1	176
	2004	194	25	12.9%	1	~130
	2005	106	24	22.6%	1	250
	2006	122	22	18.0%	1	162
	2007	108	22	20.4%	1	???
	2008	119	21	17.6%	1	140
	2009	111	19	17.1%	1	125
	2010	116	19	16.4%	1	???
	2016	58	12	20.7%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
MMCN (SPIE/ACM)	2005	100	16 (plus 8 short)	24.0%	1	???
	2006	87	17	19.5%	1	???
	2007	??	??	??.?%	1	???
	2008	57	15 (plus 9 short)	26.3%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
MMNS (IFIP/IEEE)	1997	83	22	26.5%	1	80+
	1998	61	22	36.1%	1	60+
	2000	49	22	44.9%	1	40+
	2001	105	22	21.0%	1	???
	2002	76	27	35.5%	1	???
	2003	??	30	??.?%	1	???
	2004	84	26	31.0%	1	65
	2005	66	34	51.1%	1	???
	2006	61	18	29.5%	1	86
	2007	54	14	26.0%	1	65
	2008	46	16	32.6%	1	65
	2009	37	13	35.1%	1	123
Merged with Manweek, DSOM, and IPOM into CNSM.						

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Mobiarch (IEEE/ACM)	2006	33	11	33.3%	1	33
	2007	46	11	23.9%	1	90

Cloudflare, Inc. Exhibit 1003 Page 1232

	2008	44	15	34.1%	1	???
	2009	19	7	3681%	1	22

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Mobicom (ACM)	1995	79	20	25.3%	1	???
	1996	90	18	20.0%	1	???
	1997	101	26	25.7%	1	???
	1998	147	27	18.4%	1	???
	1999	170	23	13.5%	1	???
	2000	226	28	12.3%	1	???
	2001	281	30	10.7%	1	???
	2002	364	26	7.1%	1	???
	2003	281	27	9.6%	1	???
	2004	327	26	8.0%	1	???
	2005	224	23	10.3%	1	???
	2006	298	35	11.7%	1	???
	2007	233	26	11.1%	1	???
	2008	264	31	11.7%	1	???
	2009	282	30	10.6%	1	???
	2010	233	33	14.2%	1	???
	2011	214	29	13.6%	1	???
	2012	212	32	15.1%	1	???
	2018	178	30	16.9%	1	???
	2019 (2nd rnd)	186	331	16.7%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
MobiHoc (ACM/IEEE)	2000	82	13	15.6%	1	165
	2001	144	24	16.7%	1	???
	2002	134	22	16.4%	1	???
	2003	189	29	15.3%	1	???
	2004	234	24	10.1%	1	200(?)
	2005	281	40	14.2%	1	???
	2006	318	31	9.75%	1	???
	2007	146	27	18.5%	1	???
	2008	300	44	14.7%	1.5	???
	2009	175	31	17.7%	1	???
	2010	157	6?	16.6%	1	???
	2011	127	25	19.7%	1	~100
	2012	120	24	20.0%	1	???
	2019	156	37	23.7%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
MobiOpp (ACM)	2007	37	14	37.9%	1	30

Cloudflare, Inc. Exhibit 1003 Page 1233

	2010	63	18	28.6%	1	76
	2012	44	9	20.5%	1	50

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Mobiquitous (ACM)	2006	107	19	17.8%	1	???
	2007	119	27	22.7%	1	???
	2008	???	??	??.?%	1	???
	2009	???	??	??.?%	1	???
	2010	???	??	??.?%	1	???
	2011	85	27	27.1%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Mobisys (ACM)	2007	105	22	20.9%	1	???
	2008	123	22	17.9%	1	???
	2009	128	26	20.3%	1	???
	2010	126	25	19.8%	1	???
	2011	141	25	17.7%	1	???
	2012	182	32	17.6%	1	???
	2020	175	34	19.4%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Mobiwac (ACM)	2008	50	18 (+11 posters)	36.0%	1	???
	2009	41	15 (+6 posters)	36.6%	1	???
	2010	??	?? (+? posters)	??.?%	1	???
	2011	58	15 (+6 posters)	25.8%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
MSN (IEEE)	2013	90	40	44.4%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
MSWiM	1998	20	8	40.0%	1	???
	1999	36	13	36.1%	1	???
	2000	44	16	36.4%	1	???
	2001	55	16	29.1%	1	???
	2002	81	16	19.7%	1	???
	2003	100	15	15.0%	1	???
	2004	121	21	17.4%	1	???
	2005	165	36	21.8%	1	???
	2006	160	39	24.4%	1	???
	2007	161	40	24.8%	1	???
	2008	188	39	20.7%	1	???
	2009	178	42	23.5%	1	???
	2010	132	43	32.6%	1	???
	2011	190	45 (+8 posters)	23.7%	1	???
	2012	161	42	26.1%	1	???

Cloudflare, Inc. Exhibit 1003 Page 1234

	2013	160	42	25.9%	1	???
	2014	146	32	21.9%	1	???
	2018	113	31	27.5%	1	???
	2019	103	30	29.1%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Multimedia (ACM)	1996	142	40	28.2%	2	???
	1997	162	36	22.2%	2	???
	1998	?	?	?	?	???
	1999	220	40	18.2%	?	???
	2000	214	36	16.8%	3	???
	2001	???	??	??.?%	3	???
	2002	???	??	??.?%	3	???
	2003	???	??	??.?%	3	???
	2004	331	55	16.6%	3	???
	2005	312	49	15.7%	3	335
	2006	292	48	16.4%	3	>400
	2007	298	57	19.1%	3	???
	2008	280	56	20.0%	3	???
	2009	305	50	16.4%	4	???
	2010	357	61	17.1%	3	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NEW2AN	2004	109	50	45.9%	1	62
	2006	137	49	35.8%	1	71
	2007	113	40	35.4%	1	59

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NIME (IEEE)	2004	33	10	30.3%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NDSS	2000	51	15	29.4%	1	???
	2001	66	16	24.2%	1	???
	2002	79	16	20.3%	1	???
	2003	82	17	20.7%	1	???
	2004	98	16	16.3%	1	???
	2005	124	16	12.9%	1	~100
	2006	127	17	13.4%	1	???
	2007	125	18	14.4%	1	???
	2008	121	20	16.5%	1	???
	2009	171	20	11.7%	1	???
	2010	156	24	15.4%	1	???
	2011	139	28	20.1%	1	???
	2012	???	??	??.?%	1	???

Cloudflare, Inc. Exhibit 1003 Page 1235

	2013	???	??	??.?%	1	???
	2014	???	??	??.?%	1	???
	2015	???	??	??.?%	1	???
	2016	???	??	??.?%	1	???
	2017	423	68	16.1%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NCA (IEEE)	2006	71	25	35.2%	3	???
	2011	96	27 (+30 short)	28.1%	2	???
	2014	76	24 (+25 short)	31.6%	1	40-50
	2016	105	29 (+36 short)	27.6%	1	~70

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NESEA (IEEE)	2010	23	10	43.5%	1	30
	2011	52	21	40.4%	1	30

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NetGames (ACM)	2002	19	12	63.2%	1	???
	2003	23	19	82.6%	1	??
	2004	32	15	46.9%	1	???
	2005	36	20	55.6%	1	???
	2006	30	20	66.7%	1	???
	2007	30	18	60.0%	1	???
	2008	31	15	48.4%	1	???
	2009	37	10	27.0%	1	46
	2010	27	9	33.3%	1	???
	2011	27	9	33.3%	1	40
	2012	44	9	20.4%	1	??

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NetSoft (IEEE)	2015	108	17 (+31 short)	15.7%	1	115
	2016	107	20 (+36 short)	18.7%	2	232
	2017	86	16 (+26 short)	18.6%	2	150
	2018	103	20 (+29 short)	19.2%	2	164
	2019	103	20 (+21 short)	19.2%	2	151
	2020	90	18 (+25 short)	20.0%	2	380

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Networking (IFIP)	2000	209	82	39.2%	3	???
	2002	314	82	26.1%	3	???
	2004	539	103	19.1%	3	???
	2005	430	106	24.7%	3	???
	2006	440	90	20.4%	3	???
	2007	440	99	22.5%	3	???
	2008	249	67 Full (+17 short)	26.9%	3	97

Cloudflare, Inc. Exhibit 1003 Page 1236

	2009	228	49 Full (+26 short)	21.5%	2	200
	2010	101	24	23.7%	1	85
	2011	294	64	21.8%	2	140
	2012	225	64	28.4%	2	125
	2013	191	50	26.2%	2	75
	2014	220	55	25.0%	2	90
	2015	202	48	23.8%	2	92
	2016	200	57	28.5%	2	~95
	2017	150	43	28.6%	1	120
	2018	225	55 (+8 short)	24.4%	2	118
	2019	112	32	28.5%	1	60
	2020	235	55 (+25 short)	23.4%	3	~275

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NGC	1999	50	15	30.0%	1	110
	2000	49	12	24.5%	1	120
	2001	40	14	35.0%	1	???
	2002	48	19	39.6%	1	72
	2003	51	17	33.3%	1	72 (ICQT & NGC)

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NGMAST	2007	91	44	48.3%	2	72
	2008	196	75	38.4%	3	103
	2009	217	75	34.5%	3	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NOF	2011	51	24	47.1%	1	???
	2012	82	29	35.4%	1	87
	2013	66	21	35.4%	1	55
	2014	59	21	35.6%	1	47
	2015	65	22	33.8%	1	35
	2016	58	20	34.5%	1	59
	2017	54	22 (+13 demo)	40.7%	1	61
	2018	36	11 (+5 short)	30.6%	1	27
	2019	50	13 (+7 short)	26.0%	1	40
	2020	53	15 (+9 short)	28.3%	1	95

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NOMS (IFIP/IEEE)	2000	205	61	29.7%	3	485
	2002	173	57	32.9%	3	264
	2004	224	60	26.8%	3	425
	2006	186	53	28.5%	3	331
	2008	220	64	29.1%	3	350

Cloudflare, Inc. Exhibit 1003 Page 1237

	2010	201	54	26.9%	3	369
	2012	210 (+31 short)	55	26.2%	3	260
	2014	152	44	28.9%	3	315
	2016	213	54	25.3%	3	155
	2018	189	56	29.6%	3	289
	2020	243	72 (+40 short and +18 posters)	29.6%	3 to 5	297

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NOSSDAV (ACM)	1990	???	???	???	1	???
	1991	???	???	???	1	???
	1992	???	???	???	1	???
	1993	???	???	???	1	???
	1995	???	???	???	1	???
	1996	???	???	???	1	???
	1997	81	25	30.9%	1	???
	1998	???	???	???	1	???
	1999	???	???	???	1	???
	2000	???	???	???	1	???
	2001	60	20	33.3%	1	60
	2002	59	18	30.5%	1	50
	2003	60	18	30.0%	1	40
	2004	95	24	25.3%	1	60
	2005	88	33	37.5%	1	48
	2006	74	23	31.1%	1	??
	2007	52	18	34.6%	1	55
	2008	50	17	34.0%	1	62
	2009	51	21	41.1%	1	???
	2010	51	21	41.1%	1	~50
	2011	61	21	34.4%	1	???
	2012	47	17	36.2%	1	~40

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NPC (IFIP)	2004	338	46	13.6%	3	150+

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NSDI (USENIX)	2004	118	27	22.9%	1	???
	2005	112	25	22.3%	1	???
	2006	110	28	25.4%	1	???
	2007	113	27	23.9%	1	???
	2008	175	30	17.7%	1	???
	2009	163	32	19.8%	1	???
	2010	175	29	16.6%	1	???

Cloudflare, Inc. Exhibit 1003 Page 1238

	2011	157	27	17.2%	1	???
	2012	169	30	17.8%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
OPENARCH (IEEE)	2000	42	13	31.0%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
NTMS (IFIP)	2014	136	52	38.2%	3	~100
	2015	111	38	34.2%	3	~90
	2016	107	38	35.5%	3	~70
	2018	110	38	34.5%	3	~85

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
OPENARCH (IEEE)	2000	42	13	31.0%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
OOPSLA (ACM)	2001	145	27	18.6%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
OSDI (USENIX)	1994	178	21	11.8%	1	???
	1996	110	19	17.3%	1	???
	1998	???	??	??.?%	1	???
	1999	96	20	20.8%	1	???
	2000	111	24	21.6%	1	???
	2002	???	??	??.?%	1	???
	2004	193	27	14.0%	1	???
	2006	150	27	18.0%	1	???
	2008	193	26	13.5%	1	???
	2009	???	??	??.?%	1	???
	2010	199	32	16.1%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
P2P (IEEE)	2001	21	4	19.1%	1	69
	2002	29	16	55.2%	1	66
	2003	70	19	27.1%	1	77
	2004	130	30	23.1%	1	79
	2005	136	26	19.1%	1	57
	2006	153	29	19.0%	1	80
	2007	???	??	??.?%	1	???
	2008	139	29	20.9%	1	???
	2009	126 (plus 23 short)	25 (plus 8 short)	19.8%	1	???
	2010	121 (plus 22 short)	27 (plus 7 short)	22.3%	1	???
	2011	97 (plus 26 short)	29 (plus 8 short)	29.9%	1	???
	2012	100 (inc short)	20 (plus 4 short)	24.0%	1	???
	2013	94	18	19.2%	1	~40

Cloudflare, Inc. Exhibit 1003 Page 1239

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
PAM	2004	174	30	17.2%	1	???
	2005	??	??	??.?%	1	???
	2006	??	??	??.?%	1	???
	2007	80	21	26,3%	1	???
	2008	71	23	32.4%	1	???
	2009	77	22	28.6%	1	???
	2010	78	23	29.5%	1	~80-100
	2011	56	24	42.9%	1	???
	2012	82	25	30.5%	1	???
	2013	74	24	32.4%	1	60
	2014	??	??	??.?%	1	??
	2015	??	??	??.?%	1	??
	2016	93	30	32.4%	1	??
	2017	87	20	23.0%	1	??
	2018	49	20	40.8%	1	??
	2019	75	20	26.7%	1	~53
	2020	65	19	29.92	1	~60

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
PerCom (IEEE)	2005	233	38	16.7%	2	???
	2006	280	23	8.21%	1	???
	2007	208	20	9.61%	1	???
	2008	???	??	?.??%	1	???
	2009	199	26	13.1%	1	???
	2010	227	26 (+2 short)	11.5%	1 (+workshops)	???
	2012	>150	16 (+12 short)	about 19%	1 (+workshops)	???
	2013	174	19 (+8 short)	15.5%	1 (+workshops)	240
	2014	175	25	14.3%	1 (+workshops)	???
	2015	196	29	14.7%	1 (+workshops)	???
	2016	168	20 (+5 concise)	14.9%	1 (+workshops)	???
	2017	194	28 (+4 concise)	16.5%	1 (+workshops)	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Performance (IFIP)	2002	129	30	23.3%	1	???
	2003	Not held.				
	2004	Held jointly with Sigmetrics 2004			1	???
	2005	132	29	22.0%	1	???
	2006	Held jointly with Sigmetrics 2006			1	???
	2007	103	24	23.0%	1	???

Cloudflare, Inc. Exhibit 1003 Page 1240

	2008	Not held.				
	2009	Held jointly with Sigmetrics 2009			1	???
	2010	78	20	25.6%	1	???
	2011	99	20	20.2%	1	???
	See Sigmetrics 2012					
	2013	65	17	26.2%	1	???
	2014	77	21	27.3%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
PEMWN	2012	23	16	69.6%	1	~30
	2013	24	13	54.2%	1	~40
	2014	38	20	52.6%	1	~50
	2015	41	20	48.8%	1	~60
	2016	46	16	34.8%	1	56
	2017	53	19	35.8%	1	55
	2018	30	12	40.0%	1	54
	2019	29	11	37.9%	1	~55

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
PIMRC (IEEE)	2006	956	416 (plus 149 posters)	43.5%	?	???
	2007	1200	518 (plus 180 posters)	43.2%	?	???
	2008	????	???	??.?%	?	???
	2009	???	652 (including posters)	??.?%	7	~1000
	2018	772	349 (+38 posters)	45.2%	7	537

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
PSTV (IFIP) (joined with FORTE in 1996)	1981	24	24	100.0%	1	30
	1982	40	40	100.0%	1	41
	1983	42	42	100.0%	1	???
	1984	41	41	100.0%	1	???
	1985	52	34	65.4%	1	62
	1986	48	34	70.8%	1	???
	1987	59	26	44.1%	1	162
	1988	75	28	37.3%	1	???
	1989	94	26	27.7%	1	191
	1990	67	23	34.3%	1	187
	1991	53	21	39.6%	1	187
	1992	89	22	24.7%	1	95
	1993	60	20	33.3%	1	95
	1994	51	18 (+9 short)	44.1%	1	???
	1995	75	25	33.3%	1	92

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
PVM/MPI	2005	110	56	50.9%	3	78

Cloudflare, Inc. Exhibit 1003 Page 1241

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
PWC (IFIP)	2005	209	57	27.2%	1	~120
	2006	174	49	28.2%	1	???
	2007	108	53	49.1%	1	79

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Q2SWinet (ACM)	2005	110	21	32.7%	1	???
	2006	60	18	30.0%	1	???
	2007	54	18	33.3%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
QofIS	2000	78	28	35.8%	1	160
	2001	95	23	24.2%	1	75
	2002	50	20	38.4%	1	81 (QofIS & ICQT)
	2003	76	22	28.9%	1	50
	2004	91	22	24.2%	1	103 (QofIS, ICQT and WQoSR)

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
RTAS (IEEE)	1996	86	27	31.4%	1	???
	1997	67	18	26.9%	1	???
	1998	94	26	27.7%	1	???
	1999	86	20	23.3%	1	???
	2000	84	26	31.0%	1	???
	2001	64	20	31.3%	1	???
	2002	89	27	30.3%	1	68
	2003	50	23	46.0%	1	82
	2004	205	62	30.2%	1	155
	2005	158	53	33.5%	1	150
	2006	128	38	29.7%	1	120
	2007	109	31	28.4%	1	110
	2008	140	35	25.0%	1	???
	2009	125	32	25.6%	1	???
	2010	149	33	22.1%	1	???
	2011	139	29	20.9%	1	???
	2012	127	30	23.6%	1	???
	2013	96	28	29.2%	1	94
	See Official Stats					

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
RTSS (IEEE)	1980	??	22	??.?%	1	???
	1981	??	19	??.?%	1	???
	1982	??	22	??.?%	1	???

Cloudflare, Inc. Exhibit 1003 Page 1242

1983	??	29	??.?%	1	???
1984	??	29	??.?%	1	???
1985	??	20	??.?%	1	???
1986	68	30	44.1%	1	???
1987	59	30	50.8%	1	???
1988	77	25	32.5%	1	???
1989	87	36	41.4%	1	???
1990	??	33	??.?%	1	???
1991	??	29	??.?%	1	???
1992	120	34	28.3%	1	???
1993	138	34	24.6%	1	???
1994	107	32	29.9%	1	???
1995	104	34	32.7%	1	???
1996	??	31	??.?%	1	???
1997	99	30	30.3%	1	???
1998	182	45	24.7%	1	???
1999	146	20	13.7%	1	???
2000	103	28	27.2%	1	???
2001	112	29	25.9%	1	???
2002	106	30	28.3%	1	110
2003	145	24	16.6%	1	130
2004	187	43	22.5%	1	166
2005	176	37	21.0%	1	123
2006	173	42	24.3%	1	120
2007	171	44	25.7%	1	???
2008	187	44	23.5%	1	152
2009	197	44	22.3%	1	170
2010	142	36	25.4%	1	150
2011	158	33	20.9%	1	165
2012	157	35	22.3%	1	155
2013	160	36	22.5%	1	172

See Official Stats

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
SAINT (IEEE)	2001	135	25	18.5%	3	167
	2002	70	26	37.1%	3	211
	2003	149	43	28.9%	3	218
	2004	105	29	27.7%	3	323
	2005	179	56	31.3%	3	252
	2006	99	41	41.4%	1	???
	2007	64	18	28.1%	1	???

Cloudflare, Inc. Exhibit 1003 Page 1243

	2008	67	18	26.9%	1	???
	2009	53	17	32.1%	1	???
	2010	51	14	27.5%	1	???
	2011	66	21	31.8%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
SASO (IEEE)	2007	120	25	20.8%	1	???
	2008	147	42	28.6%	1	???
	2009	74	27	36.5%	1	???
	2010	89	25	28.1%	1	???
	2011	97	21	21.6%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
SAWN (IEEE)	2001	121	30	24.8%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
SBAC-PAD	1999	90	43	47.7%	1	???
	2000	70	41	58.5%	1	???
	2001	--	32	--	1	???
	2002	74	25	33.7%	1	???
	2003	103	31	30.0%	1	???
	2004	93	29	34.9%	1	???
	2005	83	29	34.9%	1	???
	2006	71	23	32.4%	1	???
	2007	107	32	29.9%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
SCNS	2016	70	12	17.1%	1	41

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
SECON (IEEE)	2004	370	67	18.1%	3	???
	2005	203	55	27.2%	3	114
	2006	243	63	25.9%	3	134
	2007	296	63	21.3%	3	102
	2008	310	64	20.6%	3	131
	2009	437	96	22.0%	3	135
	2010	290	62	21.4%	3	???
	2011	241	66	27.4%	3	???
	2012	219	68	31.1%	3	???
	2013	280	51	18.2%	3	???
	2014	234	67	28.6%	3	???
	2015	194	55	28.3%	3	???
	2016	213	56	26.3%	2	???
	2017	170	45	26.5%	2	???
	2018	211	49	23.2%	2	???

Cloudflare, Inc. Exhibit 1003 Page 1244

	2019	189	54	28.6%	2	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Security (USENIX)	2001	83	24	28.9%	1	???
	2002	???	???	??.?%	?	???
	2003	???	???	??.?%	?	???
	2004	184	22	12.0%	1	???
	2005	148	22	14.9%	1	???
	2006	???	???	??.?%	?	???
	2007	???	???	??.?%	?	???
	2008	170	27	15.9%	1	???
	2009	176	26	14.8%	1	~400
	2010	202	30	14.9%	1	~800
	2011	204	35	17.2%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Security & Privacy	1999	61	15	24.6%	1	???
	2000	137	18	13.1%	1	???
	2003	131	19	14.5%	1	???
	2004	192	19	9.9%	1	200
	2005	192	17	8.9%	1	200
	2006	251	32	12.8%	1	???
	2007	???	??	??.?%	1	???
	2008	249	28	11.2%	1	???
	2009	254	28	11.0%	1	???
	2010	267	31	11.6%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
SenSys (ACM)	2003	137	24	17.5%	1	???
	2004	145	21	14.5%	1	???
	2005	118	21	17.8%	1	???
	2006	122	24	19.7%	1	222
	2007	149	25	16.8%	1	???
	2008	153	25	16.3%	1	???
	2009	119	21	17.6%	1	???
	2010	145	25	17.6%	1	???
	2011	123	24	19.5%	1	???
	2017	151	26	17.2%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Sigcomm (ACM)	1983	50	34	68.0%	1	???
	1984	70	33	47.1%	2	???
	1985	???	???	???	1	???
	1986	125	45	36.0%	2	???

Cloudflare, Inc. Exhibit 1003 Page 1245

	1987	94	42	44.7%	2	???
	1988	116	32	27.6%	1	???
	1989	94	29	30.9%	1	???
	1990	102	31	30.4%	1	???
	1991	128	28	21.9%	1	???
	1992	126	27	22.0%	1	???
	1993	149	28	18.8%	1	???
	1994	94	29	30.9%	1	???
	1995	143	30	21.0%	1	400
	1996	162	27	16.7%	1	???
	1997	213	24	11.3%	1	???
	1998	226	27	12.0%	1	???
	1999	190	24	12.6%	1	???
	2000	238	26	10.9%	1	407
	2001	203	23	11.3%	1	550
	2002	300	25	8.33%	1	???
	2003	319	33	10.3%	1	???
	2004	340	31	9.11%	1	???
	2005	255	27	10.6%	1	???
	2006	298	37	12.4%	1	???
	2007	258	35	13.6%	1	???
	2008	288	35	12.2%	1	619
	2009	270	27	10.0%	1	???
	2010	276	33	12.0%	1	~800
	2011	223	32	14.3%	1	???
	2012	235	31	13.2%	1	606
	2013	246	38	15.4%	1	???
	2014	242	45	18.6%	1	???
	2015	242	40	16.5%	1	???
	2016	231	39	16.9%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Sigmetrics (ACM)	1989	136	22	16.1%	1	???
	1990	102	23	22.3%	1	???
	1991	???	20	??.?%	1	???
	1992	108	20	18.5%	1	???
	1993	???	22	??.?%	1	???
	1994	110	25	22.7%	1	???
	1995	123	27	21.9%	1	???
	1996	110	24	21.8%	1	???
	1997	130	25	19.2%	1	???

Cloudflare, Inc. Exhibit 1003 Page 1246

	1998	136	25	18.4%	1	???
	1999	92	18	19.6%	1	???
	2000	165	28	17.0%	1	???
	2001	233	30	12.9%	1	???
	2002	170	23	13.5%	1	???
	2003	222	26	11.7%	1	???
	2004	265	33	12.4%	1	???
	2005	237	31	13.1%	1	152
	2006	217	30	13.8%	1	140
	2007	170	29	17.1%	1	???
	2008	201	36	17.9%	1	100+
	2009	180	27	15.0%	1	???
	2010	184	29	15.6%	1	???
	2011	177	29	14.7%	1	???
	2012	203	31	15.3%	1	???
	2013	196	26	13.3	1	???
	2014	237	40	16.9%	1	???
	2015	239	32	13.4%	1	???
	2016	208	28	13.5%	1	???
	2017	203	27	13.3%	1	???
	2018	270	54	20.0%	1	???
	2019	317	50	15.8%	2	???
	2020	286	44	15.4%	2	N/A
	2021	84 (summer only)	14	16.7%	2	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
SmartGridComm (IEEE)	2010	255	102	40.0%	3	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
SOSP (ACM)	1995	84	22	26.2%	1	???
	1996	???	???	???	???	???
	1997	120	23	19.2%	1	???
	1999	90	19	21.1%	1	???
	2001	85	17	20.0%	1	???
	2003	128	22	17.2%	1	???
	2005	155	20	12.9%	1	???
	2007	131	25	19.1%	1	471
	2009	140	23	16.4%	1	???
	2010	???	??	??.?%	1	???
	2011	157	28	17.8%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
SOSR (ACM)	2015	137	27	19.7%	1	???

Cloudflare, Inc. Exhibit 1003 Page 1247

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Supercomputing (ACM/IEEE)	2002	230	67	29.1%	2.5	???
	2003	207	60	29.0%	2.5	7000
	2004	192	59	30.7%	2.5	????
	2005	260	63	24.2%	2.5	????
	2006	239	54	22.6%	2.5	9700
	2007	???	??	??.?%	?	????
	2008	???	??	??.?%	?	????
	2009	???	??	??.?%	?	????
	2010	???	??	??.?%	?	????
	2011	352	74	21.0%	2.5	???
	2012	472	100	21.2%	2.5	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
SUSTAINIT	2012	30	9	30.0%	1	40
	2013	32	10	31.3%	1	52
	2015	28	12(+3 short)	42.9%	1	44
	2017	30	6(+6 short)	20.0%	1	50

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
Students' Tech Symp (IEEE)	2011	196	44	22.4%	7	???
	2014	410	96	23.4%	8	???
	2016	283	62	21.9%	8	120

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WCITD (IFIP)	2006	20	10	50.0%	1	50
	2008	44	18	40.0%	1	43

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
(previously IWTCS) Testcom (2000-2006) (continued Testcom/Fates)	2000	33	19	57.6%	1	126
	2002	54	21 (+5 short)	38.9%	1	138
	2003	53	19 (+5 short)	35.8%	1	74
	2004	39	14 (+8 short)	35.9%	1	60
	2005	52	24	41.3%	1	???
	2006	48	23	47.9%	1	55

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
(previously Testcom) Testcom/Fates (2007-current)	2007	80	24	30.0%	1	70
	2008	42	18	42.9%	1	87
	2009	37	13	35.1%	1	50

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
TMA (workshop)	2009	33	15	45.4%	1	~70
	2010	34	14	41.2%	1	~70

Cloudflare, Inc. Exhibit 1003 Page 1248

	2011	38	10 (+4 short)	26.3%	1	~80
	2012	29	10 (+8 short)	34.5%	1	~80
	2014	30	11	30.6%	1	50
	2015	54	16	29.2%	1	~80
	2016	47	16	34.0%	1	60

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
(combined from TMA workshop, TRAC and WNM) TMA	2017	54	19	35.2%	1	90
	2018	56	22	39.3%	1	120
	2019	65	24	36.9%	1	144
	2020	54	18	33.3%	1	115

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
UBICOMM	2010	630	189	30.0%	5	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
USENIX	1996	130	28	21.5%	1	???
	1997	74	23	31.1%	1	???
	1998	87	23	26.4%	1	1200
	1999	63	23	36.5%	1	???
	2000	90	27	30.0%	1	???
	2001	82	24	29.2%	1	???
	2002	107	25	23.4%	1	???
	2003	103	24	23.3%	2	???
	2004	164	21	12.8%	1	???
	2005	118	24	20.3%	1	???
	2006	153	21	13.7%	1	???
	2007	???	??	??.?%	?	???
	2008	176	34	19.3%	1	???
	2009	???	??	??.?%	?	???
	2010	147	24	16.3%	1	~250
	2011	???	??	??.?%	1	???
	2012	???	??	??.?%	1	???
	2013	233	33	14.1%	1	???
	2014	241	44	18.3%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
USITS (USENIX)	1997	85	23	27.1%	1	185
	1999	67	22	32.8%	1	150
	2001	70	19	27.1%	1	???
	2003	76	21	27.6%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
ACM VANET	2004	43	16	37.2%	1	???

Cloudflare, Inc. Exhibit 1003 Page 1249

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
VTC (IEEE)	2002 Fall	1146	450	39.3%	7	???
	2003 Fall	2400	720	30.0%	7	???
	2004 Spring	1200	600	50.0%	7	???
	2005 Spring	1600	650	40.6%	7	???
	2006 Spring	1364	481+134	45.1%	7	???
	2006 Fall	1126	409+208	54.8%	7	???
	2007 Spring	1440	437+210	44.9%	7	???
	2007 Fall	732	420 total	57.4%	7	???
	2008 Fall	904	420 total	46.5%	7	???
	2009 Spring	1140	405+231	55.8%	7	???
	2009 Fall	700+	416 total	59.4%	7	???
	2010 Fall	1051	510 total	48.5%	7	???
	2010 Spring	1164	423+173	51.2%	7	???
	2011 Spring	1164	423+173	51.2%	12	???
	2015 Spring	~900	421	~46.8%	15	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WiOpt (IEEE and IFIP)	2003	103	32	31.3%	1	???
	2004	186	43	23.1%	1	???
	2005	141	44	31.2%	1	???
	2006	???	??	??.?%	1	???
	2007	89	32	36.0%	1	???
	2008	144	46	31.9%	1	???
	2009	135	45	33.3%	1	???
	2010	112	37	33.0%	1	???
	2012	88	39	44.3%	1	???
	2013	131	58	44.3%	2	~100
	2014	167	48	28.7%	1	100
	2015	103	40	38.8%	1.5	~115
	2016	???	??	??.?%	1.5	???
	2017	135	44	32.6%	1.5	???
	2018	115	38	33.0%	1.5	~200
	2019	87	36	41.4%	1.5	~100
	2020	61	31	50.8%	1	~65

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WiSec (ACM)	2008	96	16	16.7%	1	83
	2009	107	14 (+14 short)	26.2%	1	85
	2010	99	9 (+12 short)	21.2%	1	~90

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WCITD (IFIP)	2006	20	10	50.0%	1	50

Cloudflare, Inc. Exhibit 1003 Page 1250

	2008	44	18	40.0%	1	43

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WCNC (IEEE)	2000	384	300	78.1%	?	???
	2002	568	167	29.4%	?	???
	2003	730+	377	52%	?	???
	2004	1087	470	43.2%	3	???
	2005	?	?	?	?	?
	2006	?	?	38.8%	?	?
	2007	1721	815	47.4%	15	???
	2008	1253	573	44.7%	10	???
	2009	1195	>500	~46.0%	11	???
	2010	1474	548	37.2%	11	???
	2011	????	???	48.1%	11	???
	2012	1407	620	44.1%	11	???
	2013	????	???	??.?%	11	???
	2014	????	???	??.?%	11	???
	2015	855	385	45.0%	4	???
	2016	964	467	48.4%	4	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WCW	1996	23	23	100.0%	1	22
	1997	35	35	100.0%	1	45
	1998	35	20	57.1%	1	124
	1999	38	19	50.0%	1	123
	2000	35	16	45.7%	1	110
	2001	61	20	32.8%	1	132

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WD (IFIP)	2008	240	105	43.8%	5	125
	2009	186	70	37.6%	4	92
	2010	230	75	32.6%	5	81
	2011	159	52	32.7%	5	90
	2012	158	55	34.8%	5	123
	2013	184	64	34.8%	6	130
	2014	105	36	34.3%	6	98
	2016	113	38	33.6%	7	79
	2017/a>	78	26	33.3%	5	91
	2018/a>	85	27	31.8%	5	85

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WICON	2008	128	48	37.5%	3	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WiMob (IEEE)	2008	336	77	22.9%	1	~120

Cloudflare, Inc. Exhibit 1003 Page 1251

	2009	190	63	33.2%	1	~120
	2010	228	66	28.9%	1	~120
	2011	160	57	35.6%	1	~120
	2012	294	70	24.5%	1	~200
	2013	219	61	27.9%	3	126
	2014	200	43	21.5%	3	~120
	2015	250	76	30.4%	3	126
	2016	220	59	26.8%	5	120
	2017	210	59	28.1%	5	117

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WMASH (ACM)	2003	29	10	34.5%	1	???
	2004	47	15	31.9%	1	48

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WMI (ACM)	2001	28	12	42.8%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WMNC (IFIP)	2008	81	37	45.6%	2	50
	2009	65	30	46.2%	3	45
	2010	78	32	41.0%	3	45
	2011	118	57	48.3%	3	61
	2012	44	16	36.4%	3	40
	2013	263	108	41.1%	3	92
	2014	82	27	32.9%	3	42
	2015	93	33	35.5%	3	48
	2016	48	22	45.8%	3	25
	2017	44	19	43.2%	3	24
	2018	34	16	47.1%	1	23

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WONS (IEEE and IFIP)	2004	77	25	32.5%	1	???
	2005	95	31	32.6%	1	???
	2006	56	24	42.9%	1	???
	2007	57	24	42.1%	1	61
	2008	49	18	36.7%	1	44
	2009	54	18	33.3%	1	43
	2010	47	15	31.9%	1	35
	2011	41	14	34.1%	1	38
	2012	55	15 full (+8 short)	27.3%	1	49
	2013	46	12 full (+8 short)	26.1%	1	47
	2014	48	11 full (+3	22.9%	1	42

Cloudflare, Inc. Exhibit 1003 Page 1252

			short)			
	2016	58 (+13 short)	17 (+9 short)	29.3%	1	~50
	2017	49 (+13 short)	22 (+9 short)	44.9%	1	~40
	2018	56 (+12 short)	22 (+10 short)	39.3%	1	~40
	2019	53 (+17 short)	23 (+6 short)	43.4%	1	~35

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WoWMoM (ACM)	2001	48	12	25.0%	1	???
WoWMoM (IEEE)	2002	46	15	32.6%	2	???
	2005	183	57	31.1%	2	???
	2006	144	48	33.3%	2	???
	2007	148	47	31.8%	2	141
	2008	181	32 (+10 short)	17.7%	2	???
	2009	160	36 (+13 short)	22.5%	2	???
	2010	173	35 (+12 short)	20.2%	2	???
	2011	173	32 (+8 short)	18.5%	2	???
	2012	145	32	22.1%	4	???
	2013	148	28 (+10 short)	18.9%	2	180
	2014	114	23	20.1%	2	???
	2015	242	48(+4 WIP)	21.5%	2	???
	2016	123	24(full), 9(short)	26.8%	2	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WORM (ACM)	2005	37	9	24.3%	1	???

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WPMC	2011	204	149	73.0%	3	244

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WQoSR	2004	28	8	28.6%	1	103 (QofIS, ICQT and WQoSR)

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WWIC	2002	18	8	44.4%	1	20
	2004	60	26	43.3%	1	40
	2005	117	34	29.1%	1	56
	2006	142	29	20.4%	1	51
	2007	257	32	12.5%	1	102
	2008	67	18	26.9%	1	63
	2009	39	13	33.3%	1	51
	2010	45	16	35.5%	1	64

Cloudflare, Inc. Exhibit 1003 Page 1253

	2011	50	26	52.0%	1	???
	2012	50	23 (+6 short)	46.0%	1	48
	2013	53	21	39.6%	1	48
	2014	63	24	38.1%	1	45
	2015	50	31	62.0%	1	47
	2016	54	27	50.0%	1	45
	2017	52	27	51.9%	1	40

Event Name	Year	Submitted	Accepted	% Accepted	Tracks	Attendees
WWW	1994	75	49	65.3%	3	385
	1994	?	?	??.?%	3	1,336
	1995	?	38	??.?%	3	1,531
	1995	189	59	31.2%	3	2,400
	1996	210	58	27.6%	3	2,351
	1997	276	62	22.5%	3	1,892
	1998	250	54	21.6%	3	1,208
	1999	304	48	15.8%	3	1,078
	2000	280	57	20.4%	3	1,439
	2001	391	78	19.9%	3	1,203
	2002	454	72	15.9%	3	913
	2003	602	77	12.8%	3	849
	2004	506	74	14.6%	3	1,003
	2005	550	77	14.0%	3	941
	2006	667	75	11.2%	3	1,250
	2007	750	111	14.8%	3	982
	2008	880	97	11.0%	3	???
	2009	823	105	12.8%	3	???
	2010	754	91	12.1%	3	???
	2011	658	81	12.3%	3	~900

Thanks to the following people who have contributed stats to this page: Dinesh Verma, Elizabeth Belding, Sugih Jamin, Christophe Diot, Brian Levine, Henning Schulzrinne, Luigi Rizzo, Jim Kurose, Sonia Fahmy, Ken Christensen, Ehab Al-Shaer, Moshe Sidi, Lars Wolf, Arun Iyengar, Fred Douglis, Samir Das, Nitin Vaidya, Adam Wolisz, Constantinos Dovrolis, Carl Staelin, Tommaso Bolognesi, Joseph Konstan, Azer Bestavros, Craig Partridge, Ahmed Helmy, Abhijit Sengupta, JJ Garcia-Luna, Mary Baker, Tristan Henderson, Marcus Brunner, Michael Smirnov, Olivier Festor, Chalermek Intanagonwiwat, German Goldszmidt, Jon Crowcroft, Bo Li, Ed Knightly, Marwan Krunz, Adarshpal Sethi, C.K. Toh, Rolf Stadler, Mostafa Ammar, Katia Obraczka, Stefan Savage, Jouni Karvo, Matthias Frank, Giuseppe Bianchi, Jose' Brustoloni, Thiemo Voigt, Guy Leduc, Christian Bettstetter, Nicolas Christin, Markus Hofmann, Jason Nieh, Cedric Westphal, Lukas Ruf, Burkhard Stiller, Murat Yuksel, Bernhard Thurm, Juergen Schoenwaelder, Peter Kropf, Stephane Genaud, Xiaoqiao Meng, Aiko Pras, Andrew Campbell, Xiao Qin, Sung Ju Lee, Mark Crovella, Sunghyun Choi, Renato Cerqueira, Jennifer Hou, William Gropp, Dan Katz, Gunnar Karlsson, Andrea Bianco, Olav Lysne, Jobin, James, Tevfik Kosar, Raj Buyya, Neil Spring, Barton Miller, Godmar Back, Nirupama Bulusu, Shaozhi Ye, Byung-Gon Chun, Jianying Zhou, Tevfik Kosar, Deep Medhi, Dmitri Loguinov, Erich Nahum, Qinghui Tang, Tom Hacker, Syed Khayam, Clay Shields, Ahsan Habib, Haijin Yan, Pouwelse, Grnmo Roy, Marten van Sinderen, Barrett Bryant, Soe-Tsyr Yuan, Hermes Senger, Sanjay Jha, David Bader, Hari Rangarajan, David Hausheer, Matthew Miller, Giovanni

Cloudflare, Inc. Exhibit 1003 Page 1254

Pau, Daniel Luo, Scott Pike, Wu-chun Feng, Yingpei Zeng, Sarang Dharmapurikar, Tim Brecht, Ben Zhao, Guofei Gu, Narasimha Datta, Hai Huang, Filippo Menczer, Brighten Godfrey, Lars Eggert, Lau Cheuk Lung, Amit Sahoo, Stephane Genaud, Soeren Sonntag, James Won-Ki Hong, Sungjin Ahn, Meng Zhang, Thomas Repantis, Anas Basalamah, Samee Khan, Lorant Farkas, Tae Hyun Kim, George Xylomenos, Katsuyuki Yamazaki, Satish Vedantam, Dongsheng Li, Hongbo Jiang, Seetharami Seelam, Felix Strohmeier, Syed Ali Khayam, Zheng Wan, Choudhury Al Sayeed, Torsten Braun, Bratislav Milic, Tim Moors, Thomas Gross, Lei Tian, Giuliano Casale, Sue Moon, Michael May, Wu Di, Minaxi Gupta, Dietrich Fahrenholtz, Kevin Butler, Geoff Voelker, Rich Wolski, Toni Cortes, Saurabh Bagchi, Jaehyuk Choi, George Pavlou, Yu Hua, Etienne Riviere, Rui Aguiar, Xiaoming Fu, Leon Ko, Alberto Montresor, Mark Jelasity, John Heidemann, Shaozhi Ye, Omer Rana, HyungJune Lee, Lex Stein, John Douceur, Sherman Chow, Fei Xing, Sailesh Kumar, Fernando Boavida, Azzedine Boukerche, Aniket Mahanti, Thomas Begin, Anirban Mahanti, Lucas Schnorr, Uichin Lee, Roch Guerin, Hao Wen, Guillermo Taboada, Wojciech Kabacinski, Hailun Tan, Yu (Jason) Gu, Yevgeni Koucheryavy, Carey Williamson, Indranil Gupta, Samer Al-Kiswany, Qing Yang, Adriana Iamnitchi, Jarmo Harju, Christina Poepper, Osamu Tatebe, Kento Aida, Jon Weissman, Toufik Ahmed, Robert Bestak, Guanying Wang, Mao Bo, Tobias Heer, Ehab Elmallah, Ryan Zhang, Seongho Cho, Konrad Iwanicki, Mark Claypool, Yee Wei Law, Pit Pichappan, Yu Huang, Bing Tang, Aleena Zahid, Maxim Martynov, Bu-Sung (Francis) Lee, Emilio Luque, Amos Waterland, Geert Heijenk, Khalid Al-Begain, Michael Menth, Matt Welsh, Jalaluddin Qureshi, Michel Hanna, Maxime Monod, Stephane Genaud, Qiang Zou, Allen MacKenzie, Dick Epema, Kuang Xu, Ennan Zhai, Elie Najm, Neal Patwari, Ricardo Jimenez-Peris, Van Nguyen, Bjorn Landfeldt, Aurelien Francillon, Jong Chun Park, Matthew Woitaszek, Eduard Glatz, Srikanth Krishnamurthy, Ana Pont, Pradeep Padala, Evgeny Osipov, Mohammad Abdul Awal, Mohamed Hefeeda, Kuai Xu, Indranil Gupta, Ruediger Kapitza, Christoph Rensing, Elmano Cavalcanti, Eyal Zohar, Jason Liu, Pedro Cuenca, Mario Di Francesco, Aruna Prem Bianzino, Mario Kolberg, Paolo Giaccone, Danny Hughes, Abderrahim Benslimane, Martin Röhricht, Markus Koegel, Matei Ripeanu, Matthias Wilhelm, Xun Shao, Stefano Avallone, Su Yi, Yinjin Fu, Vijay Gurbani, Dhinesh Dharman, Youngjae Kim, Yanhua Li, Ruichuan Chen, Renata Cruz Teixeira, Chun Tung Chou, Marco Conti, Angelo Castellani, Nizar Ben Neji, Zhou Liu, Raouf Boutaba, Elmano Ramalho Cavalcanti, Edmundo Madeira, Kartik Gopalan, Forrest Iandola, Eduardo Cerqueira, Eyuphan Bulut, Andrea Passarella, Stefano Secci, Animesh Pathak, Ramin Sadre, Gwendal Simon, Maha Abdallah, Evan Grim, Gary Wong, Mohammed Shehada, Erwin Laure, Paul Roe, Yuzhong Sun, Florentin Neumann, Zhongliang Zhao, Francesco Restuccia, Hanno Wirtz, Huichen Dai, Boris Koldehofe, Gary Wong, Rami Langar, Yuanjian Xing, Matth&#65533;us Wander, Murray Stokely, Radu Stoleru, Tom Pfeifer, Vasileios Giotsas, Fabio Ricciato, Bill Pitts, Filip De Turck, Prosper Chemouil, Enzo Mingozzi, Anna Sperotto, Jingon Joung, J&#65533;zef Wozniak, Arash Asadi, Harsha Chenji, Julius R&#65533;ckert, Anura Jayasumana, Rob Beverly, Philip Wette, Ali Dabirmoghaddam, Kyung-Hwa Kim, Ming Chen, Yehia Elkhatib, Yuming Jiang, Jordi Domingo-Pascual, Guillaume Doyen, Saeed Samet, Tony Luo Tie, Julius Schulz-Zander, Nils Aschenbruck, Esa Hyyti&#65533;, Claudia Bauzer Medeiros, Philipp Richter, Eduardo Mucelli, Steffen Bondorf, He Bingsheng, Lisandro Zambenedetti Granville, Vilen Looga, Pere Barlet, Stavros Toumpis, Kung-Kiu Lau, Wei Koong Chai, Mari Carmen Aguayo Torres, Ahmed Elhamy, Remi Badonnel, Nils gentschen Felde, Rahul Hiran, Marinos Charalambides, Ahmed Mostafa, Quirin Scheitle, Indranil Gupta, Azzam Mourad, John Harris, Thiago Teixeira, Dimitrios Katsaros, Noel Crespi, Noura Limam, Rafael Almeida, Matthias Wichtlhuber, Wei Li, Shihabur Rahman Chowdhury, Stefan Schmid, Tamoghna Ojha, Sven Zehl, Srihari Nelakuditi, Sathish Gopalakrishnan, Casas Pedro, Carlo Augusto Grazia, Yiouli Kritikou, Sebastien Varrette, Brendan Jennings, Theofanis Raptis, Marco Mellia, Oliver Gasser, Martin Gilje Jaatun, Varun Reddy, Matti M&#65533;ntym&#65533;ki, Kemal Akkaya, Anand Panangadan, Carol Fung, Quirin Scheitle, Tobias Hossfeld, Pedro Casas, Alberto Dainotti, Mu He, Li Lu, Don FJ, Francesco Ciaccia, Antonio Franco, Corinna Schmitt, Marco Fiore, Leila Saidane, Hwee-Pink Tan, Akarsh Prabhakara, Iordanis Koutsopoulos, Mohamed Kaaniche, Mike Wittie,

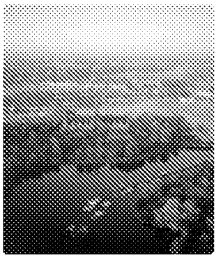
Cloudflare, Inc. Exhibit 1003 Page 1255

Declaration of Dr. Stefan Savage

ATTACHMENT L

ACM SIGCOMM 2000

Call for papers
Paper Submission
Organisation Committee
Program Committee
Conference program
Registration
Local information
Student travel awards





August 28 - September 1, 2000
Grand Hotel, Stockholm, Sweden

WELCOME TO ACM SIGCOMM 2000!

The organisation committee is most delighted to invite you to Grand Hotel, Stockholm, Sweden in August-September year 2000.

Per Gunningberg Stephen Pink

ACM SIGCOMM 2000 is the annual conference of the Special Interest Group on Data Communication (SIGCOMM), a vital special interest group of the Association for Computing Machinery (ACM).

ERICSSON

TELIA

Sprint

Full list of supporters

News

- Aug 2000: The technical program will be multicast on the Internet.
- Aug 2000: The outrageous opinions session seeks your input: the instructions are here, the deadline is approaching.
- Aug 2000: The final program is available, check out here.
- May 2000: The advance program has been published, see here.
- Feb 2000: A Newsweek article introduces Stockholm as Europe's Internet Capital.
- Jan 2000: Paper submission deadline is over - more than 230 papers submitted!

Important dates

Outrageous opinions desired:	August 25, 2000
Preconference program starts	August 28, 2000

Past dates

Paper submission:	January 28, 2000
Tutorial proposals:	February 28, 2000
Notification of acceptance:	April 21, 2000
Camera ready papers:	June 2, 2000
Proposals for student travel grant:	June 22, 2000
Proposals for poster session:	July 12, 2000
Advance registration ends:	July 28, 2000

For More Information Contact:

General Conference Chairs:
Per Gunningberg, Uppsala University, Sweden.
Stephen Pink, University of Arizona, USA.
E-mail: sigcomm2000-gcchairs@it.uu.se

Local Arrangements Chairs:
Bengt Ahlgren, Swedish Institute of Computer Science, Sweden
Christian Tschudin, Uppsala University, Sweden
E-mail: tschudin@docs.uu.se

Cloudflare, Inc. Exhibit 1003 Page 1257

Declaration of Dr. Stefan Savage

ATTACHMENT M

ACM SIGCOMM 2000
August 28 - September 1, 2000
Grand Hotel, Stockholm, Sweden

Call for papers
Paper Submission
Organisation Committee
Program Committee
• Conference program
Registration
Local information
Student travel awards

CONFERENCE PROGRAM

Multicast transmission on the Internet

We will multicast video and audio from the technical sessions of the conference live on the Internet. The transmission will start Wednesday August 30 at 9:00 and end Friday September 1 at 13:30. The local timezone is CEST (GMT+2). See the conference program for time details.

The conference can be received using the free Marratech player software, as well as the standard MBone tools. Please visit Marratech's web page for further technical information on the transmission from SIGCOMM as well as download instructions for their software.

The multicast transmission is performed by Marratech



Cloudflare, Inc. Exhibit 1003 Page 1259

Declaration of Dr. Stefan Savage

ATTACHMENT N

Cloudflare, Inc. Exhibit 1003 Page 1261

Practical Network Support for IP Traceback

Stefan Savage
University of Washington/
University of California, San Diego

David Wetherall, Anna Karlin and Tom Anderson
University of Washington, Seattle

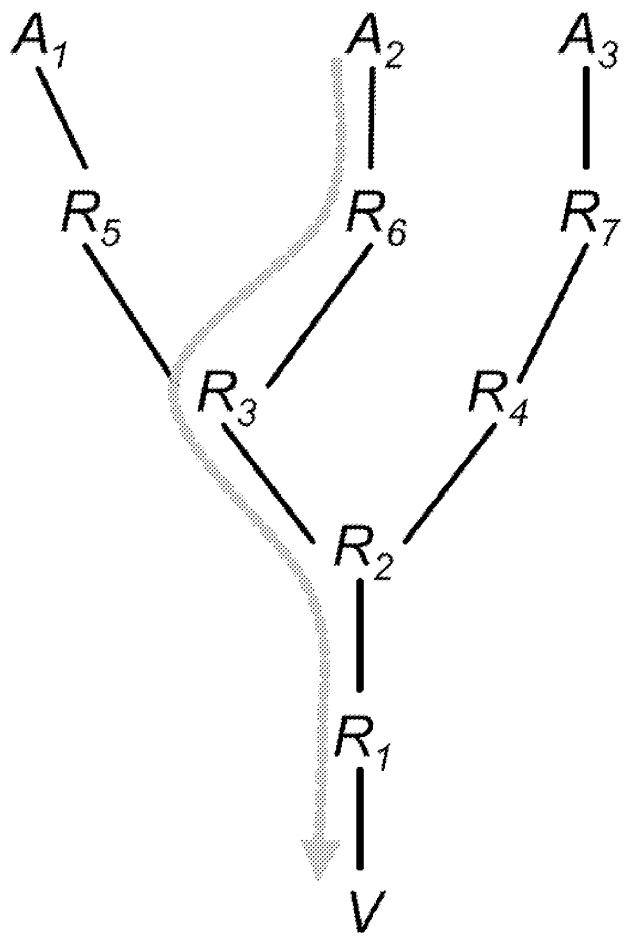
Motivation

- Denial-of-service via packet flooding
  - Attackers overwhelm network and CPU resources with spurious requests
- Internet architecture permits anonymity
  - Use of IP source address is **voluntary**
  - Attackers use “false” source address
  - Network is stateless (no audit trails)
- Difficult to respond to attack without knowing what network path(s) it traverses

Cloudflare, Inc. Exhibit 1003 Page 1262

Traceback problem

- Basic elements
  - Set of attackers A_i
  - Set of routers R_i
  - Victim V
- Attack path for A_i
  - Ordered list of routers between A_i and V
  - e.g. $\{R_6, R_3, R_2, R_1\}$
- Goal
  - Determine attack path for each attacker



Cloudflare, Inc. Exhibit 1003 Page 1263

State of the practice

- Router traffic stats + manual guesswork
- Input debugging
  - Install filter to detect attack traffic, determine upstream router, repeat
- Serious practical limitations
  - High management overhead, per-ISP only
  - Attack must be in progress

Cloudflare, Inc. Exhibit 1003 Page 1264

Design constraints (self-imposed)

- **Assumptions about environment**
  - Attacker can generate any packet
  - Multiple attackers may conspire (DDoS)
  - Attackers send many packets
  - Routers resource limited (no *per-flow* state)
  - Routers not malicious
- **Goals**
  - Real-time and *post-mortem* trace capability
  - Incrementally deployable
  - Low management and network overhead

Cloudflare, Inc. Exhibit 1003 Page 1265

Key idea

- **Routers store path state in packets**
- Naïve approach
  - Routers append their address to each packet
  - Can be expensive to implement
  - No assurance of “enough” space in packet
- Spread state across packets?

Cloudflare, Inc. Exhibit 1003 Page 1266

Edge sampling: basic approach

- Augment packets with additional fields
  - **Edge**: two adjacent router addresses (start & end)
  - **Distance**: # edges traversed since marked
- Probabilistically mark packets in routers
  - Marked packets contain “sample” of path
- Victim, or victim’s ISP, reconstructs path after receiving enough packets

Cloudflare, Inc. Exhibit 1003 Page 1267

Marking procedure at router

- Marking procedure for packets forwarded by *R*:

```
with probability p,
   write R into start field
   write 0 into distance field
else
   if distance == 0 then
      write R into end field
   increment distance field
```

Cloudflare, Inc. Exhibit 1003 Page 1268

Path reconstruction at victim

- Extract identifiers from attack packets
- Build graph rooted at victim
  - Each (*start*,*end*,*distance*) tuple is an edge
  - Eliminate edges with inconsistent distance
  - Traverse edges from root to find attack paths
- # packets needed to reconstruct path

$$E(X) < \frac{\ln(d)}{p(1-p)^{d-1}}$$

p: marking probability
d: length of path

Cloudflare, Inc. Exhibit 1003 Page 1269

Robustness

- Random marking decisions
  - Can’t be anticipated by attacker
  - Can’t be controlled by attacker
- Attackers can “lie” and invent edges
  - But not for *distances* less than their own
- Offline validation of “valid suffix”

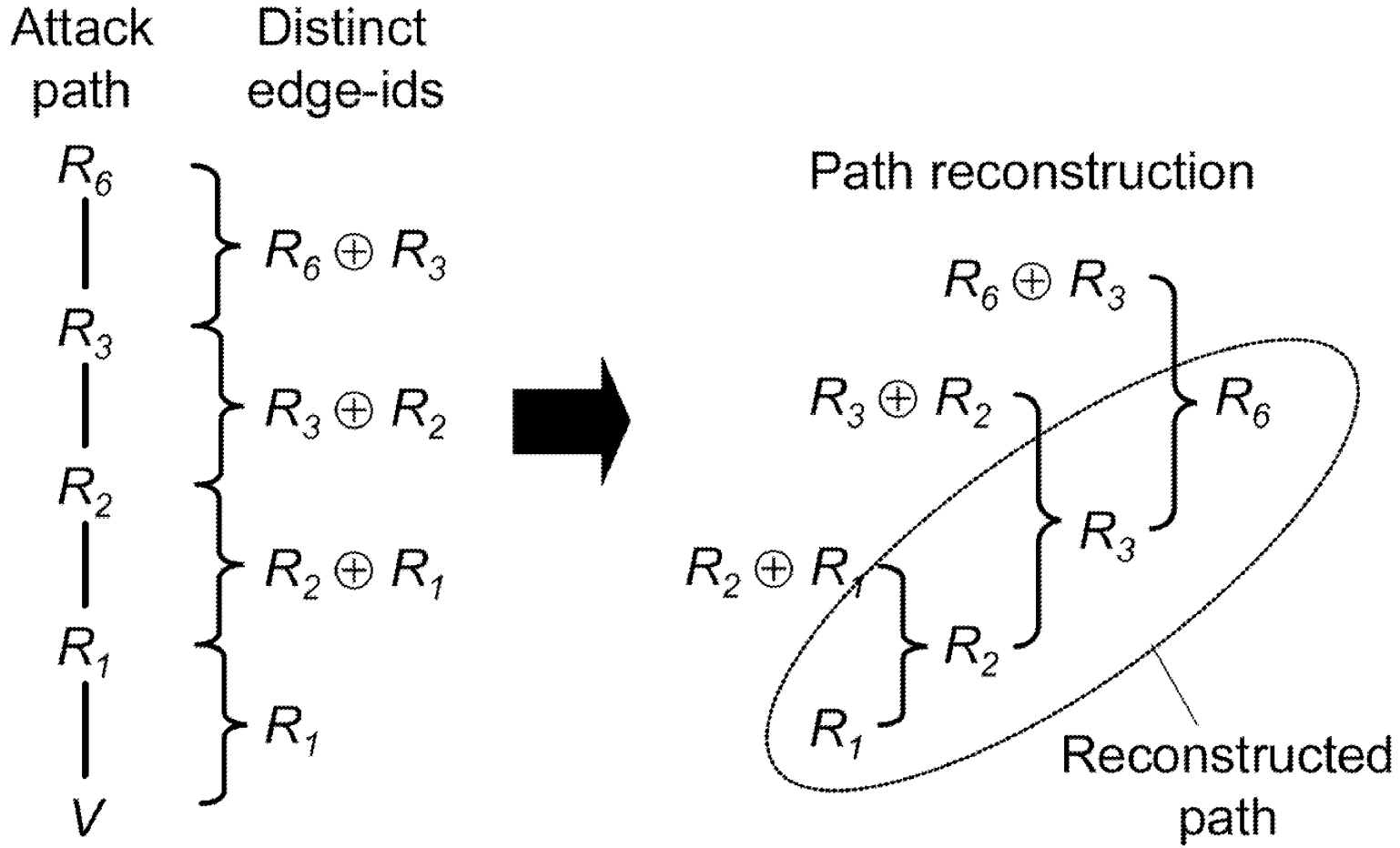
Cloudflare, Inc. Exhibit 1003 Page 1270

Practical problems

- Deployment issues
  - How to fit edge/distance data into IP header
  - Current scheme requires too much space
    - 32 bits for each router address, 8 for distance
    - Need lower per-packet overhead
- One hybrid approach
  - Trade space for reconstruction/robustness
  - Overload existing packet header field

Cloudflare, Inc. Exhibit 1003 Page 1271

Compressing edges to edge-ids using XOR



Cloudflare, Inc. Exhibit 1003 Page 1272

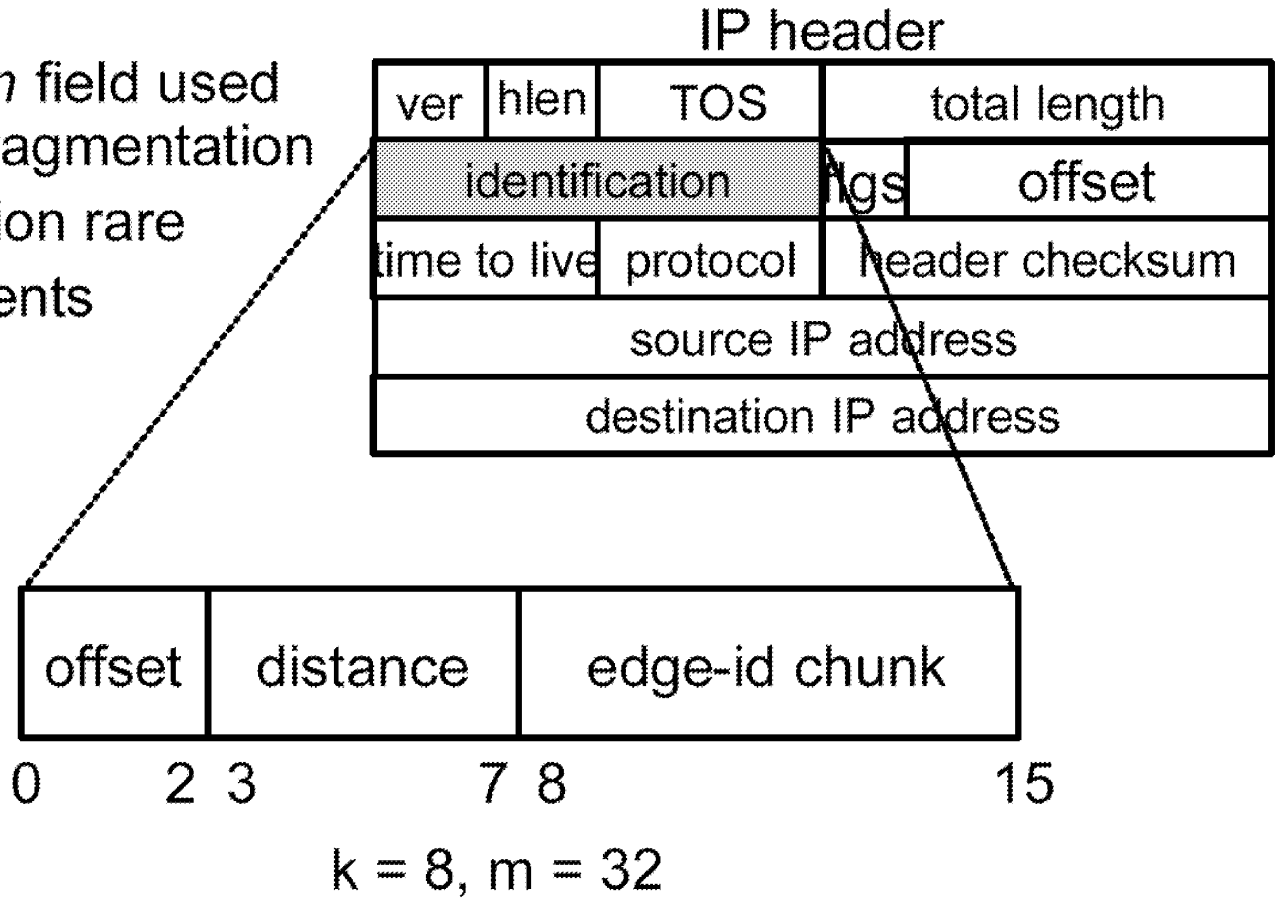
Edge-id chunks

- Divide each edge-id into *k* chunks
- When marking packet
  - Pick edge-id chunk at random
  - Write chunk and its offset into packet
- Increases reconstruction time by $k \ln(k)$
- Chunks may not be unique
  - Augment edge-id with hash of m bits
  - Validate chunk combinations at victim

Cloudflare, Inc. Exhibit 1003 Page 1273

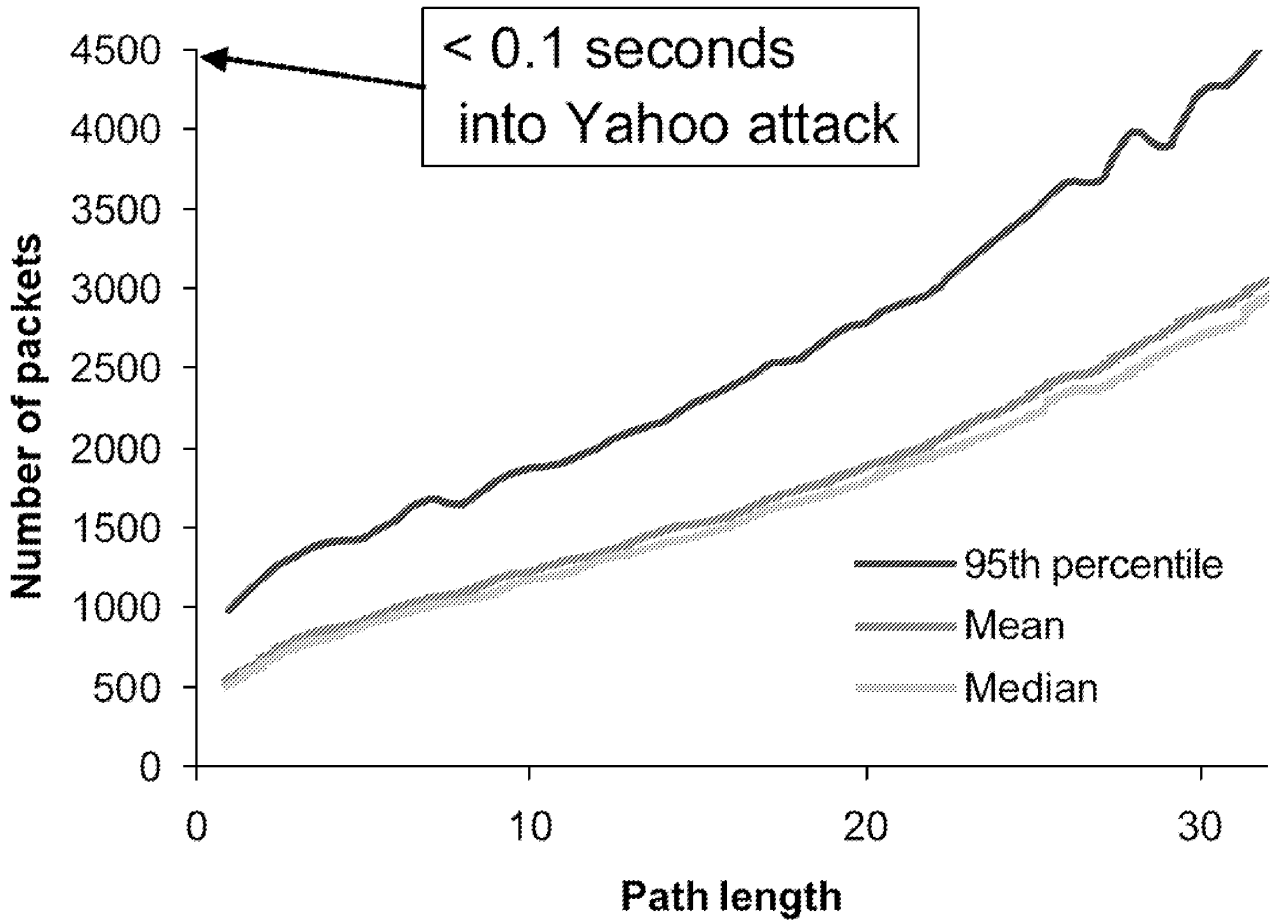
One opportunistic encoding

- *Identification* field used for packet fragmentation
- Fragmentation rare
- Mark fragments separately



Cloudflare, Inc. Exhibit 1003 Page 1274

Experimental convergence time



Cloudflare, Inc. Exhibit 1003 Page 1275

Weaknesses

- Path validation/authentication
- Robustness in highly distributed attacks
  - *Both addressed nicely in [Song&Perrig00]*

- Compatibility issues (IPsec AH, IPv6)
- Origin laundering (reflectors, tunnels, etc)

Cloudflare, Inc. Exhibit 1003 Page 1276

Summary

- An efficient algorithm for tracing anonymous attacks: *edge sampling*
- Hybrid algorithm with reduced per-packet space overhead (at a cost)
- Potential encoding into current IP packet header

Cloudflare, Inc. Exhibit 1003 Page 1277



Traceback is only one part of the problem

- **Detection**
  - How do you know you're under attack?
  - What identifies an attack?
- **Countermeasures**
  - What do you do after you backtrace the attack?
  - How to filter attack but not customer traffic?
  - How to enable inter-ISP cooperation?

Cloudflare, Inc. Exhibit 1003 Page 1279

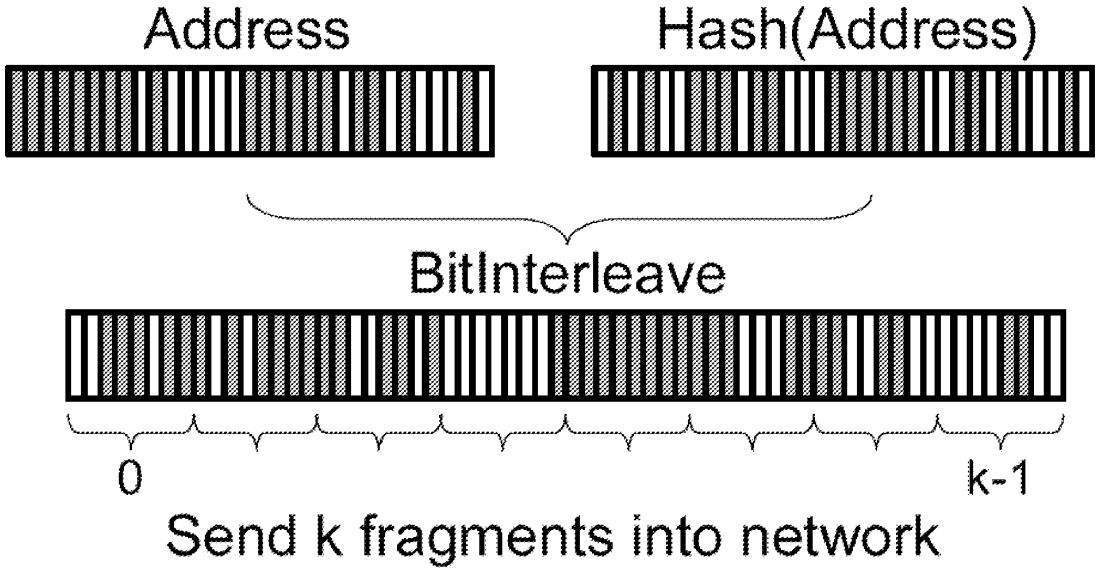
Ingress/egress filtering

- Idea is that routers block packets with "invalid" source addresses
- Great idea, but…
  - Really only possible at edge of network
  - Requires universal deployment
  - Management overhead for multi-homed hosts
  - Rarely employed inside enterprises (still 2^{16} "valid" source addresses at UW)

Cloudflare, Inc. Exhibit 1003 Page 1280

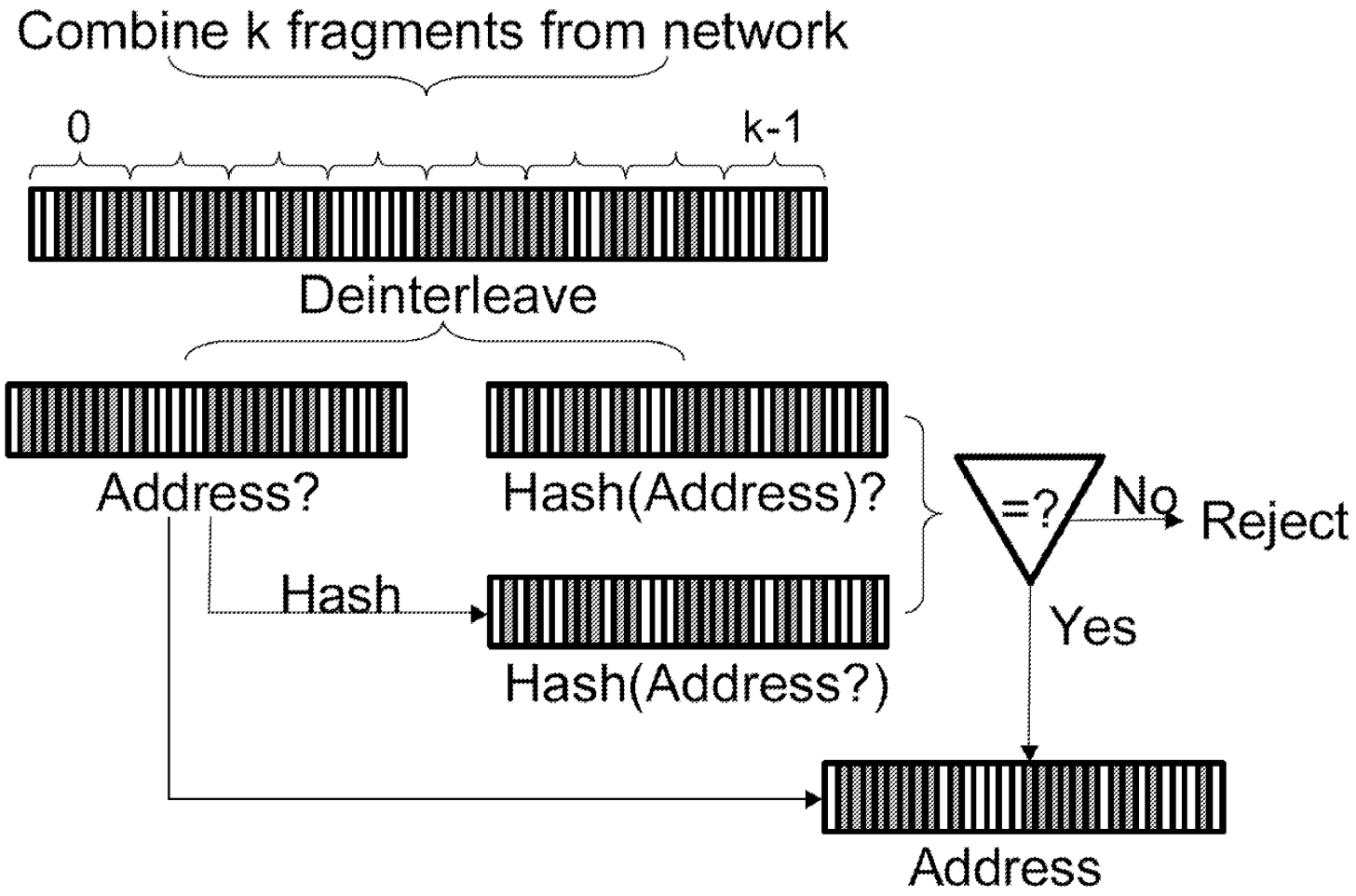
Augment router id with hash

Interleave address data with random hash



Cloudflare, Inc. Exhibit 1003 Page 1281

Hash-based consistency check



Cloudflare, Inc. Exhibit 1003 Page 1282

Modified path reconstruction

- Sort edge-id fragments by distance
  - For each distance, sort by offset
  - Try all combinations until hash matches
- Probability of "false positive" at distance d

$$1 - \left(1 - \frac{1}{2^h}\right)^{m^k}$$

m: # disjoint edges at distance d
h: length of hash in bits

- Design tension in size of *k* and *m*

Cloudflare, Inc. Exhibit 1003 Page 1283

ICMP Traceback [Bellovin, Draft spec]

- **Sample packets in routers** (~1/20,000)
- **Generate special ICMP message**
  - Includes forward hop info, reverse hop info, authentication & origin packet contents
  - Set TTL to 255
- **Reconstruct path at victim**
  - TTL ordered by distance
  - Match forward/reverse hop info

Cloudflare, Inc. Exhibit 1003 Page 1284

Comparison

- Very similar, but out-of-band
- Advantages
  - Has authentication built in
  - Few backwards-compatibility issues
- Disadvantages
  - ICMP increasingly filtered
  - Painful to trace through discontinuities caused by partial deployment
  - Must associate attack and signal
- Open challenges for both
  - Chaff (clogging attacks) and laundering

Cloudflare, Inc. Exhibit 1003 Page 1285

Edge sampling algorithm

- **Probabilistically mark packets in routers**
  - *Edge*: Two adjacent router addresses
  - *Distance*: edges traversed since marked – starts at 0
  - If packet **not** marked, increment distance
- **Reconstruct path at victim**
  - Collect marked packets, construct graph
  - # packets needed to reconstruct path

$$E(X) < \frac{\ln(d)}{p(1-p)^{d-1}}$$

p: marking probability

d: length of path in hops

Cloudflare, Inc. Exhibit 1003 Page 1286

Exhibit 1016

Curriculum Vitae of Dr. Stefan Savage

Customer No.: 000027683

Haynes and Boone, LLP
IP Section
2323 Victory Avenue, Suite 700
Dallas, Texas 75219
Telephone [214] 651.5000
Fax [214] 200.0853

Stefan Savage

Department of Computer Science and Engineering
University of California, San Diego
9500 Gilman Drive, MC 0404
La Jolla, CA 92093-0404

Email: savage@cs.ucsd.edu
Phone: (858) 822-4895
Fax: (858) 534-7029
URL: http://www.cs.ucsd.edu/~savage/

Education

Ph.D., Computer Science and Engineering — January, 2002
University of Washington, Seattle, WA.

B.S., Applied History — May 1991
Carnegie Mellon University, Pittsburgh, PA.

Research Interests

Design and analysis of experimental computer and communications systems, with a focus on security

Academic Employment

University of California at San Diego, Department of Computer Science and Engineering
- Assistant Professor, July 2000 - June 2006
- Associate Professor, July 2006 – June 2010
- Professor, July 2010 – present

University of California at San Diego, Center for Networked Systems
- Associate Director, July 2009 – July 2010
- Director, July 2010 – present

Professional Experience

ZeroFox, Baltimore, MD — 2013-2016
Technical Advisory Board

Impermium, San Francisco, CA — 2011-2013
Technical Advisory Board

Anonymizer Inc, San Diego, CA — 2010-2012
Consultant

Microsoft, Mountain View, CA — 2010
Consultant

Monitor 360, San Francisco, CA — 2010
Consultant

Netsift, San Diego, CA — 2004-2005
Co-founder & Consultant

Rendition Networks, Redmond, WA — 2001-2005
Strategy Advisory Council Member

Asta Networks, Seattle, WA — 2000-2002
Co-founder and Chief Scientist

Appliant, Seattle, WA — 2000
Consultant, Research Group

Digital Equipment Corporation, Palo Alto, CA — June 1996- September 1996
Intern, Systems Research Center

Hewlett Packard Corporation, Palo Alto, CA — July 1995
Visiting Researcher, Storage Systems Program

Honeywell Corporation, Clearwater, FL — July 1993
Consultant, Space Systems Center

Carnegie Mellon University, Pittsburgh, PA — 1991-1994

A variety of expert witness and legal support engagements

Awards and Honors

IEEE Computer Society Technical Committee on Security and Privacy Distinguished Service Award, 2019
MacArthur Fellow, 2018-2023
Irwin Mark and Joan Klein Jacobs Chair in Information and Computer Science, 2016
ACM Prize in Computing, 2016 (*previously the ACM Infosys Award*)
University of Washington, Alumni Achievement Award, 2016
ACM SIGOPS Mark Weiser Award, 2013
ACM Fellow, 2010
Alfred P. Sloan Research Fellowship, 2004-2006.

16 Award/Honored papers
- IETF **Applied Networking Research Prize**, 2021 (Trufflehunter)
- IEEE Security & Privacy **Test of Time Award**, 2020 (Security Analysis of a Modern Automobile)
- ACM Computer and Communications Security **Test of Time Award**, 2019 (Hey, you, get off of my cloud)
- Distinguished Paper, USENIX Security, 2019 (Lateral Phishing)
- USENIX Security **Test of Time Award**, 2017 (Backscatter)
- Selected Paper, ACM ASPLOS, 2011 (forwarded to ACM TOCS) (LogEnhancer)
- Jay Lepreau Best Paper Award, USENIX OSDI, 2008 (invited to CACM) (Difference Engine)
- Selected Paper, ACM Computer and Communications Security, 2008 (invited to CACM) (Spamalytics)
- Selected Paper, ACM Computer and Communications Security, 2008 (forwarded to ACM TISSEC) (RoC)
- IEEE William C. Carter Award Paper, IEEE DSN, 2005 (forwarded to IEEE TDSC) (Fatih)
- Selected Paper, International Conference on Network Protocols, 2001 (to IEEE/ACM ToN) (ECN nonce)
- Best Paper, USENIX Security Symposium, 2001 (Backscatter)
- Selected Paper, ACM SIGCOMM, 2001 (forwarded to IEEE/ACM TON) (Traceback)
- Best Student Paper, USENIX Symposium on Internet Technologies and Systems, 1999 (Sting)
- Selected Paper, IEEE Micro, 1999 (forwarded from 1998 IEEE Hot Interconnects) (Detour)
- Best Paper, ACM Symposium on Operating Systems Principles, 1997 (forwarded to ACM TOCS) (Eraser)
- Best Student Paper, USENIX Technical Conference, 1996 (AFRAID)

Network World Magazine's "Top 10 Wicked Cool Algorithms for 2008" (for XL), 2008.
William Chan Dissertation Award, University of Washington, 2002.
Jacobs Faculty Fellowship, 2001-2003.

Past/Current Research Funding

Google Faculty Research Award, 2020, PI (co-PI Aaron Schulman), "**Passive Network Analysis for Security Analysis and Assessment**" $70,750.

Facebook, 2018, PI (co-PI Geoff Voelker), "**Securing the Internet**" Award, $75,000.

Samsung GRO, 2017-2018, co-PI (PI Tajana Rosing, co-PI's Gary Cottrell and Kirill Levchenko), "**Smart and Safe Home**", $300,000.

Department of Homeland Security, 2018-2012, co-PI (Kirill Levchenko PI, Geoff Voelker, Alex Snoeren and Vern Paxson co-PIs), "**Foundations of Threat Intelligence Metrics**", $1,045,000.

Department of Homeland Security (via a subcontract from PNNL), 2017-2018, "**Security Evaluation of the Boeing 757 Environmental Systems (ACI)**", co-PI (PI Kirill Levchenko, Co-PI Alex Snoeren), $350,000.

National Science Foundation, 2017-2020, PI (co-PI's Geoff Voelker and Kirill Levchenko), "**Large-scale Characterization of DNS Abuse**", $1,200,000.

Hewlett Foundation, 2016-20187, PI (co-PI's Alex Snoeren and Kirill Levcheko), "**Quantifying the Legal Risks on Third-party Vulnerability Research**", $100,000.

Hewlett Foundation, 2016-2017, PI (co-PI's Geoff Voelker and Kirill Levchenko), "**Reasoning about Zero-Day Vulnerabilities**", $100,000.

National Science Foundation, 2016-2019, co-investigator (PI Kirill Levchenko, co-PIs Ranjit Jhala and Alex Snoeren), "**Foundations of Security Cyber-Physical Systems of Systems**", $560,000.

National Science Foundation, 2016-2019, co-investigator (PI Alex Snoeren, co-PIs Kirill Levchenko, George Porter and Geoff Voelker), "**II-New: A Dual-Purpose Data Analytics Laboratory**", $548,000

Google Research Gift, 2016, $100,000 (PI)

Google Research Gift, 2014, $250,000 (co-investigator w/Geoff Voelker and Kirill Levchenko)

Facebook gift, 2014, $50,000 (PI)

Google Research Gift, 2012, $200,000 (co-investigator w/Geoff Voelker and Kirill Levchenko)

National Science Foundation, 2012-2017, PI (co-PIs Geoff Voelker, James Fowler, Alex Snoeren, Lawrence Saul, Kirill Levchenko, Erin Kenneally, Vern Paxson, Mark Allman, Chris Grier, Chris Hoofnagle, Dan Klein, Christian Kreibich, Deirdre Mulligan, Nicholas Weaver and Damon McCoy), "**Beyond Technical Security: Developing an Empirical Basis for Socio-Economic Perspectives**", $10,000,000 (4.7M to UCSD, 4.7M to ICSI, 06M to GMU)

Microsoft Research Gift, 2012, $50,000 (co-investigator w/Geoff Voelker)

National Science Foundation, 2011-2014, Co-Investigator (PI Alex Snoeren, co-PI Kirill Levchenko), "**Understanding Network Failure**", $450,000.

Yahoo Research Gift, 2011, $10,000 (co-investigator w/Geoff Voelker and Kirill Levchenko)

Google Research Gift, 2010, $400,000 (co-investigator w/Geoff Voelker and Kirill Levchenko)

Cisco Research Gift, 2010, $90,000 (co-investigator w/Geoff Voelker and Kirill Levchenko)

National Science Foundation, 2010-2012, Principal Investigator (Co-PIs Yoshi Kohno and Ingolf Kreuger), "**TC:Medium:Collaborative Research:Foundations, Architectures, and Methodologies for Secure and Private Cyber-physical Vehicles**", $1,200,000 ($600k to UCSD, $600k to UW).

Google Research Gift (DC Office), 2010, $5,000.

Google Research Gift, 2010, Co-Investigator (PI Kirill Levchenko), "**Understanding the Economics of CAPTCHAs**", $50,000.

UCSD Center for Network Systems, 2009-2011, Co- Investigator (PI Hovav Shacham), "**Future is Cloudy: Security and Availability Issues in Outsourced Computing**", $104,000.

UCSD Center for Network Systems, 2009-2011, Co- Investigator (PI Alex Snoeren), "**Understanding Failure Patterns in Real Networks**", $104,000.

Yahoo! Research Gift, 2009, $50,000.

National Science Foundation, 2009-2012, Co-Investigator (PI Mohan Paturi, Co-PIs Alex Snoeren, Mathew McCubbins, Joel Sobel), "**NetSE:Medium:Network Structure, Incentives, and Outcomes**", $900,000.

Office of Naval Research, Multi-University Research Initiative, 2009-2014, Principal Investigator (CoPIs Geoff Voelker, Vern Paxson, Dawn Song and Dan Klein) "**Infiltration of Botnet Command & Control and Support Ecosystems**", $7,500,000 (~3.8M to UCSD, ~3.7M to Berkeley)

Cisco gift, 2009, Principal Investigator (Co-PI Geoff Voelker), $37,500.

Google Research Gift, 2008, Principal Investigator (Co-PI Geoff Voelker), "**Scam Analysis and Defense via Botnet Infiltration**", $70,000.

Yahoo! Research Gift, 2008, $80,000.

Microsoft Research Gift, 2008, $12,500.

National Science Foundation, 2008-2010. Principal Investigator (Co-PI Geoff Voelker), "**Understanding and Exploiting Economic Incentives in Internet-based Scams**", CNS-0831138, $500,000.

UCSD Center for Network Systems, Principal Investigator (Co-PI Geoff Voelker), "**Bluesky: System Support for Transparent Cloud Computing**", $129,500.

UCSD Center for Network Systems, 2008-2010, Principal Investigator (Co-PI Geoff Voelker), "**Scam Analysis and Defense via Botnet Infiltration**", $129,500.

UC Discovery Grant, 2007. Principal Investigator (Co-PIs Geoff Voelker and Alex Snoeren), "**System Support for Derived Data Management**", $94,602.

UC Discovery Grant, 2007: Co-Investigator (Geoff Voelker PI, Amin Vahdat and Alex Snoeren co-PIs), "**Extending Virtual Cluster Management and Resource Utilization**", $94,602

Cisco Systems Gift, 2007. Principal Investigator (Co-PIs Geoff Voelker) , "**Data-based Security Policies and Miscreant Infrastructure Analysis**", $200,000.

Google Gift, 2007. Principal Investigator (Co-PIs Geoff Voelker and Alex Snoeren), "**Automated Cross-layer Diagnosis of 802.11 Wireless Networks**", $65,000.

National Science Foundation, 2007-2009. Co-Investigator (Alex Snoeren, PI, Amin Vahdat and Yoshi Kohno Co-PIs), "**Privacy Preserving Attribution and Provenance**", CNS-0722031, $636,000.

Microsoft Research Gift, 2007. $25,000.

CNA Insurance Gift, 2007. $50,000.

National Science Foundation, 2006-2007. Co-Investigator (Alex Snoeren, PI, Amin Vahdat Co-PI), "**Enabling Defense and Deterrence through Private Attribution**", CNS-0627157, $400,000.

UCSD Center for Network Systems, 2006-2008, Principal Investigator (Co-PIs Geoff Voelker and Alex Snoeren), "**Automated Cross-layer Diagnosis of Enterprise Wireless Networks**", $110,000.

Microsoft Research Gift, 2006, $25,000.

UCSD Center for Network Systems, 2005-2007, Co-Investigator (Amin Vahdat PI), "**Agile Virtual Clusters**", $140,000.

National Science Foundation, 2004-2009. Principal Investigator (co-PIs Vern Paxson, Geoffrey Voelker, George Varghese and Nicholas Weaver), Cybertrust Center Grant, "**Center for Internet Epidemiology and Defenses**", CNS-0433668 and CNS-0433702, $6,200,000 ($3.1M to UCSD, $3.1M to ICSI).

UCSD Center for Network Systems, 2004-2006, Principal Investigator (Co-PIs Kimberly Claffy, Alex Snoeren, Amin Vahdat, George Varghese and Geoff Voelker), "**Internet Epidemiology**", $240,000.

UCSD Center for Network Systems, 2004-2006, Co-Investigator (Geoff Voelker PI), "**Cross-layer Wireless Measurement**", $160,000.

Microsoft Research Gift, 2004, $50,000.

Sloan Research Fellowship, 2004-2005**,** $40,000**.**

National Science Foundation, 2003-2006. Co-Investigator (Andrew Chien PI, David Kriegman and Joseph Pasquale Co-PIs), Infrastructure Grant, "**FWGrid: A Research Infrastructure with Fast Wireless, Wired, Compute and Data Infrastructure for Next Generation Systems and Applications**", EIA-0303622, $1,800,000.

National Science Foundation, 2003-2006. Co-Investigator (David Moore PI and Geoffrey Voelker Co-PI). "**Quantitative Network Security Analysis**", CNS-0311690**,** $372,183.

AT&T Network Engineering Center Gift, 2003 (managed w/George Varghese)**,** $500,000.

National Institute of Standards, 2001-2004. Co-Investigator (George Varghese PI), 60NANB1D0118, "**Real-Time Intrusion Detection**", $909,951.

Defense Advanced Research Projects Agency, 2001-2004. Principal Investigator (Co-PIs Geoffrey M. Voelker and Keith Marzullo), N66001-01-1-8933, "**Reliable Adaptive Multi-Path Routing (RAMP)**", $2,293,239.

Student Advising

Ph.D. Students Graduated (17):

- Ranjita Bhagwan, "**Automated Availability Management**", (Ph.D. 2004, co-advised with Geoff Voelker -> IBM Research -> Microsoft Research),
- John Bellardo, "**Secure and Efficient Wireless Networks**", (Ph.D. 2006 -> California Polytechnic State University),
- Yu-Chung Cheng, "**Automating Cross-Layer Diagnosis of Enterprise 802.11 Wireless Networks**" (Ph.D. 2007, co-advised with Geoff Voelker -> Google),
- Alper Mizrak, "**Detecting Malicious Routers**" (Ph.D. 2007, co-advised with Keith Marzullo -> VMWare),
- Kirill Levchenko, "**XL: A Communication-Efficient Routing Protocol**", (Ph.D. 2008, co-advised with Mohan Paturi -> Research Scientist at UCSD),
- Justin Ma, "**Learning to Detect Malicious URLs**", (Ph.D. 2010, co-advised with Geoff Voelker and Lawrence Saul -> Postdoc at UC Berkeley -> Google),
- Michael Vrable, "**Migrating Enterprise Storage Applications to the Cloud**", (Ph.D. 2011, co-advised with Geoff Veolker -> Google),
- Marti Motoyama, "**Understanding the Role of Outsourced Labor in Web Service Abuse**", (Ph.D. 2011, co-advised with Geoff Voelker and George Varghese -> FitBit),
- Chris Kanich , "**Characterizing Internet Scams Through Underground Infiltration**", (Ph.D. 2012, co-advised with Geoff Voelker -> University of Illinois, Chicago), Andreas Pittsilidis, "**Spam Value Chain: Defensive Intervention Analysis**", (Ph.D. 2013, co-advised with Geoff Voelker -> Google),
- Daniel Turner, "**Opportunistically Reconstructing a Network's Failure History**", (Ph.D. 2013, co-advised with Alex Snoeren -> TurboSquid),
- Sarah Meiklejohn, "**Flexible Models for Secure Systems**", (Ph.D. 2014, co-advised with Mihir Bellare -> University College London),
- David Wang, "**A Comprehensive Approach to Undermining Search Result Poisoning**", (Ph.D. 2014, co-advised with Geoff Voelker -> Google),
- He "Lonnie" Liu, "**A High Performance Hybrid ToR for Data Centers**", (Ph.D. 2015, co-advised with Geoff Voelker -> Google),
- Tristan Halvorson, "**Registration Intent in the Domain Name Market**", (Ph.D. 2015, co-advised with Geoff Voelker -> Google),
- Matt Der, "**Investigating Large-Sale Internet Abuse Through Web Page Classification**", (Ph.D. 2015, co-advised with Lawrence Saul and Geoff Voelker -> notch.io),
- Neha Chachra, "**Understanding URL Abuse for Profit**" (Ph.D. 2016, co-advised with Geoff Voelker -> Facebook),
- Wilson Lian, "**JIT Spraying Threats on ARM and Defense by Diversification**" (Ph.D. 2016, co-advised with Hovav Shacham -> Google),
- Louis Dekoven, "**Addresing Device Compromise from the Perspective of Large Organizations**" (Ph.D. 2019, co-advised with Geoff Voelker -> Facebook)

- Guo "Vector" Li, "**An Empirical Analysis on Threat Intelligence: Data Characteristic and Real-World Uses**" (Ph.D. 2020, co-advised with Kirill Levchenko -> Google)

Current Ph.D. Thesis Advisor (5): Ariana Miran, Gautam Akiwate and Alex Liu (w/Geoff Voelker) and Audrey Randall (w/Geoff Voelker and Aaron Schulman), Evan Johnson (co-advised with Deian Stefan)

M.S. Students Graduated (19):

- Doug Brown (2003 -> NYU Law School, co-advised w/Geoff Voelker),
- Ryan Sit (2004 -> Novatel, co-advised with William Griswold),
- Ishwar Ramani (2005 -> Juniper Networks),
- Chris Tuttle (2006 -> Google, co-advised with Alex Snoeren),
- Varun Almaula (2008 -> Cisco Systems),
- Ryan Roemer (2009 -> Microsoft, co-advised with Hovav Shacham),
- Sebastian Becerra (2009 -> Peace Corps),
- Grace Wang (2010 -> Qualcomm, co-advised with Hovav Shacham),
- Kourosh Derakshan (2011 -> Qualcomm),
- Sushma Bannur (2011 -> Microsoft, co-advised with Lawrence Saul),
- Devin Lundberg (2014 -> Pintrest, co-advised with Kirill Levchenko),
- Chris Gardner (2014 -> Arris),
- Malathi Raghavan (2014 -> Google, co-advised with Geoff Voelker),
- Gauytam Akiwate (2015 -> PureStorage, co-advised with Lawrence Saul and Geoff Voelker),
- Mohit Kothari (2015 -> Groupon, co-advised with Lawrence Saul and Geoff Voelker),
- Ian Foster (2015 -> Salesforce, co-advised with Kirill Levchenko),
- Louis Dekovan (2015 -> UCSD Ph.D., co-advised with Geoff Voelker),
- Ben Braun (2016 -> Facebook),
- Matthew Bohl (2016 -> Adobe)

Undergraduate independent projects (2): Derrick Wong (2004 -> Intel), Duong Harrison (2006 -> UPenn Ph.D. program), Brandon Enright (2007-9), Jason Bradshaw (2012), Tiffany Allan (2014-2015), Aaron Hurtado (2014-2015), Andrew Jabassa (2014-2015), Kelsey Ma (2014-2015), Ansel Blum (2018-2019)

Teaching Experience

CSE 123, "Communications Networks",
Fall 2005 CAPES: enrollment: 109, recommend class: 95%, recommend instructor: 86%
Winter 2005 CAPES: unranked
Fall 2009 CAPES: enrollment: 52, recommend class: 89%, recommend instructor: 97%
Fall 2011 CAPES: enrollment: 129, recommend class: 90%, recommend instructor: 84%
Spring 2015 CAPES: enrollment: 102, recommend class: 88%, recommend instructor: 92%
Fall 2016 CAPES: enrollment 103, recommend class: 89.7%, recommend instructor, 89.7%

CSE 123b, "Communications Software",
Spring 2002 CAPES: enrollment: 72, recommend class: 92%, recommend instructor: 89%
Spring 2003 CAPES: enrollment: 67, recommend class: 93%, recommend instructor: 90%
Spring 2004 CAPES: enrollment: 101, recommend class: 93%, recommend instructor: 93%

CSE 127, “Computer Security”,
Spring 2005 CAPES: enrollment: 63, recommend class: 98%, recommend instructor: 96%
Winter 2006 CAPES: enrollment: 71, recommend class: 98%, recommend instructor: 88%
Spring 2007 CAPES: enrollment: 83, recommend class: 98%, recommend instructor: 98%
Spring 2008 CAPES: enrollment: 59, recommend class: 100%, recommend instructor: 100%
Spring 2009 CAPES: enrollment: 54, recommend class: 100%, recommend instructor: 100%
Spring 2012 CAPES: enrollment: 137, recommend class: 96%, recommend instructor: 91%
Fall 2015 CAPES: enrollment: 152, recommend class: 96%, recommend instructor: 94%
Spring 2018 CAPES: enrollment: 170, recommend class: 100%, recommend instructor: 100%
Fall 2018 CAPES: enrollment 168, recommend class: 91.7%, recommend instructor: 91.7%
Spring 2019 CAPES: enrollment: 220, recommend class :96%, recommend instructor: 98%
Spring 2020 CAPES: enrollment: 206, recommend class: 100%, recommend instructor: 100%

CSE 221, “Graduate Operating Systems”,
Fall 2003: enrollment: 49, recommend class: 83%, recommend instructor: 89%

CSE 222a, “Graduate Communications Networks”,
Fall 2001: enrollment: 30, recommend class: 96%, recommend instructor: 94%
Winter 2003: enrollment: 46, recommend class: 92%, recommend instructor: 95%

CSE 227, “Graduate Computer Security”,
Winter 2008: enrollment: 21, recommend instructor: 100%
Spring 2010: enrollment: 22, recommend instructor: 100%
Winter 2012: enrollment: 28, recommend instructor: 100%
Winter 2015, Spring 2017, Fall 2019

GPAA477 (previously IRGN490) “Cyber Security” (co-taught w/Peter Cowhey IR/PS, Spring 2011, Spring 2013, Spring 2014, Spring 2016, Spring 2018, Spring 2019, Spring 2020.

CSE 291, “Internet Crime”, Fall 2008. (graduate seminar)
CSE 291, “Cyber Security, Privacy and US Law, Winter 2021 (graduate)

CSE 294, “Systems and Networking Seminar”, Winter 2003, Spring 2005, Fall 2005, Fall 2006, Spring 2007, Fall 2007.

CSE 291, “Selected Topics in Wide-Area Networking”, Winter 2001, Spring 2002.

Program Committee Service

2020 USENIX CSET

2019 ACM Symposium on Computer Science and Law

Program Co-chair, 2016 USENIX Security Symposium

2015 ACM SIGCOMM

2012 IEEEE CREDS

2012 USENIX/ACM OSDI

Program Co-chair, 2011 IEEE eCrime Symposium

2011 ACM SIGCOMM

2010 USENIX Workshop on Large-scale Exploits and Emerging Threats (LEET)

2010 USENIX/ACM Networked Systems: Design and Implementation Conference (NSDI)

2009 ACM Symposium on Operating System Principles (SOSP).

2009 IEEE Symposium on Security and Privacy (Oakland).

Program Co-chair, 2008 ACM SIGCOMM.

2008 USENIX Workshop on Organizing Workshops, Conferences and Symposia for Computer Systems (WOWCS).

Program Co-chair, 2007 USENIX/ACM Hot Networks Workshop.

2007 ACM Workshop On Recurring Malware (WORM) at the ACM Computer and Communications Security Conference.

2007 ACM Symposium on Operating System Principles (SOSP).

2007 ACM SIGCOMM Conference.

2006 IEEE Symposium on Security and Privacy (Oakland).

2005 ACM Symposium on Operating Systems Principles (SOSP).

2005 USENIX/ACM Networked Systems: Design and Implementation Conference (NSDI).

2004 ACM Workshop On Rapid Malcode (WORM)at the ACM Computer and Communications Security Conference.

Program Co-chair, 1st USENIX/ACM Networked Systems: Design and Implementation Conference (NSDI), 2004.

Program Chair, 1st Workshop On Rapid Malcode (WORM) at the ACM Computer and Communications Security Conference, 2004.

2003 ACM SIGCOMM Conference.

2003 DARPA Information Survivability Conference and Exposition.

2003 ACM Symposium on Operating System Principles (SOSP).

2002 ACM Hot Networks Workshop.

2002 ACM SIGCOMM Conference.

2001 IEEE Globecom Global Internet Symposium.

Other Professional Service Activities

Steering committee, CRA Computing Innovation Fellows, 2019-present

Steering Committee, USENIX Enigma, 2015-present

Steering Committee, ACM Conference on Computer and Communications Security (CCS), 2015-present

Member, Boeing Industry Cyber Technical Counsel, 2019-present

Distinguished Reviewer, National Security Agency Best Scientific Cybersecurity Paper Competition, 2017-present

Panelist, ISAT Study on Biocyber Security, 2020

Panelist, ACM SIGCOMM Panel on Privacy and Security, 2020

Steering Committee, ACM Computer Science and Law Symposium, 2019.

IEEE Security and Privacy, Chair Test-of-Time Award committee, 2018-2019

Carnegie Mellon University, President's Advisory Committee on the School of Computer Science, 2018.

IEEE Security and Privacy, Chair Test-of-Time Award establishment committee, 2017

Committee Member, National Academies Forum on Cyber Resilience, 2016

Computing Community Consortium, Sociotechnical Cybersecurity Workshop Steering Committee, 2016-2017

Panelist, IEEE CREDS, 2013

Panelist, International Anti-Counterfeiting Coalition (IACC) Conference, 2013

RSA Conference Panel speaker, 2013

Cryptographers Panel Speaker, RSA Conference, 2012

Panelist, RSA eFraud Global Forum, 2012

Presenter and Panelist, CCC "**The Impact of NITRD Symposium**", Washington D.C, 2012

Advisory Committee, National Science Foundation Directorate for Computer and Information Science and Engineering, 2011 (Chair, "Breakthrough Proposals" Working Group)

ACM Distinguished Dissertation Award Committee, 2011-2015 (Chair 2013)

Panelist, IEEE eCrime Symposium ethics panel, 2011

Presenter, National Academy of Science Study on Electronic Vehicle Controls and Unintended Acceleration, 2011.

Participant, ISAT Cybereconomics study, 2011.

ACM SIGCOMM Steering Committee, 2010-2011.

Member, ISAT Red Team, 2010.

Presenter, JASON Study group, "Science of Security", 2010.

Member, ISAT Black Cloud Study, 2010.

Director, Collaborative Center for Internet Epidemiology and Defenses (joint effort between UCSD and ICSI), 2004-2010.

Panel Member, ViaSAT/NSA Information Assurance Workshop on Information Assurance

Content Judge, Google NaCl Security Contest, 2009.

External Advisory Board, Stanford Programmable Open Mobile Internet (POMI) 2020 Project, 2008-

Committee Member, 3rd Santa Fe Institute Workshop on Malware, 2008.

Co-founder and Steering Committee Member, USENIX Workshop on Large-scale Exploits and Emerging Threats (LEET), 2007.

Member, GENI Science Council, 2007-2008.

Committee Member, United States Air Force Workshop on the Risks of Homogeneity, 2007.

Committee Member, 2nd Santa Fe Institute Workshop on Malware, 2007.

Committee Member, National Research Council's Computer Science and Telecommunications Board (NRC/CSTB) Study, "**Improving Cybersecurity Research in the United States"**, 2004-2007.

Committee Member, 1st Santa Fe Institute Workshop on Malware, 2006.

Committee Member, Potomac Institute for Policy Studies, "**DARPA Computer Science Futures Study**", 2006.

Committee Member, Department of Homeland Security/DARPA/Army Research Office Workshop on Botnets, 2006.

Committee Member, IETF Internet Architecture Board Meeting on Unwanted Traffic, 2006.

Panel Member, "Smoke, Mirrors and Virtualization", Network and Distributed Systems Security Symposium, 2006.

Committee Member, Department of Homeland Security/Army Research Office Workshop on Malcode Detection, 2005.

Co-Chair, DARPA ISAT Study on "Bolt-on Security", 2004.

Panelist, Microsoft Research Mesh Networking Summit, 2004.

Co-Organizer, Department of Homeland Security (DHS) Program Development Workshop on Large-scale Datasets for Network Security Testing and Evaluation, 2004.

Co-founder and Steering Committee Member, ACM Workshop on Rapid Malcode, 2003-2007.

Committee Member, CRA Conference on Grand Challenges in Information Assurance and Security, 2003.

Co-Organizer, DIMACS Workshop on Large-scale Internet Attacks, 2003.

Panelist, Institute for Pure & Applied Mathematics Workshop on Large Scale Communications Networks, 2003.

Panelist, UW-MSR-CMU Software Security Summer Institute, 2003.

Co-Organizer, ACM Workshop on Network Research: Exploration of Dimensions and Scope (NREDS), 2003.

Co-Chair, DARPA ISAT Incubator Study on Software Design for Large-Scale Embedded Systems, 2003.003.

Committee Member, DARPA Cognitive Networking Workshop, 2002.

Session Organizer, Institute for Pure & Applied Mathematics Workshop on Large Scale Communications Networks, 2002.

Committee Member, Internet2 Academic Advisory Board, 2002-2004.

Committee Member, Defense Advanced Projects Research Agency Information Science And Technology Study Group (DARPA ISAT), 2001-2004 (mandated 3 year term)

Peer Review Committee Member, Active Warden Research Project, Los Alamos National Lab, 2001-2004.

Committee Member, National Research Council's Computer Science and Telecommunication Board study, *Research Horizons in Networking*, 2001.

Executive Committee Member, IEEE Technical Committee on Operating Systems, 1999-2004.

Referee for ACM SIGCOMM, TON, SOSP, PLDI, ASPLOS, TOCS, and CCR, USENIX TC and OSDI, IEEE INFOCOM, TPDS, RTSS and others.

University/Department Service Activities

Co-chair, CSE Fellows postdoctoral award committee, 2020-2022

Recruiting committee, GPS Dean search, 2020-2021

Hiring committee, UCSD Center for Peace and Security Studies postdoctoral program, 2021.

Co-Chair, UCSD Department of Computer Science, Brochure design and content committee, 2018-2020

Co-chair, UCSD Department of Computer Science, Faculty Recruiting Committee, 2018.

Advisory Board Member, University of California (systemwide) Cyber Risk Governance Committee, 2016-present.

Ad hoc member, University of California, San Diego, Cybersecurity Governance Committee, 2017-present.

Recruiting committee, University Librarian search, 2017-2018.

Chair, UCSD Department of Computer Science, Faculty Recruiting Committee, 2007, 2014, 2020

Chair, UCSD Department of Computer Science and Engineering Publicity Committee, 2005, 2006, 2007. 2008, 2009, 2010, 2011, 2012. 2015

UCSD Department of Computer Science, Executive Committee, 2004 -- 2010.

UCSD Academic Senate Representative, 2004-2005.

UCSD Department of Computer Science and Engineering, Faculty Recruiting Committee, 2001, 2002, 2003, 2006, 2008 (*ex officio*) 2012, 2015, 2020.

UCSD Department of Computer Science and Engineering, External Relations Committee, 2002, 2003.

UCSD Department of Computer Science and Engineering, Ad Hoc Hiring Committee for Manager of Academic and External Affairs, 2001.

San Diego Network Access Point Advisory Board, Department of Computer Science and Engineering representative, 2002.

San Diego Supercomputer Center, Committee on Networking and Security Visions, 2001.

University of Washington Department of Computer Science and Engineering, Ad Hoc Committee on Commercialization in the Academic Environment, 1998-1999.

Articles

1. Jeffrey Tully, Mark Jarrett, Stefan Savage, Joshua Corman, Christian Dameff, "**Digital Defenses for Hacked Hearts: Why Software Patching Can Save Lives**", Journal of the American College of Cardiology, 72(1): 126-127, July 2018. (**note: letter**)
2. Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker and Stefan Savage, "**A Fistful of Bitcoins: Characterizing Payments Among Men with No Names**", USENIX 'login: 38(6), December 2013. . (**note: ;login not externally refereed, single editor**)
3. Harsha V. Madhyastha, John C. McCullough, George Porter, Rishi Kapoor, Stefan Savage, Alex C. Snoeren and Amin Vahdat, "**scc: Cluster Storage Provisioning Informed by Application Characteristics and SLAs**", USENIX ;login: 36(3), June 2012. (**note: ;login not externally refereed, single editor**)
4. Erik Buchanan, Ryan Roemer, Hovav Shacham and Stefan Savage, "**Return-oriented Programming: Systems, Languages and Applications**", *ACM Transactions on Information and Systems Security* (TISSEC), 15(1), March 2012.
5. Ding Yuan, Soyeon Park, Jing Zheng, Yuanyuan Zhou and Stefan Savage, "**LogEnhancer: Automatic Tailored Log Enhancement to Improve Software Diagnosability against Production-Run Failures**", *ACM Transactions on Computer Systems*, 30(1), February 2012.
6. Rik Farrow interview with Stefan Savage, "**On the Spam Payment Trail**", USENIX ;login: 35(4), August 2011. (**note: ;login not externally refereed, invited interview w/single editor**)
7. Mikhail Afanasyev, Justin Ma, Nicholas Murphy, Yoshi Kohno, Stefan Savage, Alex Snoeren and Geoffrey M. Voelker, **"Network Support for Privacy-preserving Forensic Attribution**", *Communications of the ACM* (CACM), 54(5):78-87, May 2011.
8. Justin Ma, Lawrence Saul, Stefan Savage and Geoffrey M. Voelker, "**Learning to Detect Malicious URLs**", In *ACM Transactions on Intelligent Systems and Technology* (TIST), 2(3), April 2011.
9. Diwaker Gupta, Sangmin Lee, Michael Vrable, Stefan Savage, Alex C. Snoeren, George Varghese, and Geoffrey M. Voelker, "**The Difference Engine: Harnessing Memory Redundancy in Virtual Machines**", *Communications of the ACM* (CACM), 53(10): 85-93, October 2010.
10. Michael Vrable, Stefan Savage and Geoffrey M. Voelker, "**Cumulus: Filesystem Backup to the Cloud**", *ACM Transactions on Storage*, 5(4): 1-28, December 2009.
11. Michael Vrable, Stefan Savage and Geoffrey M. Voelker, "**Cumulus: Filesystem Backup to the Cloud**", USENIX ;login: 34(4):7-13, August 2009. (**note: ;login not externally refereed, single editor**)
12. Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Vern Paxson, Geoff Voelker and Stefan Savage, **"Spamalytics: an Empirical Analysis of Spam Marketing Conversion**", *Communications of the ACM* (CACM), 52(9):99-107, September 2009.
13. Diwaker Gupta, Sangmin Lee, Michael Vrable, Stefan Savage, Alex C. Snoeren, George Varghese, and Geoffrey M. Voelker, "**The Difference Engine: Harnessing Memory Redundancy in Virtual Machines**", USENIX ;login: 34(2): 24-31, April 2009. (**note: ;login not externally refereed, single editor**)
14. Alper Mizrak, Keith Marzullo and Stefan Savage, "**Detecting Malicious Packet Losses**", *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 20(2): 191-206, February 2009.
15. Brandon Enright, Geoff Voelker, Stefan Savage, Chris Kanich and Kirill Levchenko, "**Storm: when researchers collide**", USENIX ;login: 33(4): 6-13, August 2008. (**note: ;login not externally refereed, single editor**)
16. Alper Mizrak, Stefan Savage and Keith Marzullo, "**Detecting Compromised Routers via Packet Forwarding Behavior**", *IEEE Network*, 22(2):34-39, March 2008.

17. Fan Chung, Ron Graham, Ranjita Bhagwan, Geoffrey M. Voelker and Stefan Savage, "**Maximizing Data Locality in Distributed Systems**", *Journal of Computer and System Sciences*, 72(8):1309-1306, December 2006.
18. Alper Mizrak, Yu-Chung Cheng, Keith Marzullo and Stefan Savage, "**Detecting Malicious Routers via Traffic Validation**", *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 3(3): 230-244, July 2006.
19. David Moore, Geoffrey M. Voelker and Stefan Savage, "**Inferring Internet Denial-of-Service Activity"**, *ACM Transactions on Computer Systems (TOCS)*, 24(2): 115-139, May 2006.
20. David Moore, Vern Paxson Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver, "**Inside the Slammer Worm**", *IEEE Security and Privacy*, 1(4):33-39, July 2003.
21. Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson, "**Network Support for IP Traceback**", *IEEE/ACM Transactions on Networking (ToN)*, 9(3):226-237, June 2001.
22. Stefan Savage, Neal Cardwell, David Wetherall, and Tom Anderson, "**TCP Congestion Control with a Misbehaving Receiver**", *Computer Communications Review*, 29(5):71-78, October 1999.
23. Stefan Savage, Tom Anderson, Amit Aggarwal, David Becker, Neal Cardwell, Andy Collins, Eric Hoffman, John Snell, Amin Vahdat, Geoffrey M. Voelker and John Zahorjan, "**Detour: a Case for Informed Internet Routing and Transport**", *IEEE Micro,* 19(1):50-59, January 1999.
24. Stefan Savage, Michael Burrows, Greg Nelson, Patrick Sobalvarro and Tom Anderson, "**Eraser: A Dynamic Data Race Detector for Multithreaded Programs**", *ACM Transactions on Computer Systems (TOCS)*, 15(4):391-411, November 1997.

Refereed Conference and Workshop Papers

1. Evan Johnson, Maxwell Bland, Yifei Zhu, Joshua Mason, Stephen Checkoway, Stefan Savage and Kirill Levchenko, "**Jetset: Targeted Firmware Rehosting for Embedded Systems"**, In *Proceedings of the USENIX Security Symposium,* Vancouver, B.C., Canada, August 2021.
2. Anil Yelam, Ariana Mirian, Keerthana Ganesan, Shibani Subbareddy, and Stefan Savage, **"CoResident Evil: Covert Communications in the Cloud with Lambdas"** In *Proceedings of the Web Conference (WWW)*, Ljubljana, Solvenia, April 2021.
3. Guo Li, Gautam Akiwate, Kirill Levchenko, Geoffrey M. Voelker and Stefan Savage, "**Clairvoyance: Inferring Blocklist Use on the Internet**", *Proceedings of the Passive and Active Measurement Conference (PAM)*, Brandenberg, Germany, March 2021. (44% acceptance, 33/75)
4. Evan Johnson, David Thien, Yousef Alhessi, Shravan Narayan, Fraser Brown, Sorin Lerner, Tyler McMullen, Stefan Savage, and Deian Stefan, "**Доверяй, но проверяй: SFI safety for native-compiled Wasm**", In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2021.
5. Gautam Akiwate, Raffaele Sommese, Mattijs Jonker, Ian Foster, Stefan Savage Geoffrey M. Voelker and kc Claffy, "**Unresolved Issues: Prevalence, Persistence and Perils of Lame Nameservers**", In *Proceedings of the Internet Measurement Conference (IMC)*, Pittsburgh, Pennsylvania, October 2020. (25% acceptance, 53/216)
6. Audrey Randall, Enze Liu, Gautam Akiwate, Ramakrishna Padmanabhan, Stefan Savage, Geoffrey M. Voelker and Aaron Schulman, "**TruffleHunter: Cache Sniffing Rare Domains at Large Public DNS Resolvers**", In *Proceedings of the Internet Measurement Conference (IMC)*, Pittsburgh, Pennsylvania, October 2020. 25% acceptance, 53/216)
7. Louis DeKoven, Audrey Randall, Ariana Mirian, Gautam Akiwate, Ansel Blume, Lawrence K. Saul, Aaron Shulman, Geoffrey M. Voelker and Stefan Savage, "**Measuring Security Practices and How They Impact Security**", In *Proceedings of the ACM Internet Measurement Conference (IMC)*, Amsterdam, Netherlands, October 2019. (19% acceptance, 38/197)

8. Sam Crow, Brown Farinholt, Brian Johannesmeyer, Karl Koscher, Stephen Checkoway, Stefan Savage, Aaron Schulman, Alex C. Snoeren and Kirill Levchenko, "**Triton: Software-Reconfigurable Federated Avionics Testbed**", In *Proceedings of the Workshop on Cyber Security Experimentation and Test (CSET)*, Santa Clara, CA August 2019. (31% acceptance, 19/61)

9. Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M. Voelker, Stefan Savage and Kirill Levchenko, "**Reading the Tea Leaves: A Comparative Analysis of Threat Intelligence**", In *Proceedings of the USENIX Security Symposium*, Santa Clara, CA, August 2019. (19% acceptance, 88/465)

10. Grant Ho, Asaf Cidon, Lior Gavish, Marco Schweighauser, Vern Paxson, Stefan Savage, Geoffrey M. Voelker and David Wagner, "**Detecting and Characterizing Lateral Phishing at Scale**", In *Proceedings of the USENIX Security Symposium*, Santa Clara, CA, August 2019. (19% acceptance, 88/465) (**Award paper**)

11. Ariana Mirian, Joe DeBlasio, Stefan Savage, Geoffrey M. Voelker and Kurt Thomas, "**Hack for Hire: Exploring the Emerging Market for Account Hijacking**", In Proceedings of the World Wide Web Conference, San Francisco, CA, May 2019. (18% acceptance, 225/1247)

12. Ariana Mirian, Nikunj Bhagat, Caitlin Sadowski, Adrienne Porter Felt, Stefan Savage and Geoffrey M. Voelker, "**Web Feature Deprecation: A Case Study for Chrome**", In *Proceedings of the International Conference on Software Engineering* (ICSE), Montreal, QC, Canada, May 2019. (25% acceptance, 30/118 SEIP track)

13. Stefan Savage, "**Lawful Device Access without Mass Surveillance Risk**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Toronto, Canada, October 2018. (17% acceptance 134/809)

14. Louis F Dekoven, Trevor Pottinger, Stefan Savage, Geoffrey M. Voelker and Nektarios Leontiadis, "**Following their Footsteps: Characterizing Account Automation Abuse and Defenses**", In *Proceedings of the Internet Measurement Conference (IMC)*, Chicago, IL, November 2018. (23% acceptance, 40/174)

15. Alexander Gamero-Garrido, Stefan Savage, Kirill Levchenko and Alex C. Snoeren, "**Quantifying the Pressure of Legal Risks on Third Party Vulnerability Research**", In Proceedings of the ACM Conference on Computer and Communication Security, Dallas, TX, October 2017. (18% acceptance, 151/836)

16. Joe DeBlasio, Stefan Savage, Geoffrey M. Voelker and Alex C. Snoeren, "**Tripwire: Inferring Internet Site Compromise**", In *Proceedings of the Internet Measurement Conference*, London, UK, November 2017. (23% acceptance, 42/179)

17. Louis DeKoven, Stefan Savage, Geoffrey M. Voelker and Nektarious Leontiadis, "**Malicious Browser Extensions at Scale: Bridging the Observability Gap between Web Site and Browser**", In *Proceedings of the USENIX Workshop on Cybersecurity Experimental and Test (CSET)*, Vancouver, Canada, August 2017.

18. Mortiz Contag, Guo Li, Andre Pawlowski, Flexi Domke, Stefan Savage, Kirill Levchenko, and Thorsten Holz, "**How They Did It: An Analysis of Emission Defeat Devices in Modern Automobiles**", In *Proceedings of the IEEE Symposium on Security and Privacy*, San Jose, CA, May 2017 (13% acceptance, 60/450)

19. Wilson Lian, Hovav Shacham and Stefan Savage, "**A Call to ARMs: Understanding the Costs and Benefits of JIT Spraying Mitigations**", In *Proceedings of the Network and Distributed Systems Symposium* (NDSS), San Diego, CA, February 2016. (16% acceptance, 68/463)

20. Frank Li, Zakir Durumeric, Jakub Czyz, Damon McCoy, Stefan Savage, Michael Bailey, and Vern Paxson, "**You've Got Vulnerability: Exploring Effective Vulnerability Notifications"**, In*Proceedings of the USENIX Security Symposium*, Austin, TX, August 2016. (17% acceptance, 72/423)

21. He Liu, Matthew K. Mukerjee, Conglong Li, Nicolas Feltman, George Papen, Stefan Savage, Srinivasan Seshan, Geoffrey M. Voelker, David G. Anderson, Michael Kaminsky, George Porter and Alex C. Snoeren, "**Scheduling Techniques for Hybrid Circuit/Packet Networks**", In *Proceedings of the ACM CoNEXT Conference*, Heidelberg, Germany, December 2015. (20% acceptance, 37/186)

22. Ian Foster, Jon Larson, Max Masich, Alex C. Snoeren, Stefan Savage and Kirill Levchenko, "**Security by Any Other Name: On the Effectiveness of Provider Based Email Security**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Denver, CO, October 2015. (20% acceptance, 128/646)

23. Suqi Liu, Ian Foster, Stefan Savage, Geoffrey M. Voelker and Lawrence K. Saul,, "**Who is .com? Learning to Parse WHOIS Records**", In *Proceedings of the ACM Internet Measurement Conference* (IMC), Tokyo, Japan, October 2015. (26% acceptance, 44/146)

24. Tristan Halvorson, Matthew F. Der, Ian Foster, Stefan Savage, Geoffrey M. Voelker and Lawrence K. Saul,, "**From .academy to .zone: An Analysis of the new TLD Land Rush**", In *Proceedings of the ACM Internet Measurement Conference* (IMC), Tokyo, Japan, October 2015. (26% acceptance, 44/146)

25. Neha Chachra Stefan Savage and Geoffrey M. Voelker, "**Affiliate Crookies: Characterizing Affiliate Marketing Abuse**", In *Proceedings of the ACM Internet Measurement Conference* (IMC), Tokyo, Japan, October 2015. (16% acceptance,, 12/73)

26. Kurt Thomas, Danny Yuxing Huang, David Wang, Elie Burszstein, Chris Grier, Tom Holt, Christopher Kruegel, Damon McCoy, Stefan Savage and Giovanni Vigna, "**Framing Dependencies Introduced by Underground Commoditization**", In *Proceedings of the Workshop on the Economics of Information Security (WEIS)*, Delft, The Netherlands, June 2015.

27. Wilson Lian, Hovav Shacham and Stefan Savage, "**Too LeJIT to Quite: Implementing Jit-spryaing on ARM**", In *Proceedings of the ISOC Network and Distributed Systems Symposium (NDSS)*, San Diego, February 2015. (17% acceptance rate, 15/302)

28. David Y. Wang and Matt Der and Mohammad Karami and Lawrence Saul and Damon McCoy and Stefan Savage and Geoffrey M. Voelker, "**Search + Seizure: The Effectiveness of Interventions on SEO Campaigns**", In *Proceedings of the ACM Internet Measurement Conference (IMC)*, Vancouver, BC, Canada, Nov 2014. (23% acceptance, 43/188)

29. Borbala Benko, Elie Bursztein, Daniel Margolis, Tadek Pietraszek, Andy Archer, Allan Aquino, Andreas Pitsillidis and Stefan Savage, "**Handcrafted Fraud and Extortion: Manual Account Hijacking in the Wild**", In *Proceedings of the ACM Internet Measurement Conference (IMC)*, Vancouver, BC, Canada, Nov 2014. . (23% acceptance, 43/188)

30. David Lundberg, Brown Farinholt , Edward Sullivan, Ryan Mast, Stefan Savage, Alex Snoeren and Kirill Levchenko, "**On the Security of Mobile Cockpit Information Systems**", In *Proceedings of the ACM Conference on Computer and Communication Security (CCS)*, Scottsdale, AZ, Nov 2014. (19% acceptance, 114/585)

31. Paul Pearce, Chris Grier, Vern Paxson, Vacha Dave, Damon McCoy, Kirill Levchenko, Geoffrey M. Voelker, Stefan Savage and Saikat Guha, "**Characterizing Large-Scale Click Fraud in ZeroAccess**", In *Proceedings of the ACM Conference on Computer and Communication Security (CCS),* Scottsdale, AZ, Nov 2014. (19% acceptance, 114/585)

32. Mathew F. Der, Lawrence K. Saul, Stefan Savage and Geoffrey M. Voelker, "**Knock it Off: Profiling the Online Storefronts of Counterfeit Merchandise**", In *Proceedings of the ACM SIG KDD Conference*, New York, NY, August 2014. (22% acceptance, 44/197)

33. Neha Chachra, Damon McCoy, Stefan Savage and Geoffrey M. Voelker, "**Empirically Characterizing Domain Abuse and the Revenue Impact of Blacklisting**", In *Proceedings of the Workshop on the Economics of Information Security (WEIS)*, State College, PA, June 2014. (23% acceptance, 9/40)

34. Tristan Halvorson, Kirill Levchenko, Stefan Savage and Geoffrey M. Voelker, "**XXXtortion? Inferring Registration Intent in the .XXX TLD**", In *Proceedings of the World Wide Web Conference (WWW)*, Seoul, Korea, April 2014. (13% acceptance, 84/650)

35. Danny Huang, Sarah Meiklejohn, Damon McCoy, Chris Grier, Kirill Levchenko, Alex Snoeren and Stefan Savage, "**Botcoin: Monetizing Stolen Cycles**", In *Proceedings of the ISOC Network and Distributed Security Symposium (NDSS)*, San Diego, CA, February 2014. (19% acceptance, 55/295)

36. Daniel Turner, Kirill Levchenko, Stefan Savage and Alex Snoeren, "**A Comparison of Syslog and IS-IS for Monitoring Link State**", In *Proceedings of the ACM Internet Measurement Conference* (IMC), Barcelona, Spain, October 2013. (24% acceptance, 42/178)

37. Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker and Stefan Savage, "**A Fistful of Bitcoins: Characterizing Payments Among Men with No Names**", In

Proceedings of the ACM Internet Measurement Conference (IMC), Barcelona, Spain, October 2013. (24% acceptance, 42/178)

38. Wilson Lian, Eric Rescorla, Hovav Shacham and Stefan Savage, "**Measuring the Practical Impact of DNSSEC Deployment**", In *Proceedings of the USENIX Security Symposium*, Washington D.C., August 2013. (16% acceptance, 45/277)

39. David Wang, Stefan Savage and Geoffrey M. Voelker, "**Juice: A Longitudinal Study of an SEO Campaign**", In *Proceedings of the ISOC Network and Distributed Security Symposium (NDSS)*, San Diego, CA, February 2013. (19% acceptance, 47/250)

40. Andreas Pittsilidis, Chris Kanich, Kirill Levchenko, Geoffrey M. Voelker and Stefan Savage, "**Taster's Choice: A Comparative Analysis of Spam Feeds**", In *Proceedings of the ACM Internet Measurement Conference* (IMC), Boston, MA, November 2012. (25% acceptance, 45/183)

41. Ding Yuan, Soyeon Park, Peng R. Huang, Yang R. Liu, Mihnjong M. Lee, Xiaoming Tang, Yuanyuan Zhou and Stefan Savage, "**To Log or Not to Log: Characterizing and Improving Software Error Reporting**", In *Proceedings of the ACM/USENIX Operating System Design and Implementation Conference (OSDI)*, Hollywood, CA, October 2012. (12% acceptance, 25/215)

42. Chris Grier, Lucas Ballard, Juan Caballero, Neha Chachra, Christian J. Dietrich, Kirill Levchenko, Panayiotis Mavrommatis, Damon McCoy, Antonio Nappa, Andreas Pitsillidis, Niels Provos, Zubair Rafique, Moheed Abu Rajab, Christian Rossow, Kurt Thomas, Vern Paxson, Stefan Savage and Geoffrey M. Voelker, "**Browser Exploits as a Service: The Monetization of Driveby Downloads**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Raleigh, NC, October 2012. (19% acceptance, 80/423)

43. Damon McCoy, Hitesh Dharmdasani, Christian Kreibich, Geoffrey M. Voelker and Stefan Savage, "**Priceless: The Role of Payments in Abuse-advertised Goods**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Raleigh, NC, October 2012. (19% acceptance, 80/423)

44. Damon McCoy, Andreas Pitsillidis, Grant Jordan, Nicholas Weaver, Christian Kreibich, Brian Krebs, Geoffrey M. Voelker, Stefan Savage and Kirill Levchenko, "**PharmaLeaks: Understanding the Business of Online Pharmaceutical Affiliate Programs**", *Proceedings of the USENIX Security Symposium*, Bellevue, WA, August 2012. (19% acceptance, 43/222)

45. Feng Lu, Jiaqi Zhang and Stefan Savage, "**When Good Services Go Wild: Reassembling Web Services for Unintended Pruposes**", *Proceedings of the USENIX Workshop on Hot Topics in Security (HotSec)*, Bellevue, WA, August 2012. (28% acceptance, 11/39)

46. Ross Anderson, Chris Barton, Rainer Boehme, Richard Clayton, Michel J.G. van Eeten, Michael Levi , Tyler Moore and Stefan Savage, "**Measuring the Cost of Cybercrime**", *Proceedings of the Workshop on the Economics of Information Security* (WEIS), Berlin, Germany, June 2012. (38% acceptance, 20/53)

47. Michael Vrable, Stefan Savage and Geoffrey M. Voelker, "**A Cloud-Backed File System for the Enterprise**", *Proceedings of the USENIX Conference on File and Storage Technologies* (FAST), San Jose, CA, February 2012. (19% acceptance, 26/137)

48. Harsha V. Madhyastha, John C. McCullough, George Porter, Rishi Kapoor, Stefan Savage, Alex C. Snoeren and Amin Vahdat, "**scc: Cluster Storage Provisioning Informed by Application Characteristics and SLAs**", *Proceedings of the USENIX Conference on File and Storage Technologies* (FAST), San Jose, CA, February 2012. (19% acceptance, 26/137)

49. Sushma Nagesh Bannur, Lawrence K. Saul and Stefan Savage, "**Judging a site by its content: learning the textual, structural, and visual features of malicious Web pages**", *Proceedings of the ACM Workshop on Artificial Intelligence and Security* (AISEC), Chicago, IL, October 2011.

50. Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage and Geoffrey M. Voelker, "**An Analysis of Underground Forums**", In *Proceedings of the ACM Internet Measurement Conference* (IMC), Berlin, Germany, November 2011. (19% acceptance, 42/220)

51. David Wang, Stefan Savage and Geoffrey M. Voelker, "**Cloak and Dagger: Dynamics of Web Search Cloaking**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Chicago, IL, October 2011. (14% acceptance, 60/429)

52. Daniel J. Turner, Stefan Savage and Alex Snoeren, **"On the Empirical Performance of Self-calibrating WiFi Location**", In *Proceedings of the IEEE Conference on Local Computer Networks* (LCN), Bonn, Germany, October 2011. (<30% acceptance)

53. Keaton Mowrey, Sarah Meiklejohn and Stefan Savage, "**Heat of the Moment: Characterizing the Efficacy of Thermal Camera-Based Attacks**", In *Proceedings of the USENIX Workshop on Offensive Technologies (WOOT)*, August 2011. (43% acceptance, 15/35)

54. Chris Kanich, Neha Chachra, Damon McCoy, Chris Grier, David Wang, Marti Motoyama, Kirill Levchenko, Geoffrey M. Voelker and Stefan Savage, "**No Plan Survives Contact: Experience with Cybercrime Measurement**", In *Proceedings of the USENIX Workshop on Cyber Security Experimentation and Test (CSET)*, August 2011. (40% acceptance, 12/30)

55. Chris Kanich, Nicholas Weaver, Damon McCoy, Tristan Halvorson, Christian Kreibich, Kirill Levchenko, Vern Paxson, Geoffrey M. Voelker and Stefan Savage, "**Show Me the Money: Characterizing Spam-advertised Revenue**", In *Proceedings of the USENIX Security Symposium*, August 2011. (17% acceptance, 35/204)

56. Stephen Checkoway, Damon McCoy, Danny Anderson, Brian Kantor, Hovav Shacham, Stefan Savage , Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno , "**Comprehensive Experimental Analysis of Automotive Attack Surfaces**", In *Proceedings of the USENIX Security Symposium*, August 2011. (17% acceptance, 35/204)

57. Marti Motoyama, Damon McCoy, Kirill Levchenko, Geoffrey M. Voelker, Stefan Savage, "**Dirty Jobs: The Role of Freelance Labor in Web Service Abuse**", In *Proceedings of the USENIX Security Symposium*, August 2011. (17% acceptance, 35/204)

58. Mashael AlSabah, Kevin Bauer and Ian Goldberg and Dirk Grunwald and Damon McCoy and Stefan Savage and Geoffrey M. Voelker, "**DefenestraTor: Throwing out Windows in Tor**", In *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, Waterloo, Canada, July 2011. (25% acceptance, 15/61)

59. Kirill Levchenko, Neha Chachra, Brandon Enright, Mark Felegyhazi, Chris Grier, Tristan Halvorson, Chris Kanich, Christian Kreibich, He Liu, Damon McCoy, Andreas Pitsillidis, Nicholas Weaver, Vern Paxson, Geoffrey M. Voelker and Stefan Savage, "**Click Trajectories: End-to-End Analysis of the Spam Value Chain**", In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 2011. (11% acceptance, 34/306)

60. He Liu, Kirill Levchenko, Mark Felegyhazi, Christian Kreibich, Gregor Maier, Geoffrey M. Voelker and Stefan Savage, "**On the Effects of Registrar-level Intervention**", In *Proceedings of the USENIX Workshop on Large-Scale Exploits and Emerging Threats (LEET)*, Boston, MA, March 2011. (44% acceptance; 11/25)

61. Qing Zhang, Thomas Ristenpart, Stefan Savage and Geoffrey M. Voelker, "**Got Traffic? An Evaluation of Click Traffic Providers**", In *Proceedings of the WICOW/AIRWeb Workshop on WebQuality (WebQuality)*, Hyderabad, India, March 2011.

62. Ding Yuan, Soyeon Park, Jing Zheng, Yuanyuan Zhou and Stefan Savage, "**LogEnhancer: Automatic Tailored Log Enhancement to Improve Software Diagnoseability against Production-Run Failures**", In *Proceedings of the ACM Architecture Support for Programming Languages and Operating Systems (ASPLOS) Conference*, March 2011. (21% acceptance, 32/152)

63. Andreas Pitsillidis, Yinglian Xie, Fang Yu, Martin Abadi, Geoffrey M. Voelker, and Stefan Savage, "**How to Tell and Airport from a Home: Techniques and Applications**", In *Proceedings of the ACM Workshop on Hot Topics in Networks*, Monterey, CA, October 2010. (21% acceptance, 22/104)

64. Daniel Turner, Kirill Levchenko, Alex C. Snoeren and Stefan Savage, "**California Fault Lines: Understanding the Causes and Impact of Network Failures**", In *Proceedings of the ACM SIGCOMM Conference*, New Dehli, India, August 2010. (12% acceptance, 33/276)

65. Marti Motoyamai, Kirill Levchenko, Chris Kanich, Damon McCoy, Geoffrey M. Voelker and Stefan Savage, **"Re: CAPTCHAs -- Understanding CAPTCHA Breaking from an Economic Context**", In *Proceedings of the USENIX Security Symposium*, Washington, D.C., August 2010. (15% acceptance, 30/202)

66. Mehran Bozorgi, Lawrence Saul, Stefan Savage and Geoffrey M. Voelker, "**Beyond Heuristics: Learning to Classify Vulnerabilities and Predict Exploits**", In *Proceedings of the ACM SIG KDD Conference,* pp. 105—113, Washington D.C. July 2010. (20% acceptance, 20/101)

67. Marti Motoyama, Brendan Meeder, Kirill Levchenko, Geoffrey M. Voelker and Stefan Savage, "**Measuring Online Service Availability Using Twitter**", In *Proceedings of the Workshop on Social Networks (WOSN),* Boston, MA, June 2010. (33% acceptance, 13/33)

68. Yuvraj Agarwal, Stefan Savage and Rajesh Gupta, "**Sleep-Servers: Energy Savings for Enterprise PCs by Allowing them to Sleep**", In *Proceedings of the USENIX Annual Technical Conference*, Boston, MA, June 2010. (17% acceptance, 24/141)

69. Karl Koscher, Alexei Czeskis, Fanziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham and Stefan Savage, **"Experimental Security Analysis of a Modern Automobile**", In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 2010. (11% acceptance, 26/237)

70. Qing Zhang, John McCullough, Justin Ma, Navil Schear, Michael Vrable, Geoffrey M. Voelker, Alex C. Snoeren and Stefan Savage, "**Neon: System Support for Derived Data Management**", In *Proceedings of the ACM Conference on Virtual Execution Environments (VEE)*, Pittsburgh, PA, March 2010. (27% acceptance 14/51).

71. Andreas Pitsillidis, Kirill Levchenko, Chris Kanich, Christian Kreibich, Brandon Enright, Vern Paxson, Geoff Voelker and Stefan Savage, "**Botnet Judo: Attacking Spam with Itself**", In *Proceedings of the ISOC Network and Distributed Security Symposium (NDSS)*, San Diego, CA, February 2010. (15% acceptance, 24/156)

72. Tom Ristenpart, Eran Tromer, Hovav Shacham and Stefan Savage, "**Hey, You, Get off of my Cloud! Exploring Information Leakage in Third-Party Compute Clouds**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Chicago IL, November 2009. (18% acceptance, 58/315)

73. Scott Yilek, Eric Rescorla, Hovav Shacham, Brandon Enright and Stefan Savage, **"When Private Keys are Public: Understanding the Debian OpenSSL Debacle**", In *Proceedings of the USENIX Internet Measurement Conference*, Chicago, IL, November 2009. (22% acceptance, 41/13)

74. Justin Ma, Lawrence K. Saul, Geoffrey M. Voelker and Stefan Savage, **"Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs**", In *Proceedings of the ACM SIGKDD Conference*, Paris, France, July 2009. (28% acceptance, 34/122)

75. Justin Ma, Lawrence Saul, Stefan Savage and Geoffrey M. Voelker, "**Identifying Suspicious URLs: An Application of Large-scale Online Learning**", In *Proceedings of the International Conference on Machine Learning (ICML)*, Montreal, Quebec, June 2009. (26% acceptance, 160/595)

76. Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Vern Paxson, Geoff Voelker and Stefan Savage, "**Spamcraft: an Inside Look at Spam Campaign Orchestration**", In *Proceedings of the USENIX Workshop on Large-Scale Exploits and Emerging Threats (LEET)*, Boston, MA, April 2009 (36% acceptance, 9/25)

77. Michael Vrable, Stefan Savage and Geoffrey M. Voelker, "**Cumulus: Filesystem Backup to the Cloud**", In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST)*, pp. 225-238, San Francisco, CA, February 2009. (23% acceptance, 23/102)

78. Diwaker Gupta, Sangmin Lee, Michael Vrable, Stefan Savage, Alex C. Snoeren, George Varghese, and Geoffrey M. Voelker, "**The Difference Engine: Harnessing Memory Redundancy in Virtual Machines**", In *Proceedings of the ACM/USENIX Operating System Design and Implementation Conference (OSDI)*, pp. 309-322, San Diego, CA, December 2008. (13% acceptance, 26/193) **(Jay Lepreau Best Paper Award)**

79. Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Vern Paxson, Geoff Voelker and Stefan Savage, "**Spamalytics: an Empirical Analysis of Spam Marketing Conversion**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pp. 3-14, Alexandria, VA, October 2008. (18% acceptance, 51/281)

80. Benjamin Laxton, Kai Wang and Stefan Savage, "**Reconsidering Physical Key Secrecy: Teleduplication via Optical Decoding**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pp. 469-478, Alexandria, VA, October 2008. (18% acceptance, 51/281)

81. Erick Buchanan, Ryan Roemer, Hovav Shacham and Stefan Savage, "**When Good Instructions Go Bad: Generalizing Return-oriented Programming to the SPARC**", In *Proceedings of the ACM Conference on*

Computer and Communications Security (CCS), pp. 27-38, Alexandria, VA, October 2008. (18% acceptance, 51/281)

82. Kirill Levchenko, Geoffrey M. Voelker, Mohan Paturi and Stefan Savage, "**XL: An Efficient Network Routing Protocol**", In *Proceedings of ACM SIGCOMM*, pp. 15-26, Seattle, WA, August 2008. (13% acceptance, 37/288)

83. Christian Kriebeck, Chris Kanich, Kirill Levchenko, Brandon Enright, Vern Paxson, Geoff Voelker and Stefan Savage, "**On the Spam Campaign Trail**", In *Proceedings of the USENIX Workshop on Large-Scale Exploits and Emerging Threats (LEET)*, San Francisco, CA, April, 2008. (32% acceptance, 11/34)

84. Brandon Enright, Chris Kanich, Kirill Levchenko, Geoff Voelker and Stefan Savage, "**Understanding the Heisenbot Problem**", In Proceedings of the *USENIX Workshop on Large-Scale Exploits and Emerging Threats (LEET)*, San Francisco, CA, April, 2008. (32% acceptance, 11/34)

85. Jason Franklin, Adrian Perrig, Vern Paxson and Stefan Savage, "**An Inquiry into the Nature and Causes of the Wealth of Internet Miscreants**", In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Alexandria, VA, October 2007. (18% acceptance, 55/303)

86. Chris Fleizach, Geoffrey M. Voelker and Stefan Savage, "**Slicing Spam with Occam's Razor**", In *Proceedings of the Conference on E-mail and Anti-Spam (CEAS)*, Mountain View, CA, August 2007. (32% acceptance, 25/79)

87. Yu-Chung Cheng, Mikhail Afanasyev, Peter Benko, Patrick Verkaik, Jennifer Chiang, Alex C. Snoeren, Stefan Savage, and Geoffrey M. Voelker, "**Automating Cross-Layer Diagnosis of Enterprise Wireless Networks**", In *Proceedings of the ACM SIGCOMM Conference*, Kyoto, Japan, August 2007 (14% acceptance, 35/258)

88. David S. Anderson, Chris Fleizach, Stefan Savage and Geoffrey M. Voelker, "**Spamscatter: Characterizing Internet Scam Hosting Infrastructure**", In *Proceedings of the USENIX Security Symposium*, Boston, MA, August 2007. (12% acceptance, 23/187)

89. Justin Ma, Kirill Levchenko, Christian Kriebeck, Stefan Savage and Geoffrey M. Voelker, "**Automated Protocol Inference: Unexpected Means of Identifying Protocols**", In *Proceedings of the ACM/USENIX Internet Measurement Conference (IMC)*, Rio de Janerio, Brazil, October 2006. (22% acceptance, 34/155 – one of 19 long papers)

90. Justin Ma, John Dunagan, Helen J. Wang, Stefan Savage and Geoffrey M. Voelker, "**Finding Diversity in Remote Code Injection Exploits**", In *Proceedings of the ACM/USENIX Internet Measurement Conference (IMC)*, Rio de Janerio, Brazil, October 2006. (22% acceptance, 34/155 – one of 19 long papers)

91. Yu-Chung Cheng, John Bellardo, Peter Benko, Alex Snoeren, Geoffrey M. Voelker and Stefan Savage, "**Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis**", In *Proceedings of the ACM SIGCOMM Conference*, Pisa, Italy, September 2006 (12% acceptance, 37/298).

92. Martin Casado, Tal Garfinkel, Weidong Cui, Vern Paxson, and Stefan Savage, "**Opportunistic Measurement: Spurious Network Events as a Light in the Darkness**", In *Proceedings of the 4th ACM Workshop on Hot Topics in Networks (HotNets)*, College Park, MD, November 2005. (20% acceptance, 20/102)

93. Justin Ma, Geoffrey M. Voelker and Stefan Savage, "**Self-stopping Worms**", In *Proceedings of the ACM Workshop on Rapid Malcode (WORM)*, Fairfax, VA, November 2005. (25% acceptance, 9/37)

94. Michael Vrable, Justin Ma, Jay Chen, David Moore, Erik Vadekeift, Alex S. Snoeren, Geoffrey M. Voelker and Stefan Savage, "**Scalability, Fidelity and Containment in the Potemkin Virtual Honeyfarm**", In *Proceedings of the ACM Symposium on Operating System Principles (SOSP)*, Brighton, UK, October 2005. (13% acceptance, 20/155)

95. Alper Mizrak, Yu-Chung Cheng, Keith Marzullo and Stefan Savage, "**Fatih: Detecting Malicious Routers via Traffic Validation**", In *Proceedings of the IEEE Symposium on Dependable Systems and Networks (DSN)*, Yokohama, Japan, June 2005. (24% acceptance, 78/319) (**Award paper**)

96. Ishwar Ramani and Stefan Savage, "**SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks**", In *Proceedings of the IEEE Infocom Conference*, Miami, FL, March 2005. (17% acceptance, 244/1419)

97. Sumeet Singh, Cristian Estan, George Varghese and Stefan Savage, "**Automatic Worm Fingerprinting**", In *Proceedings of the ACM/USENIX Operating System Design and Implementation Conference (OSDI),* San Francisco, CA, December 2004. (14% acceptance, 27/193)

98. Yu-Chung Cheng, Urs Hoelzle, Neal Cardwell, Stefan Savage and Geoffrey M. Voelker, "**Monkey See, Monkey Do: A Tool for TCP Tracing and Replaying**", In *Proceedings of the USENIX Technical Conference*, Boston, MA, June 2004. (13% acceptance, 21/164)

99. Ranjita Bhagwan, Kiran Tati, Yu-Chung Cheng, Stefan Savage and Geoffrey M. Voelker, "**Total Recall: System Support for Automated Availability Management**", In *Proceedings of the USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, March 2004. (23% acceptance; 27/118)

100. Renata Teixeira, Keith Marzullo, Stefan Savage and Geoffrey M. Voelker, "**In Search of Path Diversity in ISP Networks**", In *Proceedings of the USENIX/ACM Internet Measurement Conference (IMC)*, Miami, FL, October 2003. (30% acceptance; 33/109)

101. Cristian Estan, Stefan Savage and George Varghese, "**Automatically Inferring Patterns of Resource Consumption In Network Traffic**", In *Proceedings of ACM SIGCOMM Conference*, Karlsruhe, Germany, August 2003. (10% acceptance; 33/319)

102. John Bellardo and Stefan Savage, "**802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions**", In *Proceedings of the USENIX Security Symposium*, Washington D.C, August 2003. (17% acceptance; 21/125)

103. Alper Mizrak, Yu-Chung Cheng, Vineet Kumar and Stefan Savage, , "**Structured Superpeers: Leveraging Heterogeneity to Provide Constant-Time Lookup**", In *Proceedings of the IEEE Workshop on Internet Applications*, San Jose, CA, June 2003. (19% acceptance; 10/53 full papers)

104. Flavio Junqueira, Ranjita Bhagwan, Keith Marzullo, Stefan Savage and Geoffrey M. Voelker, "**The Phoenix Recovery System: Rebuilding from the Ashes of an Internet Catastrophe**", In *Proceedings of the USENIX/IEEE Workshop on Hot Topics in Operating Systems*, Lihue, Hawaii, May 2003. (21% acceptance; 31/150; one of 13% invited to present; 19/150)

105. David Moore, Colleen Shannon, Geoffrey M. Voelker and Stefan Savage, "**Internet Quarantine: Requirements for Containing Self-Propagating Code**", In *Proceedings of the IEEE Infocom Conference*, San Francisco, CA, April 2003. (20% acceptance; 224/1100)

106. Ranjita Bhagwan, Stefan Savage and Geoffrey M. Voelker, "**Understanding Availability**", In *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS)*, Berkeley, CA, February 2003. (16% acceptance; 26/165)

107. John Bellardo and Stefan Savage, "**Measuring Packet Reordering**", In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, October, 2002. (26% acceptance; 16/62 full papers)

108. David Ely, Neil Spring, David Wetherall, Stefan Savage and Tom Anderson, "**Robust Congestion Signaling**", In *Proceedings of the International Conference on Network Protocols*, pages 332-341, Riverside, CA, November 2001. (23% acceptance; 36/157) (**Selected Paper**)

109. David Moore, Geoffrey M. Voelker and Stefan Savage, "**Inferring Internet Denial-of-Service Activity**", In *Proceedings of the USENIX Security Symposium*, Washington, D.C., August 2001. (29% acceptance; 24/83) (**Best Paper**)

110. David Ely, Stefan Savage and David Wetherall, "**Alpine: A User-Level Infrastructure for Network Protocol Development**", In *Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS)*, pages 171-183, San Francisco, CA, March 2001. (27% acceptance; 19/70)

111. Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson, "**Practical Network Support for IP Traceback**", In *Proceedings of the ACM SIGCOMM Conference*, pages 295-306, Stockholm, Sweden, August 2000. (11% acceptance; 26/238) (**Selected Paper**)

112. Neal Cardwell, Stefan Savage and Tom Anderson, "**Modeling TCP Latency**", In *Proceedings of the IEEE Infocom Conference*, pages 1742-1751, Tel-Aviv, Israel, March 2000. (26% acceptance; 192/735)

113. Amit Aggarwal, Stefan Savage and Tom Anderson, "**Understanding the Performance of TCP Pacing**", In *Proceedings of the IEEE Infocom Conference*, pages 1157-1165, Tel-Aviv, Israel, March 2000. (26% acceptance; 192/735)

114. Stefan Savage, "**Sting: a TCP-based Network Measurement Tool**", In *Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS)*, pages 71-79, Boulder, Co, October 1999. (33% acceptance; 22/67) (**Best Student Paper**)

115. Stefan Savage, Andy Collins, Eric Hoffman, John Snell, and Tom Anderson, "**The End-to-end Effects of Internet Path Selection**", In *Proceedings of the ACM SIGCOMM Conference*, pages 289-299, Cambridge, MA, September 1999. (13% acceptance; 24/190)

116. Stefan Savage, Neal Cardwell and Tom Anderson, "**The Case for Informed Transport Protocols**", In *Proceedings of the IEEE Workshop on Hot Topics in Operating Systems (HotOS)*, pages 58-63, Rio Rico, AZ, March 1999.

117. Stefan Savage, Michael Burrows, Greg Nelson, Patrick Sobalvarro and Tom Anderson, "**Eraser: A Dynamic Data Race Detector for Multithreaded Programs**", In *Proceedings of the ACM Symposium on Operating System Principles (SOSP)*, pages 27-37, Saint-Malo, France, October 1997. (21% acceptance; 23/109) (**Award Paper**)

118. Wilson Hsieh, Marc Fiuczynski, Charles Garrett, Stefan Savage, David Becker and Brian Bershad, "**Language Support for Extensible Operating Systems**", In *Proceedings of the Workshop on Compiler Support for System Software*, pages 127-133, Tucson, AZ, February 1996.

119. Emin Gun Sirer, Stefan Savage, Przemyslaw Pardyak, Greg Defouw, Mary Ann Alapat and Brian Bershad, "**Writing an Operating System with Modula-3**", In *Proceedings of the Workshop on Compiler Support for System Software*, pages 134-140, Tucson, AZ, February 1996.

120. Stefan Savage and John Wilkes, "**AFRAID – A Frequently Redundant Array of Independent Disks**", In *Proceedings of the USENIX Technical Conference*, pages 27-39, San Diego, CA, January 1996. (22% acceptance; 28/130) (**Best Student Paper**)

121. Brian Bershad, Stefan Savage, Przemyslaw Pardyak, Emin Gun Sirer, David Becker, Marc Fiuczynski, Craig Chambers and Susan Eggers, "**Extensibility, Safety and Performance in the SPIN Operating System**", In *Proceedings of the ACM Symposium on Operating System Principles (SOSP),* pages 267-284, Copper Mountain, CO, December 1995. (26% acceptance; 22/84)

122. Brian Bershad, Stefan Savage, Przemyslaw Pardyak, David Becker, Marc Fiuczynski and Emin Gun Sirer, "**Protection is a Software Issue**", In *Proceedings of the IEEE Workshop on Hot Topics in Operating Systems*, pages 62-65, Orcas Island, WA, May 1995.

123. Brian Bershad, Craig Chambers, Susan Eggers, Chris Maeda, Dylan McNamee, Przemyslaw Pardyak, Stefan Savage and Emin Gun Sirer, "**SPIN – An Extensible Microkernel for Application-specific Operating System Services**", In *Proceedings of the ACM SIGOPS European Workshop*, Dagstuhl, Germany, September 1994.

124. Clifford Mercer, Stefan Savage and Hideyuki Tokuda, "**Processor Capacity Reserves: Operating System Support for Multimedia Applications**", In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pages 90-99, Boston, MA, May 1994.

125. Clifford Mercer, Stefan Savage and Hideyuki Tokuda, "**Processor Capacity Reserves: An Abstraction for Managing Processor Usage**", In *Proceedings of the IEEE Workshop on Workstation Operating Systems*, pages 129-134, Napa, CA, October 1993.

126. Stefan Savage and Hideyuki Tokuda, "**Real-Time Mach Timers: Exporting Time to the User**", In *Proceedings of the USENIX Mach Symposium*, pages 111-118, Santa Fe, NM, April 1993.

Other Papers and Reports

1. Stefan Savage, "**In Planning Digital Defenses, the Biggest Obstacle Is Human Ingenuity**", *New York Times* Science Section, page D10, December 6th, 2011.

2. Ge Wang, He Liu, Sebastian Becerra, Kai Wang, Serge Belongie, Hovav Shacham and Stefan Savage, "**Verilogo: Proactive Phishing Detection via Logo Recognition**", Technical Report CS2011-0969, August 2011.
3. Stefan Savage And Fred B. Schneider, "**Security is Not a Commodity: The Road Forward for Cybersecurity Research**", Computing Research Initiatives for the 21st Century, Computing Community Consortium, February 2009.
4. Yu-Chung Cheng, Mikhail Afanasyev, Patrick Verkaik, Peter Benko, Jennifer Chiang, Alex C. Snoeren, Stefan Savage and Geoffrey M. Voelker, "**The Shaman Automatic 802.11 Wireless Diagnosis System**", *Technical Report CS2010-0959*, Mary 2010.
5. Yuvraj Agarwal, Stefan Savage and Rajesh Gupta, "**SleepServer: Energy Savings for Enterprise PCs By Allowing them to Sleep**", *Technical Report CS2009-0953*, November 2009.
6. Mikhail Afanasyev, Tadayoshi Kohno, Justin Ma, Nick Murphy, Stefan Savage, Alex C. Snoern and Geoffrey M. Voelker, "**Network Support for Privacy-Preserving Forensic Attribution**", *Technical Report CS2009-0940*, March 2009.
7. Qing Zhang, John McCullough, Justin Ma, Nabil Schear, Michael Vrable, Amin Vahdat, Alex C. Snoeren, Geoffrey M. Voelker and Stefan Savage, "**Neon: System Support for Derived Data Management**", *Technical Report CS2008-0934*, December 2008.
8. Michael Vrable, Stefan Savage and Geoffrey M. Voelker, "**Cumulus: Filesystem Backup to the Cloud**", *Technical Report CS2008-0927*, July 2008.
9. Alper Mizrak, Keith Marzullo and Stefan Savage, "**Detecting Compromised Routers via Packet Forwarding Behavior**", *Technical Report CS2007-0899*, June 2007.
10. Chris Fleizach, Geoffrey M. Voelker and Stefan Savage, "**Slicing Spam with Occam's Razor**", *Technical Report CS2007-0893*, June 2007.
11. Alper Mizrak, Keith Marzullo and Stefan Savage, "**Detecting Malicious Packet Losses**", *Technical Report CS2007-0889*, March 2007.
12. David S. Anderson, Chris Fleizach, Stefan Savage and Geoffrey M. Voelker, "**Spamscatter: Characterizing Internet Scam Hosting Infrastructure**", *Technical Report CS2007-0887*, March 2007.
13. Yu-Chung Cheng, Mikhail Afanasyev, Peter Benko, Patrick Verkaik, Jennifer Chiang, Alex C. Snoeren, Stefan Savage, and Geoffrey M. Voelker, "**Automating Cross-Layer Diagnosis of Enterprise Wireless Networks**", *Technical Report CS2007-0885*, March 2007.
14. Justin Ma, Kirill Levchenko, Christian Kreibich, Stefan Savage and Geoff Voelker, "**Automatic Protocol Inference: Unexpected Means of Identifying Protocols**", *Technical Report CS2006-0850*, February 2006.
15. Yu-Chung Cheng, John Bellardo, Peter Benko, Alex Snoeren, Geoff Voelker and Stefan Savage, "**Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis**", *Technical Report CS2006-0849*, February 2006.
16. Sumeet Singh, Cristian Estan, George Varghese and Stefan Savage, "**Accuracy Bounds for the Scaled Bitmap Data Structure",** *Technical Report CS2005-0819*, March 2005.
17. David Moore, Colleen Shannon, Geoffrey M. Voelker and Stefan Savage, "**Network Telescopes",** *Technical Report CS2004-0795*, July 2004.
18. Alper Mizrak, Keith Marzullo and Stefan Savage, "**Fault-Tolerant Forwarding in the Face of Malicious Routers**", In *Proceedings of the International Workshop on Future Directions in Distributed Computing*, Bertinoro, Italy, June 2004. (invited paper)
19. Alper Mizrak, Keith Marzullo and Stefan Savage, "**Detecting Malicious routers",** *Technical Report CS2004-0789*, May 2004.
20. Ishwar Ramani and Stefan Savage, "**SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks**", *UCSD Technical Report CS2004-0787*, May 2004.

21. Fan Chung, Ronald Graham, Ranjita Bhagwan, Stefan Savage and Geoffrey M. Voelker, "**A Near-Optimal Algorithm for a Locality-Maximizing Placement Problems**", *UCSD Technical Report CS2004-0777*, January 2004.

22. Sumeet Singh, Cristian Estan, George Varghese and Stefan Savage, "**Earlybird: Automated Worm Fingerprinting**", In *Proceedings of the ACM Symposium on Operating System Principles*, Bolton Landing, NY, October 2003. (poster, award finalist)

23. Ranjita Bhagwan, Kiran Tati, Yu-Chung Cheng, Stefan Savage and Geoffrey M. Voelker, "**TotalRecall: System Support for Automated Availability Management**", In *Proceedings of the ACM Symposium on Operating System Principles*, Bolton Landing, NY, October 2003. (poster)

24. Sumeet Singh, Cristian Estan, George Varghese and Stefan Savage, "**The Earlybird System for Real-Time Detection of Unknown Worms**", *CS2003-0761*, August 2003

25. Renata Teixeira, Keith Marzullo, Stefan Savage and Geoffrey M. Voelker, "**Characterizing and Measuring Path Diversity of Internet Topologies**", In *Proceedings of the ACM SIGMETRICS Conference*, San Diego, CA, June 2003. (short paper & poster)

26. Abhishek Agrawal, Douglas Brown, Aditya Ojha and Stefan Savage, "Bucking **Free-Riders: Distributed Accounting and Settlement in Peer-to-Peer Networks***", CS2003-0751*, June 2003

27. Cristian Estan, Stefan Savage and George Varghese, "**Automatically Inferring Patterns of Resource Consumption in Network Traffic**", *UCSD Technical Report CS2003-0746*, June 2003

28. Sumeet Singh, Cristian Estan, George Varghese and Stefan Savage, "**Real-time Detection of Known and Unknown Worms**", *UCSD Technical Report CS2003-0745*, May 2003.

29. Flavio Junqueira, Ranjita Bhagwan, Keith Marzullo, Stefan Savage and Geoffrey M. Voelker, "**The Phoenix Recovery System: Rebuilding from the Ashes of an Internet Catastrophe**", *UCSD Technical Report CS2003-0732*, January 2003.

30. Cristian Estan, Stefan Savage and George Varghese, "**Automated Measurement of High-Volume Traffic Clusters**", In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, October, 2002. (short paper)

31. Ranjita Bhagwan, Stefan Savage and Geoffrey M. Voelker, "**Replication Strategies for Highly Available Peer-to-Peer Storage Systems**", *UCSD Technical Report CS2002-0726*, November 2002.

32. Ranjita Bhagwan, David Moore, Stefan Savage and Geoffrey M. Voelker, "**Replication Strategies for Highly Available Peer-to-Peer Storage**", In *Proceedings of the International Workshop on Future Directions in Distributed Computing*, Bertinoro, Italy, June 2002. (invited paper)

33. Stefan Savage, **"Protocol Design in an Uncooperative Internet"**; Ph.D. Thesis, Department of Computer Science and Engineering, University of Washington, January 2002.

34. David Patterson, David Clark, Anna Karlin, Jim Kurose, Edward Lazowska, David Liddle, Derek McAuley, Vern Paxson, Stefan Savage and Ellen Zegura, **Looking Over the Fence at Networks: A Neighbor's View of Networking Research***, National Academies Press*, 2001.

35. Eric J. Anderson, Thomas E. Anderson, Steven D. Gribble, Anna R. Karlin, Stefan Savage, "**A Quantitative Evaluation of Traffic-Aware Routing Strategies",** In *Proceedings of the 2001 ACM SIGCOMM Conference*, San Diego, CA, August 2001. (poster)

36. Stefan Savage, Tom Anderson, Amit Aggarwal, David Becker, Neal Cardwell, Andy Collins, Eric Hoffman, John Snell, Amin Vahdat, Geoffrey M. Voelker and John Zahorjan, "**Detour: a Case for Informed Internet Routing and Transport**", *University of Washington Technical Report UW-CSE-98-10-05*, October 1998.

37. Wilson Hsieh, Marc Fiuczynski, Charles Garrett, Stefan Savage, David Becker and Brian Bershad, "**Language Support for Extensible Operating Systems**", *University of Washington Technical Report UW-CSE-95-11-02*, November 1995.

38. Emin Gun Sirer, Stefan Savage, Przemyslaw Pardyak, Greg Defouw, Mary Ann Alapat and Brian Bershad, "**Writing an Operating System with Modula-3**", *University of Washington Technical Report UW-CSE-95-11-03*, November 1995.

39. Brian Bershad, Craig Chambers, Susan Eggers, Chris Maeda, Dylan McNamee, Przemyslaw Pardyak, Stefan Savage and Emin Gun Sirer, "**SPIN – An Extensible Microkernel for Application-specific Operating System Services**", *University of Washington Technical Report UW-CSE-94-03-03*, March 1994.

40. Clifford Mercer, Stefan Savage and Hideyuki Tokuda, "**Processor Capacity Reserves for Multimedia Operating Systems**", *Carnegie Mellon Technical Report, CMU-CS-93-157*, May 1993.

Invited Presentations

"Cyberphysical System Security", Invited talk, ISAT Study on Bio-Cyber Security, 2020.

"Experiences in Tracking Cybercrime", Invited talk, Emerging Tech and Partnerships for Counterproliferation Workshop, National Academies of Science, 2020.

"Cybersecurity as an evidence-based discipline", Keynote, NSF Cybersecurity Summit, 2019.

"Modern Automotive Vulnerabilities: Causes, Disclosures and Outcomes", Distinguished Lecture, University of Massachusetts, Amherst, 2019.

"Modern Automotive Vulnerabilities: Causes, Disclosures and Outcomes", Distinguished Lecture, University of Massachusetts, Amherst, 2019.

"Automotive Cybersecurity: Research driving Practice", San Diego Chapter Computer Science Teachers Association, 2019.

"Modern Automotive Vulnerabilities: Causes, Disclosures and Outcomes", Distinguished Lecture, University of California, Riverside, 2019.

"Lawful Device Access without Mass Surveillance Risk", Qualcomm Invited Lecture, 2018.

"Lawful Device Access without Mass Surveillance Risk", Cornell Tech Distinguished Lecture, 2018.

"**Cybersecurity in the Age of the Internet**", Fleet Science Center "Suds & Science" series, 2018.

"**Driving Under the Influence: Cybersecurity of Modern Automobiles**", Invited Talk, Bohemian Club, 2018.

"**Modern Automotive vulnerabilities: Causes, Disclosures and Outcomes"**, Invited Talk, UCSD Oshur Center, 2018.

"**Modern Automotive vulnerabilities: Problems, Causes and Outcomes**", Qualcomm Mobile Security Summit, 2018.

"**Ethical Issues in Cybersecurity Research**", Design@Large UCSD Lecture series, 2018.

"**Modern Automotive vulnerabilities: Problems, Causes and Outcomes**", NSF Distinguished Lecture, 2018.

"**Fighting cybercrime by Understanding Economics**", UCSD Town and Gown, 2018.

"**Experiences with Automotive Security: Vulnerabilities, Challenges and Outcomes**", JASON Autonomous Driving study, 2017.

"**Experiences with Automotive Security: Vulnerabilities, Challenges and Outcomes**", Microsoft Research, 2016.

"**Understanding the Effects of Security Interventions**", Microsoft Research, 2016.

"**Modern Automotive Vulnerabilities: Causes, Disclosures and Outcomes**", USENIX Enigma Conference, 2016.

"**Abuse Economics: Spam, Drugs, Affiliate Programs Payments and Interventions**", Google Security Summit, 2015.

"**Experiences for Automotive Security: Vulnerabilities, Causes and Challenges**", Stanford Cyber BootCamp for Congressional Staffers, 2015.

"**Data Driven Vulnerability Assessment**", CMU/SwissRe Workshop on Cyber Risk, 2015.

"**Demonetizing Advertising Based eCrime**", Michigan State University Interdisciplinary Conference on Cybercrime, 2015.

"**Cybersecurity: Challenges and Opportunities**", National Research Council Symposium on Continuing Innovation in Information Technology, 2014.

"**On the Effects of Civil Interventions**", Interpol/Cymru Underground Economy 2014,

"**Experiences from ten years of data-driven security research**", UW/MSR Security Analytics Workshop, 2014.

"**Abuse Economics: Spam, Drugs, Affiliate Programs, Payments and Interventions**", Google Abuse Summit, Mountain View, 2014.

"**Demonetizing advertising-based eCrime**", Microsoft Research (NYC), 2014.

"**Automotive Cyber Threats**", ITSG, 2014.

"**Buying Drugs for Science: Addressing the Cybercrime through Economics**", Andreesen Horowitz Academic Roundtable, 2013.

"**Buying Drugs for Science: Addressing the Economics of Cybercrime**", UCSD Gordon Center Think Tank, 2013.

"**Bitcoin Abuse: from Botnets to Payments**", Interpol/Cymru Underground Economy, 2013.

"**Experiences with Automotive Security: Vulnerabilities, Causes and Challenges**", IEEE NTS (Boeing) Invited Talk, 2013.

"**Buying Drugs for Science: Addressing the Economics of Cybercrime**", Osher Lifelong Learning Institute, 2013.

"**Buying Drugs for Science: Addressing the Economics of Cybercrime**", Microsoft Faculty Summit, 2013.

"**Experiences with Automotive Security: Vulnerabilities, Causes and Challenges**", ESCAR USA, 2013.

"**Buying Drugs for Science: Addressing the Economics of Cybercrime**", Department of Justice Invited Talk, 2013.

"**The Role of Payments in Monetizing Abusive Advertising**", Microsoft Digital Crime Conference, 2013.

"**Experiences with Automotive Security: Vulnerabilities, Causes and Challenges**", DARPA HACMS PI Meeting, 2013.

"**Experiences with Automotive Security: Vulnerabilities, Causes and Challenges**", NSF Future Directions in Formal Methods keynote, 2013.

"**Reflecting on Automotive Cybervulnerabilities**", DoD CyberVehicle Security Workshop, 2012.

"**Buying Drugs for Science: Addressing Cybercrime from an Economic Perspective**", Yahoo Big Thinker Talk, 2012.

"**Payment Processing for Advertising-based E-crime**", Interpol/Cymru Underground Economy, 2012.

"**Buying Drugs for Science: Addressing Cybercrime from an Economic Perspective**", Carnegie Mellon, 2012.

"**Inside the Economics of Illicit Online Pharma: Demand, Marketing and Payments**", NCFTA Pharma Conference, 2012.

"**Buying Drugs for Science: Understanding the Economics of Cybercrime"**, Facebook, 2012.

"**Experimental Security Analysis of a Modern Automobile**", JASON invited talk, 2012.

"**Looking before you Leap: the argument for data-driven security**", Distinguished Lecture, University of California, Berkeley, 2012.

"**Successes and Challenges of Computer Security Research**", CCC's "The Impact of NITRD" Symposium, 2012.

"**Looking before you Leap: the argument for data-driven security**", Distinguished Lecture, University of Illinois 2012.

"**Looking before you Leap: the argument for data-driven security**", Distinguished Lecture, Stanford University 2012.

"**Why the hard problem of computer security needs the soft sciences**", Invited talk, NSF Washington Area Trustworthy Computing Hour (WATCH), 2011.

"**The Spam Ecosystem: Demand, Revenue and Payments**", Trend Micro, 2011.

"**Monetizing the botnet ecosystem**", Panel session, Microsoft Digital Crime Consortium (DCC), 2011

"**The Spam Ecosystem: Demand, Revenue and Payments**", Interpol/Cymru Underground Economy, 2011.

"**Looking before you leap: an argument for data-driven security**", Distinguished Lecture, University of Maryland, 2011.

"**Looking before you leap: an argument for data-driven security**", MIT 2011.

"**Experimental Security Analysis of a Modern Automobile**", Google, 2011.

"**Follow the Money: Understanding the Spam Value Chain**", UCSD Campus LISA, 2011.

"**Experimental Security Analysis of a Modern Automobile**", Distinguished Lecture, University of British Columbia, 2011.

"**Beyond Heuristics: Learning to Classify Vulnerabilities and Predict Exploits**", HP Labs, 2011.

"**Analyzing and Understanding Online Criminals and Behavior** ", In-Q-Tel, 2011.

"**Experimental Security Analysis of a Modern Automobile**", In-Q-Tel, 2011. (talk given w/Yoshi Kohno)

"**The Value Chain of Spam-advertised Goods"**, Galois Systems, 2011.

"**An Agenda for Empirical Cybercrime Research**", Keynote USENIX Annual Technical Conference, 2011.

"**Experimental Security Analysis of a Modern Automobile**", Qualcomm Security Symposium, 2011.

"**The Value Chain of Spam-advertised Pharmacies"**, US Food and Drug Administration, 2011.

"**Experimental Security Analysis of a Modern Automobile**", National Academy Science, Study group on Unintended Acceleration, 2011. (talk given w/Yoshi Kohno)

"**Security and Criminal Economics**", ISAT Study group on Security Economics, 2011.

"**An Agenda for Empirical Cybercrime Research**", Google, 2011.

"**Drug Dealing for Science and Other Stories…**", Duke University, 2010.

"**Understanding Modern Cybercrime: Technology and Economics**", BAE Systems, 2010.

"**An Agenda for Empirical Cybercrime Research**", Microsoft, 2010.

"**Defining the Threat: Technical/Commercial Drivers**", Lawrence Livermore National Lab, 2010.

"**Beyond Heuristics: Learning to Classify Vulnerabilities and Predict Exploits"**, invited talk USENIX Workshop On Offensive Technology (WOOT), 2010.

"**Re: CAPTCHAs: Understanding CAPTCHA Solving from an Economic Context**", Google, 2010.

"**Computer Security: Science, Engineering, Economics and a bit of Reality**", JASON study group, 2010.

"**Cybereconomics**", Cyber Leap Year panel at IEEE Security and Privacy, 2010.

"**The Underground Economy: A broad overview**", Invited Talk, INFOSEC Research Council, 2010.

"**The Security Arms Race: Yesterday, Today and Tomorrow**", Invited talk, CONNECT Research and Innovation Summit, 2010.

"**Re: CAPTCHAs: Understanding CAPTCHA Solving from an Economic Context**", Yahoo, 2010.

"**Exploring the Economics of Cybercrime**", National Defense Industry Association Cyber Symposium, 2009.

"**Exploring Internet Crime, Economics and Security**", Invited talk, CONNECT Frontiers in Science and Technology Seminar Series, 2009.

"**Internet Criminal Economics**", Invited talk, La Jolla Conference on Innovation Support for National Security, 2009.

"**Internet Outbreaks: Epidemiology and Defenses**", Invited talk, ViaSAT/NSA Information Assurance Summit, Carlsbad, CA 2009

Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams", Invited Talk, Harvard University, 2009.

Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams", Distinguished Lecture, University of Pennsylvania, 2009.

Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams", Invited Talk, Mail Anti-Abuse Working Group meeting, 2009.

"**Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams**", Invited Talk, University of Washington, 2008.

"**Reconsidering Physical Key Secrecy: Teleduplication via Optical Decoding**", Invited Talk, University of Washington, 2008.

"**Issues of Security and Risk for Third-Party Cloud Computing**", Cisco Cloud Computing Research Symposium, 2008.

"**Exploring Internet Criminal Economics**", Distinguished Lecture, University of Virginia, 2008.

"**Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams**", Invited Talk, Yahoo!, Sunnyvale, CA, 2008.

"**Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams**", Invited Talk, Microsoft Research, Mountain View, CA, 2008.

"**Spam Analytics: Exploring the technical and economic factors in bulk e-mail scams**", Invited Talk, Campus LISA, UCSD, 2008.

"**Exploring the Underground Economy**", Invited Lecture at University of Chicago, 2008.

"**Understanding the Growth of the Underground Economy"**, Distinguished Lecture at Dartmouth College, 2008.

"**Underground Economics**", Keynote speaker at Anti-Spyware Coalition Workshop, 2008.

"**Underground Economics**", Santa Fe Institution Malware Workshop, 2007.

"**Understanding and Addressing the Threat of Internet Worms**" (joint w/Vern Paxson, Geoff Voelker and Nick Weaver), Day-long tutorial at USENIX Security Symposium, Vancouver, B.C., 2006.

"**Large-scale Internet Threats: Epidemiology and Defenses**", Keynote talk, World Conference on Structural Control and Monitoring, San Diego, CA, 2006.

"**Scalable Internet Threat Monitoring**", Hewlett-Packard Laboratories, Palo Alto, CA 2006.

"**Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis**", Google, Mountain View, CA, 2006.

"**Active Research in Addressing Unwanted Traffic**", Internet Architecture Board review, Marina del Rey, CA, 2006.

"**Scalable Internet Threat Monitoring**", UCSD Jacobs School of Engineering Research Expo, San Diego, CA, 2006.

"**Scalable Internet Threat Monitoring**", University of Toronto Distinguished Lecture, Toronto, 2005.

"**Scalable Internet Threat Monitoring**", Intel Research Council Workshop on Network Security, Hillsboro, OR, 2005.

"**Scalable Internet Threat Monitoring**", ACM Workshop on Rapid Malcode, Fairfax, VA, 2005.

"**Scalable Internet Threat Monitoring**", ARO/DHS Workshop on Malware 2005, Washington, D.C., 2005.

"**Internet Outbreaks: Epidemiology and Defenses**", Cornell University, Ithaca, NY 2005.

"**Internet Outbreaks: Epidemiology and Defenses**", Jacobs School of Engineering, Corporate Affiliates Program Board Meeting, San Diego, CA 2005.

"**Internet Outbreaks: Epidemiology and Defenses**", Network and Distributed System Security Symposium, San Diego, CA 2005.

"**Internet Outbreaks: Epidemiology and Defenses**", University of Washington, Seattle, WA 2005.

"**Low-cost 802.11 Wireless Infrastructure Networks**", Microsoft Mesh Networking Summit, Snoqualmie, WA, 2004.

"**The UCSD Network Telescope: A Real-time Monitoring System for Tracking Internet**", UCSD Jacobs School of Engineering Research Review Plenary Session, San Diego, CA, 2004..

"**Network Worms: Research and Engineering Challenges**", IPAM Large Scale Communication Networks Workshop, Lake Arrowhead, CA, 2003.

"**Network Worms: Research and Engineering Challenges**", Microsoft Security Retreat, Stevenson, WA, 2003.

"**Network Worms: Research and Engineering Challenges**", Intel, Hillsboro, OR, 2003.

"**Defending against Network Worms**", Hewlett-Packard Labs, Palo Alto, CA, 2003 (cancelled).

"**Denying Denial-of-Service**", Networld+Interop, Atlanta, GA, 2001.

"**DDoS Attacks: Why We Should Still Worry**", The Internet Security Conference, Los Angeles, CA, 2001.

"**Estimating Global Denial-of-Service Activity**", North American Network Operators Group, Scottsdale, AZ, 2001.

"**Internet Denial-of-Service Activity**", Stanford, Department of Computer Science, Palo Alto, CA, 2001.

"IP Traceback Techniques", North American Network Operators Group, Albuquerque, NM, 2000.

"**Sting: a TCP-based Network Measurement Tool**", CMU School of Computer Science, Pittsburgh, PA, 1999.

Awarded U.S. Patents

Thomas Anderson, David Wetherall and Stefan Savage, "**Detecting network misuse**", *US patent 8,509.086, Filed June 20, 2002, Issued August 13th, 2013.*

Sumeet Singh, George Varghese, Cristian Estan and Stefan Savage, "**Detecting public network attacks using signatures and fast content analysis**", *US patent 8,296,842, Filed Dec 1st 2004, Issued October 23rd, 2012.*

David Wetherall, Stefan Savage and Thomas Anderson, "**Independent detection and filtering of undesirable packets**", *US patent 8,271,678, Filed April 3rd 2001, Issued September 18t, 2012.*

Thomas Anderson, David Wetherall and Stefan Savage, "**Detecting and preventing undesired network traffic from being sourced out of a network domain**", *US patent 7970886, Filed November 2nd 2000, Issued June 28th 2011.*

Sumeet Singh, George Varghese, Cristian Estan and Stefan Savage, "**Detecting public network attacks using signatures and fast content analysis**", *US patent 7996658, Filed April 8th 2004, Issued June 21st 2011.*

David Wetherall, Stefan Savage and Thomas Anderson, "**Distributed service level management for network traffic**", *US patent 7475141, Filed July 31s 2001, Issued January 6th 2009.*

Thomas Anderson, Stefan Savage and David Wetherall, "**Network traffic regulation including consistency based detection and filtering of packets with spoof source addresses**", *US patent 7444404, Filed February 5th, 2001, Issued October 28th, 2008.*