

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

PALO ALTO NETWORKS,
INC.,
Petitioner, v.

BT AMERICAS INC.,
Patent Owner.

Case IPR2023-00889
Patent 7,895,641

DECLARATION OF WENKE LEE IN SUPPORT OF BT AMERICAS INC.'S
PRELIMINARY RESPONSE UNDER 35 U.S.C. §313 AND 37 C.F.R. §42.107(a)

TABLE OF CONTENTS

I.	INTRODUCTION	4
II.	PERSON OF ORDINARY SKILL IN THE ART	7
III.	SHORT SUMMARY OF CONCLUSIONS	8
IV.	MY QUALIFICATIONS, COMPENSATION, AND EXPERIENCE.....	15
V.	THE INVENTIONS DESCRIBED AND CLAIMED IN THE '641 PATENT	17
VI.	THE PRIOR ART	30
	A. Duvall.....	30
	1. Transmitting Content with TCP/IP	32
	2. Other Aspects of Duvall.....	45
	B. Chu	47
	1. Chu Focuses on Trust Management.....	47
	2. Chu's REFEREE Trust Management System	49
	3. Chu's Sample REFEREE Policy	50
	C. Cogger	52
VII.	THE PETITION FAILS TO ESTABLISH A REASONABLE LIKELIHOOD THAT THE ASSERTED REFERENCES RENDER ANY CLAIM UNPATENTABLE.....	55
	A. Multiple Features Present in Every Claim Are Missing from Petitioner's Asserted References	55
	1. Duvall and Chu Lack Analysis of Post-Filter Residue Status Data to Identify Potential Security-Related Events	56
	2. Duvall and Chu Lack a Second-level of Analysis By an Analyst System	62
	3. The Limitations Regarding Receipt of Feedback at the Probe to Dynamically Enhance Its Analytical Capabilities Are Missing from the Asserted References	68

B. There Is No Basis for Combining Duvall and Chu, No Details How to Do So, and No Reasonable Expectation of Success	71
1. Duvall Is Unsuitable for Use as a Network Security Monitoring System	72
2. Petitioner’s Motivation Relies on a Mischaracterization of Duvall’s Concerns about Subjectively “Filtering Too Much or Too Little”	78
3. Duvall and Chu Use Filters in Different Ways and for Different Purposes	79
4. A POSA Would Not Combine Chu with Duvall Given the Differences In Concerns, Underlying Assumptions, and Goals.	80
5. Even If a POSA Was Motivated to Search for Techniques to Resolve the Status of Residue, Chu Does Not Teach Any Such Techniques.	81
6. A POSA Would Not Have Arrived at the Claims Given Duvall and Chu.	82
C. There Is No Way to Combine Cogger into Duvall-Chu To Arrive At Independent Claim 18	89
VIII. CONCLUSION.....	91

I. INTRODUCTION

1. My name is Wenke Lee and BT Americas Inc. (“BT”) has retained me as a technical expert witness in connection with this proceeding.

2. A curriculum vitae (“CV”), which I have recently reviewed and believe to be accurate, is attached as Exhibit A to this Declaration. My particular area of expertise is network security and attendant technologies, with an emphasis on the detection of malware and network intrusions, as well as remedial measures to be employed following detection.

3. I have been asked to provide insight into the state of the art during the relevant time frame. In this regard, the stated Field of Invention of the U.S. Patent 7,895,641 (“’641 Patent” or “Schneier”) (EX1001, 1:18-20) matches my own area of expertise and work, starting with my PhD Thesis, which was completed in 1999. My thesis is directed to a data mining framework for constructing features and models for network intrusion detection systems. It describes algorithms and procedures for processing network traffic data and extracting useful attributes and patterns (or signatures) for the purpose of detecting attacks to enterprise networks.

4. I will also address the technology described and inventions claimed in Schneier, the references cited by Petitioner, and the differences between the inventions claimed in Schneier and the disclosures of the cited references. I will

also address related technical points which may prove helpful to the Board in assessing factual assertions and opinions set forth in Petitioner's papers and exhibits, all of which I have read and will address as appropriate to the present context.

5. I am not an attorney, but I am a named inventor in six patents, five of which are directed to detecting intrusions, malware, or responding to network attacks (as well as detecting them). I have also served as an expert and have provided testimony in multiple patent litigations. Consequently, I have a general familiarity with principles of patent law respecting validity and non-obviousness.

6. I understand that a patent is obvious if the differences between the claimed subject matter and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art. BT's counsel has informed me that obviousness is a question of law based on underlying questions of fact.

7. I understand that the underlying factual inquiries in an obviousness analysis include: (1) determining the scope and content of the prior art; (2) resolving the level of ordinary skill in the prior art; (3) ascertaining the differences between the claimed invention and the prior art; and (4) considering objective evidence of nonobviousness.

8. I also understand that the Petitioner, Palo Alto Networks, Inc. (“PAN” or “Petitioner”), has the burden to show that a person of ordinary skill in the relevant field had a reason to combine the elements in the manner claimed when asserting obviousness in view of a combination of references. BT’s counsel has informed me that I should value “common sense” over “rigid preventative rules” in determining whether a motivation to combine references exists. I understand that any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed. However, I also understand that if a person of ordinary skill in the art would have had reason to attempt to make the composition or device, or carry out the claimed process, Petitioner must also demonstrate that such a person would have had a reasonable expectation of success in doing so.

9. I understand that the express, implicit, and inherent disclosures of a prior art reference may be relied upon in the rejection of claims for anticipation and obviousness.

10. Further, I understand that hindsight may not be used to determine obviousness. In other words, it is wrong to use the patent as a guide through the prior art references, to combine the right references in a particular way so as to achieve the result of the claims of the challenged patent.

11. I further understand that a reference may be said to teach away when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the patent applicant. I also understand that a reference may teach away from a use when that use would render the result inoperable.

12. I understand that there is no suggestion or motivation to make a modification to a prior art reference if the proposed modification would render the prior art invention unsatisfactory for its intended purpose. I also understand that an obviousness allegation cannot be supported by a combination of references that would require a substantial reconstruction and redesign of the elements shown in the primary reference as well as a change in the basic principle under which the primary reference was designed to operate.

II. PERSON OF ORDINARY SKILL IN THE ART

13. For the purposes of this declaration, I assume that a person of ordinary skill in the art (a “POSA”) pertaining to Schneier at the time it was invented has the education and/or experience that Dr. Jeffay has set out in his declaration. *See, e.g.,* EX1003 ¶¶29-32.

III. SHORT SUMMARY OF CONCLUSIONS

14. I understand that the immediate question is whether there is a reasonable likelihood that Petitioner would prevail in showing any of the claimed inventions are invalid for obviousness. I do not believe there is any such likelihood. The inventions claimed in Schneier constitute a substantial advance in the art and were not obvious in any way as I understand that term to be employed in the law of patents. There are at least three separate features present in independent claim 1 of Schneier that would remain missing from the cited references, EX1004 (“Duvall”) and EX1005 (“Chu”). The claimed limitations related to these features are missing even if the references were combined in the manner presented by Petitioner. And there is no reason to combine those references in the manner Petitioner suggests. Likewise, at least one feature from claim 18 of Schneier is also missing from the cited reference, Duvall, Chu, and EX1033 (“Cogger”). There is also no reason to combine Cogger into Duvall-Chu as Petitioner proposes. My principal reasons for reaching these conclusions follow.

15. There are two independent claims in Schneier: claims 1, and 18. The relevant claim language present in each claim which relates to the first analytical step of filtering out residue status data and analyzing that residue to find potential events can be found in element b of claim 1 as follows:

analyze status data to identify potentially security-related events represented in the status data, wherein the analysis includes filtering followed by an analysis of post-filtering residue, wherein the post-filtering residue is data neither discarded nor selected by filtering[.]

EX1001, 33:30-35 (claim 1) (emphasis added)

It can also be found within the first step of independent method claim 18 as follows:

creating an event record for information received about an identified potentially security-related event occurring on the network, wherein the potentially security-related event is identified by filtering followed by an analysis of post-filtering residue, wherein the post-filtering residue is neither discarded nor selected by the filtering[.]

EX1001, 34:25-30 (emphasis added).

16. This feature is neither disclosed nor suggested by any reference cited by Petitioner. One major advantage of this aspect of the architecture described and claimed in Schneier is that it identifies, from residue, potentially security-related events, before anything is sent to the secure operations center (“SOC”). The claim limitation that relates to this (i.e., an analysis of post-filtering residue to identify potentially security-related events) is not in the references cited by Petitioner.

17. The next limitation present in each claim relates to the subsequent transmission of information about a potential event to the analyst system. The relevant language is in element c of claim 1:

transmit information about the identified events to an analyst system associated with the security monitoring system[.]

EX1001, 33:36-38 (claim 1) (emphasis added).

18. In this regard, there is so much traffic (and attendant status data) in most networks, that it is not realistic (in terms of cost, speed, and other operational parameters affected by high volumes of traffic) to send everything indeterminate on to the SOC for analysis. By conducting an analysis of residue to flag potentially security-related events before transmission to the SOC, the architecture described and claimed in Schneier manages to discover previously unknown intrusions into the system and manages the enormous amount of data filtered more efficiently. Furthermore, by conducting two levels of analysis, the first at the probe prior to transmission across pipes 3000 and the second at the SOC after transmission, false positives are reduced. *See* EX1001, 2:13-18.

19. Duvall is not as Petitioner claims. Duvall does not discuss filtering status data into three buckets (positive, negative, and residue) so that residue can be analyzed to flag potential events. Duvall does not employ filtering to flag potential events from residue, first, and only then transmit information about that potential event to a secure operation center for further analysis to determine whether a security event actually occurred.

20. Furthermore, Duvall only matches by comparison to known criteria, such as IP/ports, keywords, and patterns. Duvall does not find anything *potentially* security-related and is not capable of detecting intrusions that are completely unknown, *i.e.*, not variants of known criteria to which the processor can match.

21. The inventions of Schneier offer a solution that is capable of finding a far broader range of intrusions, including those previously unknown, by flagging any kind of potentially security-related event in residue status data. Schneier has a SOC. There, a security analyst system has access to multiple databases that provide security intelligence and sophisticated tools with which to determine whether a potential event flagged from residue is an actual intrusion and/or address how best to resolve an event by invoking all the resources available at the SOC.

22. Duvall and Chu also lack other features of the invention. In particular, there is no dynamic feedback mechanism to update the analysis capability of the probe through the receipt of feedback at the probe, as claimed in every claim of Schneier. The relevant claim language appears in elements d and e of claim 1 as follows:

d) a receiver for receiving feedback at the probe based on empirically-derived information reflecting operation of the security monitoring system; and

e) a modification control system for dynamically modifying an analysis capability of the probe during operation thereof based on the received feedback.

EX1001, 33:39-45

23. But Duvall provides an editing manager that allows users to tune the filters based on their subjective preferences. Subjective preferences are not empirically-derived information, because they are not objectively verifiable—varying from person to person. Regardless, Duvall does not ever disclose the capability to update its filters during operation. *See, infra*, ¶¶175-177. Instead, Duvall suggests the system would need to be taken offline to edit the filters and also proposes a search algorithm that is incompatible with dynamic updates. *Id.*

24. Chu also does not flag potentially security-related events from an analysis of residue so that information about potential events so identified can be transmitted to a SOC. Since Chu is missing the entire step of conducting an analysis of residue to flag potential events, it does not teach a second-level analysis to confirm the potential events. It cannot be combined in a way that even arguably approaches the architecture of Schneier.

25. Like Duvall, Chu simply misses the entire point of the filtering subsystem, as claimed and called out by the Examiner in her Reasons for Allowance. Duvall and Chu do not flag potential events from residue that was

neither selected nor discarded as the claims require and focus the subsequent transmission of information on the potential events so identified. In Chu, a response of “unknown” is simply returned for data not matching any filters. Not only does Chu not perform a second-level confirmatory analysis on identified potential events, Chu never even performs the first-level analysis to identify those potential events from the unknown data in the first place.

26. Chu allows for user intervention by the end-user because each end-user can immediately decide whether to accept the risk of accessing data with indeterminate trust. But Chu never teaches any further analysis by a trained security analyst (via an analyst system), and never teaches sending anything except the “unknown” response to the original application (rather than an analyst system). Because neither Duvall nor Chu perform an analysis that could sufficiently narrow what is sent to the (undisclosed) analyst system, Petitioner’s modification to add the analyst system suffers the exact problem that Schneier intended to fix—flooding the analyst system with an unmanageable amount of information.

27. To summarize, residue would have been considered unimportant by most persons skilled in the art at the time, but the inventors recognized that it was in the residue that clues about previously unknown intrusions might be found. Thus, they created an architecture that allowed for discovery of previously

unknown intrusions, while managing large volumes of traffic and reducing false positives.

28. Furthermore, the inventors dynamically update the probe with feedback from the SOC so that the probe can filter more efficiently in the future. Duvall does not provide dynamic feedback to the probe in order to update its capabilities. Instead, feedback is provided based on subjective preferences while the system is not operational.

29. In addition to the fact that three different claimed features are missing from Petitioner's references, however forced in combination, there is also no reason to combine Petitioner's references at all since Duvall is not even suitable as a security monitoring system and completely unrelated to Chu. Neither reference suggest filtering out residue status data from which to flag potential events to allow subsequent transmission of information about them for further analysis.

30. Moreover, not only would a POSA not combine Duvall and Chu as Petitioner proposes, a POSA would also not add Cogger to Duvall and Chu. Cogger only discloses an interface that allows for remote access to a pre-existing trouble-ticketing system used by telecommunications service providers. This system does not already exist in Duvall and Chu and merely adding the disclosed user interface from Cogger would not allow Duvall-Chu to enable remote

management of its filtering system such that it would read on Schneier claim 18.

IV. MY QUALIFICATIONS, COMPENSATION, AND EXPERIENCE

31. I am currently employed as a Professor at Georgia Institute of Technology, where I hold the John P. Imlay Chair in the College of Computing. I am receiving compensation from BT for my services in this proceeding at the rate of \$750/hr. My compensation is not dependent upon the outcome of this proceeding and has not affected the statements I am making in this Declaration in any way.

32. My main area of interest and the focus of my current research is malware analysis and detection. A fuller description of my current interests and areas of research can be found at the bottom of page 2 and top of page 3 of my CV.

33. As reflected in the succeeding pages of my CV (pages 4-7), I have taught courses directed to Network and Information Security essentially continuously since 1999.

34. I have published more than a hundred academic papers in top journals and conference proceedings, published half a dozen books or parts of books, and participated in well over a hundred Conference Presentations and over a dozen Workshop Presentations, the vast preponderance of which relate to Network Security and most of which relate to Intrusion Detection in particular.

35. I have also performed and supervised a substantial amount of research in the field of Network Security. A list of the 51 Research Proposals and Grants in which I have participated can be found at pages 31-37 of my CV.

36. Research Honors and Awards I have received are listed on pages 38-39 of my CV and include an award received in 2018 for co-authorship on a paper deemed to have had the greatest impact on security research and practice of the past decade. I am a Fellow of the Association for Computing Machinery (ACM), which is the world's largest scientific and educational computing society; only the top 1 percent of ACM members are recognized as Fellows. I am also a Fellow of the Institute of Electrical and Electronics Engineers (IEEE), the professional home for the engineering and technology community worldwide and the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.

37. I am also active in Conference Committee Activities and Seminars, all as reflected towards the back end of my CV.

38. I am named as an inventor in twenty patents (listed on pages 46-47 of my CV), most of which relate to the Field of Invention at hand, which is described in the next paragraph.

V. THE INVENTIONS DESCRIBED AND CLAIMED IN THE '641 PATENT

39. The Field of Invention to which Schneier relates is “network security and, more specifically, to methods and systems for dynamic network intrusion monitoring, detection and response.” EX1001, 1:17-20.

40. The inventions described in Schneier “offer[] methods and systems for dynamic network intrusion monitoring, detection and response.” *Id.*, 1:57-59. In one implementation, discussed in the Summary of Invention, Schneier is directed to a managed security monitoring system for monitoring a customer’s network. *Id.*, 1:59-2:2; 2:3-5; 2:28-29.

41. Managed security monitoring was a business in its infancy in 2000 and the architecture disclosed in Schneier, while not limited to monitoring customer sites, is certainly well suited to it as reflected in Figure 1. The dominant commercial practice was to install pattern-matching anti-virus software to analyze incoming traffic at the edge of the network where traffic comes in from the internet and update that anti-virus software periodically with software upgrades. Remote monitoring services were known, but not common, and there was no well-considered architecture to run them to help discover unknown intrusions.

42. Schneier describes an entirely novel architecture for remotely

monitoring a network for intrusions. An overview of a preferred embodiment is provided in Figure 1 copied below:

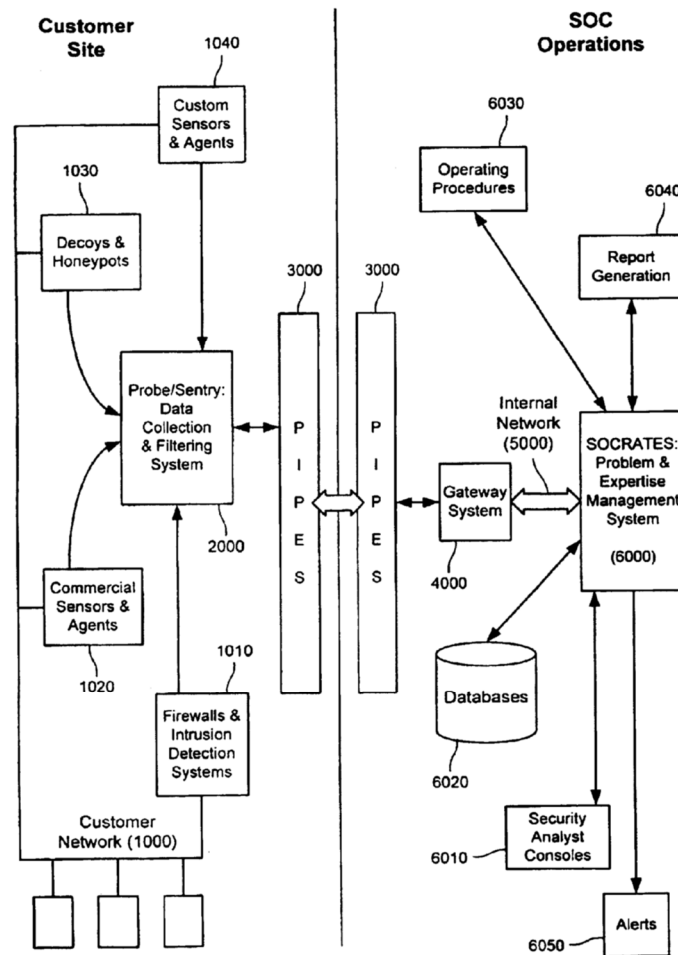


FIG. 1

43. In Figure 1, there is a customer site, which is depicted to the left of the figure, and a SOC which is depicted to the right. *Id.*, 4:44-49. Pipes 3000 provide a communications pathway, preferably encrypted and secure, between the customer site and the SOC. *Id.*, 5:50-54.

44. For purposes of clarity, I will move through the system as would typically occur during its operation. Consequently, I will first address the operation of the probe at the customer site as shown to the left of Figure 1, and, while doing so, focus in particular on some of the novel features captured by the claims. After discussing transmission of information to the SOC across pipes 3000, I will discuss the operation of the SOC and components shown to the right of Figure 1. Then, I will address the transmission of feedback from the SOC to the probe so as to dynamically update the probe once the analysis at the SOC has been completed.

45. A probe 2000, located at the customer site (i.e., to the left in Figure 1), collects and filters status data extracted or derived from traffic, conceptually speaking, into three categories. EX1001, 8:47-62. The first is positively-filtered data, the second is negatively-filtered data, and the third is “residue” which is itself analyzed to flag potentially security-related events. *Id.*

46. The status data which is filtered includes audit log and other audit information. EX1001, 4:64-5:3. Status data provides information about the status of the network 1000 and its components and, in the preferred embodiment, is exemplified by fields extracted from packets and the identification of components involved in a potential event. *See, e.g.*, EX1001, Table 6.

47. Schneier also states that status data is collected into sentry messages sent to the SOC, where the sentry messages are transformed into gateway messages as shown, for example, in Table 6. As stated, EX1001, 8:66-9:2:

“Communications and resource coordinator 2060 creates sentry messages out of the interesting status data and forwards those messages on to gateway system 4000...” Consequently, Table 6, located at columns 21-22, exemplifies the type of information comprising status data that might be included in a sentry message and gateway message derived from it. *Id.*, 9:2-7.

48. Towards obtaining status data in the first instance, the probe collects status data from sensors in the form of firewalls and intrusion detection systems 1000, commercial or custom sensors or agents 1020 and 1040, and decoys and honeypots 1030. EX1001, 4:55-64. The probe can monitor and collect information from any component that can provide status data. *Id.*, 4:64-5:3.

49. The features of the architecture of the filtering subsystem and associated components of the probe, which work together to create a sentry message for transmission to the SOC, are a unique feature of the invention and are depicted, schematically, in Figure 2, copied below:

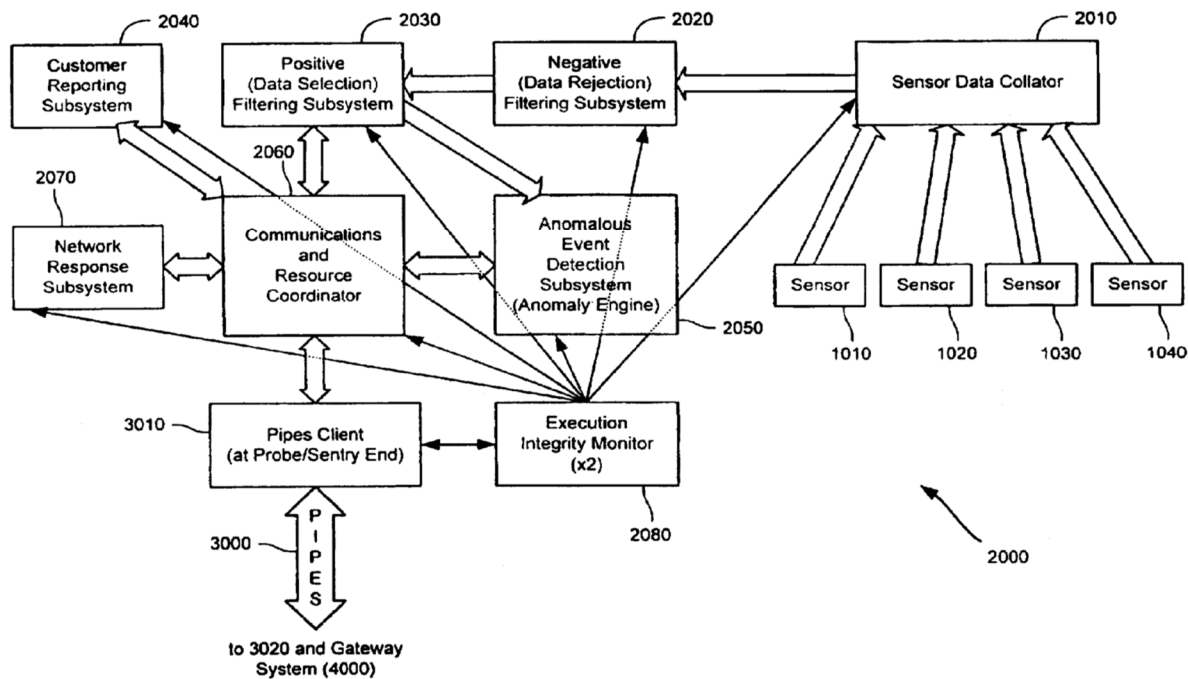


FIG. 2

50. In Figure 2, which depicts probe/sentry system 2000, a sensor data collator 2010 collates status data collected by the sensors. Then, as stated, EX1001, 8:51-59:

Once collated, the data is first filtered by negative filtering subsystem 2020, which discards uninteresting information, and then by positive filtering subsystem 2030, which selects possibly interesting information and forwards it to communications and resource coordinator 2060. Data neither discarded by negative filtering subsystem 2020 nor selected out as interesting by positive

filtering subsystem 2030 form “residue,” which is sent to anomaly engine 2050 for further analysis.

51. Anomaly engine 2050 picks through the residue to determine what residue information is sufficiently interesting that it should be sent to the SOC where trained analysts using analyst systems, having access to tools and databases allowing for a more sophisticated analysis, can make an informed determination as to whether there has been an intrusion and/or address how to best resolve an event. *Id.*, 8:59-63.

52. It is noteworthy that the inventors had the foresight to pick through leftover residue status data to flag information within it sufficiently interesting that it was worth sending to the SOC. Most persons skilled in the art at the time would have considered it uninteresting and discarded it. However, the residue is where the status data corresponding to unknown intrusions (i.e., those which the filtering system was unable to identify and filter out as bad) actually resides.

53. After anomaly engine 2050 identifies residue interesting enough to send to the SOC, it is sent to resources coordinator 2060, which creates sentry messages sent through pipes 3000 to the SOC. *Id.*, 8:67-9:10.

54. More particularly, after filtering by the probe, status data of sufficient interest is incorporated into sentry messages, which are sent to the SOC through

pipes 3000 and are received by gateway system 4000, where they are transformed into gateway messages. *Id.*, 5:25-29, 5:44-49.

55. In this fashion, a decision is made at the probe, after filtering, based on an analysis by anomaly engine 2050 from leftover status data (residue), as to whether there might have been a potentially security-related event worth investigating. The significant advantage of conducting the analysis of leftover residue at the probe is to flag potential security related events before transmission in order to control the amount of information subsequently transmitted to the SOC.

56. The generation of “residue” status data and the analysis of it to flag potential events worth investigating at the SOC is an entirely novel aspect of Schneier. Consequently, I digress briefly to point out the claim limitations of the independent claims directed to these particular novel features.

57. The generation and analysis of residue is captured by language found in element (b) of claim 1 (and its dependents) as follows:

analyze status data to identify potentially security-related events represented in the status data, wherein the analysis includes filtering followed by an analysis of post-filtering residue, wherein the post-filtering residue is data neither discarded nor selected by filtering;

See id., 33:31-35 (claim 1) (emphasis added); *see also id.*, 34:25-30 (showing

claim 18).

58. The next limitation of claim 1, which establishes that the information transmitted to the SOC is about potentially security-related events already identified (through analysis of residue status data) is found in element (c) of claim 1 (and its dependents) as follows:

transmit information about the identified events to an analyst system associated with the security monitoring system;

See id., 33:36-38 (emphasis added)

59. This first level of analysis takes place at the probe, after which information about an event identified through that analysis is sent to the SOC. Moving to the SOC shown to the right in Figure 1, sentry messages first pass through pipes 3000 into a gateway system 4000.

60. Gateway system 4000 is broken out schematically in Figure 3 and described at 9:42-53 of Schneier. Sentry update manager 4010 is responsible for updates sent back to the probe and a communications and management coordinator 4020 converts sentry messages into gateway messages passed onto the SOC. *Id.*, 9:50-51.

61. The “problem and expertise management system” that manages and

addresses actual and potential problems at the customer site, e.g., as reported through sentry messages generated from an analysis of residue status data, is called SOCRATES (an acronym for Secure Operations Center Responsive Analyst Technical Expertise System per col. 3:61-62), which is identified as element 6000 in Figure 1.

62. As stated at 9:56-62 of Schneier:

The SOCRATES system is a consolidated system used to manage customers' problems and the supporting data helpful in resolving such problems. It provides security analysts at a SOC a single, integrated system with which to track information concerning, for example, problems, companies, people, contacts, tools, and installed network components and known vulnerabilities.

63. The SOCRATES system and its associated databases and console is best shown in Figure 4, with the succeeding Figures showing incident workflows and the handling of customers calls when the system is in use.

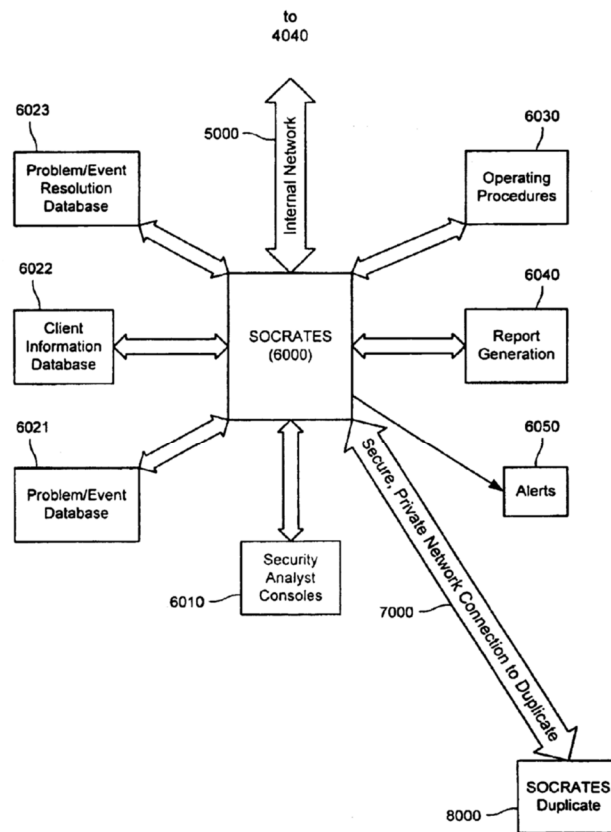


FIG. 4

64. As reflected in Figure 4, security analyst consoles 6010, which interface with SOCRATES, are available to security analysts, i.e., “personnel specializing in the analysis of network attacks.” *Id.*, 2:43-44. In an exemplary embodiment, security analysts work in concert with SOCRATES to categorize, prioritize, investigate, and respond to customer incidents or problems. *Id.*, 3:42-45.

65. SOCRATES has access to multiple databases, shown as 6021, 6022, and 6023, which provide “security intelligence” in the form of characteristics and

known vulnerabilities of certain products, characteristics of hacker tools, and the effectiveness of alternative responses. *Id.*, 2:55-64.

66. In keeping with the preferred implementation of providing a managed security monitoring system for customers (col. 1:59-2:2), SOCRATES can generate customer-specific reports 6040 or send customer-specific alerts 6050. *Id.*, 7:7-9; 7:32-33. The operating procedures 6030 for operating SOCRATES itself can be tailored to meet specific customers' needs. *Id.*, 7:61-63.

67. In operation, a sentry message is sent from the probe, reformatted into a gateway message and transmitted across internal network 5000, to inform SOCRATES of the potential event. *Id.*, 10:10-15. SOCRATES then uses the gateway message to form an event record. The event record is linked with other event records, customer information, problem resolution information, and other information available from the databases associated with SOCRATES to form "problem tickets" displayed on console 6010 for a security analyst to handle. *Id.*, 10:15-32; 10:45 *et seq.*

68. Once the analysis of a potential event has been completed at the SOC, the security monitoring system provides feedback from the SOC to the probe which contains a communications and resource coordinator within 2060 to update software at the probe, dynamically, to provide the probe with an enhanced

capability with which to perform filtering. *Id.*, 5:32-43.

69. As reflected in the Summary of the Invention, the preferred embodiment of Schneier describes an overall managed security monitoring service “that monitors a customer’s network activity using a probe or ‘sentry’ system, collects status data from monitored components, filters or otherwise analyzes the collected data for activity possibly implicating security concerns, alerts and transmits information about such activity to trained security analysts working at secure operations centers (‘SOCs’). . . .” *Id.*, 1:59-67.

70. As is also reflected in the Summary of the Invention, the component parts of that service might include a probe/sentry system located at a customer site (3:11-26), a gateway system to link the probe to the SOC (3:27-29), and a SOC (2:65–3:9) where security analysts work (2:42-50, 2:65-67) and can access the SOCRATES problem and expertise management system (3:42-45) which has access to multiple databases that provide security intelligence (2:55-64).

71. There are many benefits to this architecture, as implemented by the inventors in a manner which was unheard of in 2000 and has now come to be widely employed. I start with the fact that the identification of the potential security event occurs from an examination of post-filtering residue comprised of status data that was neither selected by positive filtering nor discarded by negative

filtering.

72. In this regard, there is an initial analysis at the probe in the customer network to determine whether there is a potential security-related event. The initial analysis is performed, post-filtering, upon residue status data and it is performed before any information is sent to the SOC. Considered to be of little interest by many, the inventors of Schneier recognized that it is in just such residue that clues about unknown intrusions are most likely to reside. This is the more effective and the right approach to identify a wider variety of previously undiscovered forms of malware and attack traffic than previously employed techniques such as pattern matching with variants of patterns of known malware and attacks.

73. The initial analysis at the probe cuts down considerably on the flow of traffic to the SOC by limiting that transmission to information about potential events flagged by the anomaly engine. Prior art approaches that transmit all traffic or all information extracted or derived from traffic would be comparatively slow and expensive. Prior art approaches that leave residue discarded to reduce volume were unable to use that residue to uncover previously unknown intrusions.

74. Further, by conducting a second analysis at the SOC and providing a feedback mechanism to dynamically update the probe (*see, e.g.*, EX1001, 5:32-43), rather than relying on the probe to make the decision whether there has actually

been a security event as traffic comes in, “false positives” are reduced. Still further, when a previously unknown attack is confirmed by the SOC, the probe can be updated with new information from the SOC to detect future occurrences of the same attack (now identified as a known attack).

75. Although analysis of post-filtering residue at the probe before transmission to the SOC makes the initial determination about whether there has been a potential event, a more detailed analysis of that event can be performed at the SOC, with all the tools available to the system and security analysts working there. *See id.*, 2:15-18 (“[P]articulars may then be examined and analyzed by trained security analysts continuously monitoring the customer’s network to further understand the incident and eliminate false positives.”). In this fashion, false positives, a significant problem at the time, are reduced.

VI. THE PRIOR ART

A. Duvall

76. Duvall focuses on censorship by filtering “personally objectionable” “material received over the Internet.” See EX1004, 2:12-13. Personally objectionable material is subjectively pre-determined “depending on [each] user’s individual tastes and sensitivities.” *Id.*, 2:12-16.

77. Duvall explains many “parents are particularly concerned that their

children can get access to [] objectionable materials,” such as “sexually explicit materials.” *Id.*, 1:13-17. Other examples of objectionable material disclosed by Duvall include material that is “sexually explicit, violent, politically extreme, or otherwise.” *Id.*, 2:13-15.

78. Duvall therefore censors based on subjectively predefined criteria. *See* EX1004, 1:13-17, 2:12-13. Duvall uses filters to match criteria and filter-actions to block or allow transmissions that are matched by its filters. *See id.*, 2:12-16.

79. Although Duvall does not explicitly define two categories of users, Duvall contemplates two different types of users. There are users with privilege to configure the database (*e.g.*, parents/companies), and the end-users lacking those privileges (*e.g.*, children/employees) to whom the censorship is meant to apply. *See, e.g., id.*, 8:1-16, 11:18-20. For example, Duvall explains that many “parents are particularly concerned that their children can get access to [] objectionable materials.” *Id.*, 1:13-17. Therefore, the parents are the users that would configure the filters while the children are the end-users to whom the censorship is meant to apply.

80. Duvall’s filtering system “compares portions of incoming and/or outgoing messages to filtering information in a filter database and determines

whether to block or allow incoming and/or outgoing transmissions of messages in response to the comparison.” *Id.*, Abstract. I understand that this description provided by Duvall describes both the direct-action and deferred-action filters. In this way, Duvall treats its filters as a uniform set of filters. Indeed, all filters are represented as entries in a common database, with fields for criteria and fields specifying an action to take. *See* 3:64-4:15; 5:8-19.

81. Though Duvall’s disclosure concerns a filtering system, Duvall’s main focus is on providing different ways to censor (or block) content matched by its filters. Objectionable material is blocked “[i]n response to a match” with a single filter from the database (*id.*, Abstract; *id.*, 1:35-40), which “takes action based on the first matched filter” (*id.*, 4:29-30). So, for Duvall the filter itself specifies the criteria that should be matched and an action to take. But each filter can have different criteria and can also specify different actions to take on the matched transmissions. Specifically, Duvall discloses three blocking actions available when filters match a transmission: “discarding incoming data, preventing execution of an open command, or replacing parts of received data.” *Id.*, 1:35-40.

82. Before getting into how these three blocking actions work, it is necessary to understand the context of the transmissions that Duvall operates on.

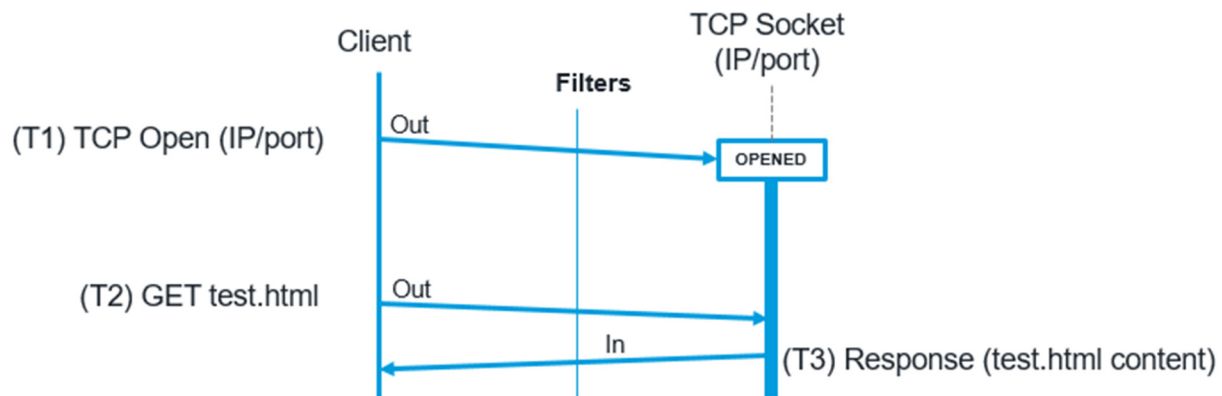
1. Transmitting Content with TCP/IP

83. Duvall’s filtering system operates on TCP/IP traffic. Duvall explains that “[m]essages are transmitted over the Internet ... with a transport protocol known as Transmission Control Protocol/Internet Protocol (TCP/IP).” *Id.*, 2:41-44. Notably, a POSA would understand that while TCP is a transport layer protocol (*i.e.*, layer 4) commonly used to transmit information over the Internet, it is not the only transport layer protocol used over the Internet. In other words, data may also be transmitted over the Internet without using TCP. Because Duvall is focused primarily on censorship, this narrow focus is fine and to be expected. In the context of a parent-child or employer-employee setting, as contemplated by Duvall, the censorship system should focus primarily on data sent through TCP/IP in layers 6/7 (the application and presentation layers) because that is where objectionable content is most likely to be found.

84. As Duvall explains, TCP “provides an error-free data connection between computers, and sits underneath and works in conjunction with higher level network protocols, including HyperText Transfer Protocol (HTTP).” *Id.*, 2:44-48. And with HTTP, “a user can access web sites over the world wide web using one of a number of commercially available web browsers.” *Id.*, 2:63-65. HTTP is an example of an application layer protocol where potentially objectionable content is likely to be found (*i.e.*, when browsing the web using a browser).

85. Duvall explains how a website is typically accessed using TCP. An understanding of this process is important to understanding how and why Duvall's filtering system works. When an end-user wants to access a website, the desired URL is first resolved into an IP address ("IP") and port. *Id.*, 2:66-3:19. The end-user's client device then attempts to open a TCP connection with the host at that particular IP/port. *Id.* This TCP data stream allows the client to communicate over the Internet with the website host at that IP/port. *Id.*, 2:58-62. Once the data stream is opened, the desired website can be requested and received over the opened data stream. *See id.*, 3:18-19; 6:24-25. In the case of requesting access to a webpage through a browser, the content is transmitted in accordance with the HTTP protocol through the TCP connection.

86. Below I provide a sequence diagram created for illustrative purposes, which shows how messages are sent in Duvall's detailed example:



87. After resolving a URL's IP/port, the client sends an initial outgoing transmission (T1) to open the TCP connection (*i.e.*, TCP Socket). *See Id.*, 2:66-3:19. Once opened, the client requests the desired webpage through the opened TCP data stream in an outgoing transmission (GET test.html, T2). *See id.*, 3:18-19; 6:24-25. Once the client request (T2) is received by the host, the host responds and transmits the requested content back to the client in an incoming transmission (test.html content, T3) via the same stream. *See id.*

88. Duvall monitors these incoming and outgoing transmissions (*i.e.*, T1-T3). *Id.*, 3:50-63. After matching the transmission to a filter, Duvall takes the action specified by that filter—*i.e.*, blocking action or allowing action. *See id.*, 1:34-40. With respect to each individual transmission in the sequence, Duvall only searches for a single filter from its database that should take action. *Id.*, 4:29-30.

Duvall does this by attempting to apply its filters from its database in sequential order until the first matching filter is found. *Id.* However, as discussed below, the action taken by the filter on each matched transmission varies depending on the type of action specified in the filter.

a. Duvall's Filter Actions

89. Duvall's filtering system censors content based on users' personal tastes and sensitivities. *See id.*, 2:12-16. Users (*e.g.*, parents) subjectively define what they want to censor by specifying the matching criteria of the filters (either the known objectionable content itself, or the known sources of such content). *See id.*, 6:60-7:8. Matched content can then be censored with Duvall's "blocking actions" or allowed. *Id.*, 1:34-40.

90. Duvall's three blocking actions enable the same set of filters in Duvall's database to work on either incoming or outgoing traffic and also block specific objectionable content in the traffic or known sources of objectionable content. *See id.*; *id.*, 4:22-28. Duvall explains that "[b]locking actions that are available for incoming messages are different from those for outgoing messages." *Id.*, 5:35-37. Outgoing transmission that would otherwise open the data stream can be blocked when the IP address is known to be a source of objectionable content—unconditionally censoring communication with the host by "preventing execution

of an open command.” *Id.*, 1:36-40; *see* 4:56-61; 6:14-18. For incoming transmissions, Duvall can “discard[] incoming data ... or replac[e] parts of received data” to censor specific pages or specific words/content on a page the user finds objectionable. *Id.*, 1:36-40; *see id.*, 5:35-40.

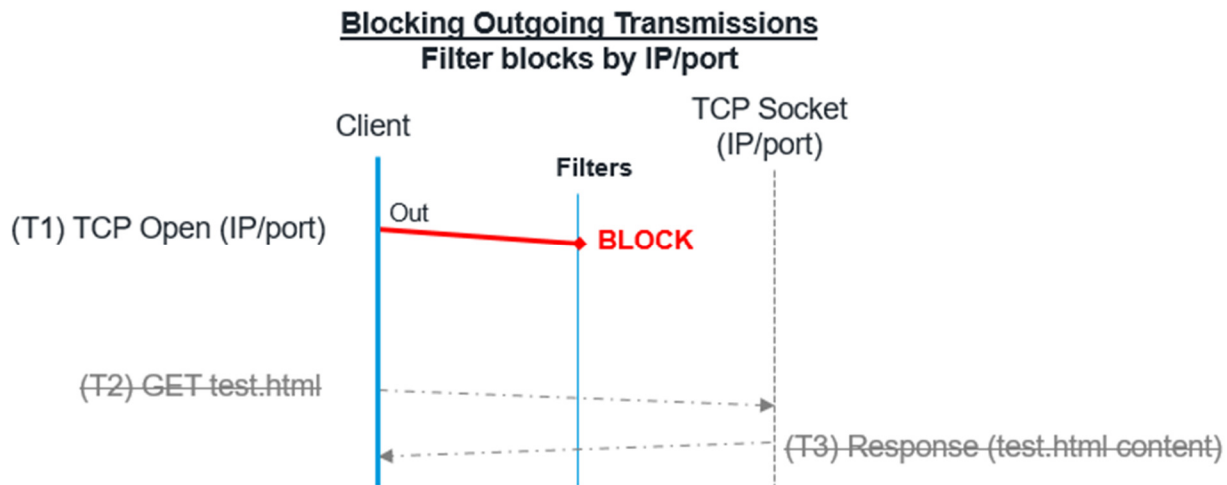
91. Duvall categorizes its filter actions into two logical categories—direct and deferred-action. Notably, both are used to block/allow known objectionable content or known sources of objectionable content. But they can apply to the sequence of TCP/IP transmissions to censor in different ways (summarized below).

92. As will become apparent, in Duvall, neither the filters nor the filter-actions ever identify anything previously unknown or elevate it to the level of a “potential” event. There is also no feedback loop to update the system dynamically to capture the previously “unknown” as “known” next time.

i. Blocking All Traffic From a Source

93. Duvall can censor a domain known to be a source of objectionable content by blocking initial outgoing transmissions used to open TCP connections based on the IP. *See id.*, 1:36-40.

94. A visualization of Duvall’s detailed example is provided below for illustrative purposes.



95. Duvall prevents the TCP stream from opening by blocking the transmission to open the stream (*i.e.*, T1 never reaches destination). *See id.*, 4:56-58. Without an open stream, content cannot be requested or received from the host (*i.e.*, request (T2) cannot be sent, and host cannot respond (T3)). The request (T2) cannot be sent because there is no TCP stream to send the request through. The host will not respond (T3) because it never received a request for content (T2). And regardless, the host could not transmit data to the client (T3) because there is no TCP stream to send the data through. Therefore, all content from this IP/port is unconditionally censored because no TCP connection is ever made. *See* 4:38-55.

96. Alternatively, a filter's action can be to allow all communications with an IP/port (*i.e.*, by allowing the data stream to be opened). *See id.* This lets the client request and receive all content from that stream unconditionally. With

respect to each matched transmission, the filter's action is just to allow. But in the context of the sequence of transmissions within the TCP stream, the same ALLOW filter will match with every transmission in that stream because they all share the same IP/port. These BLOCK/ALLOW actions are "direct-actions" "carried out immediately" on the matched transmission and they "indicate that the system should unconditionally allow or unconditionally block the transmission." *Id.*, 4:15-20.

ii. Blocking Specific Content From a Source

97. Instead of unconditionally censoring/allowing, users may want the ability to only block specific content known to be subjectively objectionable. *See id.*, 3:37-40. Duvall can block/allow specific pages or content from a domain after the content is received. *Id.*, 5:35-40. Duvall describes the same set of filters being applied to the outgoing transmission for that content. *Id.*, 5:35-40. The user also can target incoming content with the filters, allowing the system to censor in different ways (*i.e.*, "discarding incoming data ... or by replacing parts of received data"). *Id.*, 1:39-40. *See also id.*, 5:35-40.

98. Duvall thus provides additional actions that can instead "defer the decision whether to allow or block, and then monitor transmissions to search for a particular command and a particular filter pattern." *Id.*, 1:46-48. Instead of

immediately blocking or allowing the matched transmission, Duvall explains that it will continue to monitor other transmissions in the stream. *Id.* When a later transmission is matched, action can be taken on that later-matched transmission. *Id.* This action therefore waits to act on later transmissions in the TCP sequence. This type of action is a “deferred-action.”

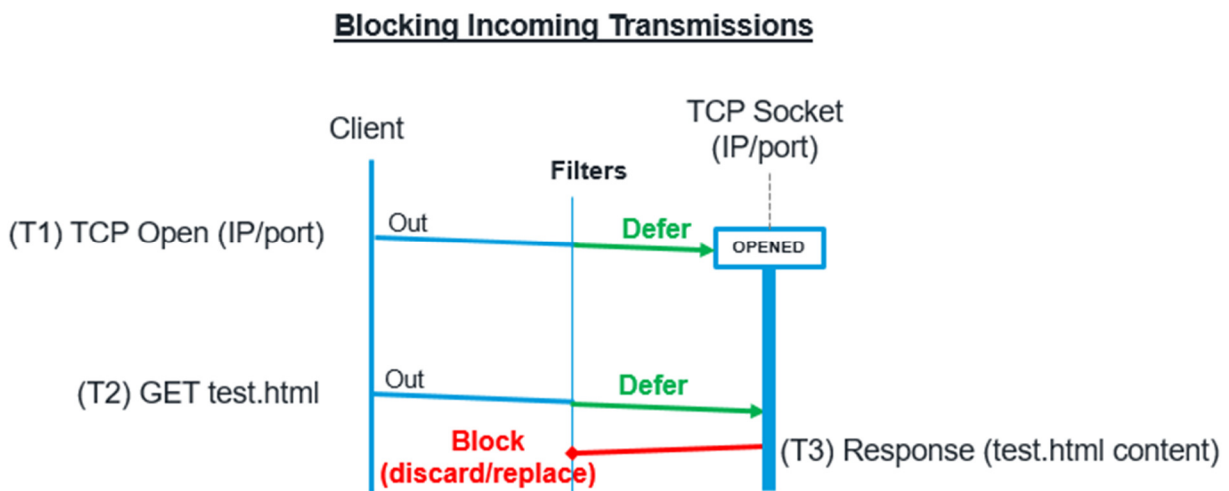
99. Notably, the filtering process always remains the same with respect to each transmission, *i.e.*, Duvall only attempts to match the filtering criteria to the transmission in sequence until the first matching filter is found. But the action taken on each matched transmission differs depending on the action specified and where the matched transmission is in the sequence of TCP transmissions (*i.e.*, T1, T2, or T3). *See id.*, 1:46-48; 6:19-26.

100. For example, the action taken by a filter on an initially matched outgoing transmission may be “deferred.” *See id.*, 6:19-20 (“Rather than specifying blocking at this time, a filter can indicate a deferred action.”). Based on Duvall’s example, this initially matched outgoing transmission is effectively allowed when the filter “indicate[s] a deferred action” because Duvall discloses that it is successfully transmitted to the host such that “the host server will respond with data for that directory.” *See id.*, 6:19-26. But when a different transmission within this sequence is matched (incoming transmission), “the system can block

the incoming data by discarding it or replacing it.” *Id.*, 6:27-28.

101. Filters using deferred-actions can specify keywords and filter patterns that can be mapped directly to content to determine whether it is known to be objectionable and therefore should be censored. *See id.*, 5:8-19.

102. A visualization of the process described at EX1004, 6:10-28, is provided below.



103. In Duvall’s example, incoming transmissions are responses to an outgoing request for data (*e.g.*, GET test.html, T2). *See* EX1004, 6:10-28; *see also id.*, 1:49-50. Notably, the outgoing transmissions (T1-T2) must be successfully transmitted before the host can respond with incoming data. T1 must be successfully transmitted to open the TCP stream. T2 is the request for data, which

must be successfully transmitted before the host can respond to any requests (T3).

104. Thus, Duvall can allow the data stream to open (T1) and permit the detected outgoing transmission (T2) while also queueing the filters for later incoming transmissions (T3). *See* EX1004, 6:19-28, 5:30-35. Specifically, Duvall explains:

The filters are preferably queued so that filter pattern matching is performed when incoming data is received from the host server following a particular detected outgoing command. With such queueing, a similar scan of the message is performed on the outgoing transmission, but blocking is performed on the incoming message.

Id., 5:30-35.

105. As Duvall explains, the outgoing command is detected, but blocking is not performed on that outgoing message. Duvall defers action on that detected outgoing message—in effect allowing the transmission to succeed. Instead, blocking is “deferred” until the incoming message matches the filter.

106. Notably, an otherwise matching deferred-action filter will not have a chance to act if it is arranged after a matching direct-action filter because Duvall only acts according to the first match. If a direct-action filter is matched before the deferred-action filter, the direct-action filter will always take precedence.

Therefore, if a deferred-action is desired, no other filters can take precedence when

setting the ordering of the filters.

107. Thus, Duvall can defer action on the outgoing transmissions (T1-T2) by effectively just allowing them and waiting to match later incoming transmissions in the sequence. *Id.* When an incoming transmission is matched (T3), its data can be discarded or replaced. *Id.*

108. A POSA would recognize that deferring action in this way is important for Duvall's system to enable censoring by discarding or replacing received content. This is—in part—because Duvall allows users to configure default filters that block by default. *Id.*, 5:1-7, 7:5-13. If Duvall's system filters only matched the incoming transmission, but not the outgoing transmission requesting that content, those outgoing request would end up blocked by default. With the initial request blocked, there would be no incoming data to censor. But by deferring action on the outgoing transmission, the outgoing transmission will not be blocked by default. This deferral is therefore important to allow the outgoing transmission to be transmitted such that the incoming response can be censored.

b. Duvall Acts on the First-Matched Filter

109. Duvall applies a single set of filters from its filtering database to each transmission and “takes action based on the first match[.]” *Id.*, 4:30-31; *see also*

id., 1:42-52. Although Duvall categorizes the types of actions into two logical groups (*i.e.*, direct and deferred-action), each filter just exists as a filter entry in Duvall's filtering database, regardless of its action type. *See id.*, 4:12-21. Duvall simply applies the same set of filters to each transmission when sequentially searching for the first singular match. *See id.*, 4:28-37, 5:8-29.

110. Duvall also provides users with a default filter—which the user can use to BLOCK/ALLOW transmissions that do not match with another filter in the list. *Id.*, 5:1-3. But data matched by the default filter is not data the user is uncertain about. Instead, it is data the user prefers to block or allow according to the user's subjective preference.

111. For example, Duvall suggests a user may want to “block access to all sites, and then allow access to specific sites.” *Id.*, 7:7-8. Any sites the user wants to allow access to would be placed before the default filter. *Id.* Because the user wants to block access to all other sites, the user would use a default block filter to accomplish that. *Id.* The data blocked by the default filter is not data the user is uncertain about (*i.e.*, not sure whether he wants to see that data), but instead data that the user intentionally wants to block. *Id.*

112. In short, Duvall's entire system only filters based on criteria known to the user (*i.e.*, based on predetermination that certain content or source of content is

objectionable). Duvall's filters never identify anything that the user was uncertain about, which would rise to the level of a *potential* event. The user subjectively determines whether a transmission should be blocked or allowed through the specification of its filters. Duvall therefore has no need for any second-level objective confirmation given this subjective nature.

2. Other Aspects of Duvall

113. Duvall discloses an updating mechanism allowing “existing filters in the filter database to be replaced with ... newly downloaded filters.” *Id.*, 7:27-29. While these updates are provided over the Internet, Duvall never discloses updating while operational. *See id.*, 7:18-29.

114. An editing manager “allows the user to edit the database” using “known database editing techniques.” *Id.*, 8:11-12. Users can “make custom changes if the user believes that the filtering system is filtering too much or too little.” *Id.*, 8:6-8. The editing manager allows users to “tweak” the overall system to censor more or censor less based on their subjective preferences. *See id.* But Duvall never suggests “filtering too much or too little” because a user is unsure whether something is objectionable.

115. Duvall contemplates editing the filters offline—not while the system is in operation. Duvall explains that “during a session over the Internet, a user can

copy URLs for **later** editing.” *Id.*, 8:13-14 (emphasis added). Then, after a session over the Internet, the user “can then copy those URLs for inclusion in the database, or can edit the entry.” *Id.*, 8:14-16. Duvall’s language here makes more sense when considering that dial-up internet would have been prevalent at the time of Duvall’s invention.

116. The “filtering system can [also] be provided from a server 30 that is on the client’s own network 40.” *Id.*, 8:19-22. The “filter database [will be] located on server 30.” *Id.*, 8:26-27. An alternative search technique is needed to “be efficient enough for a server environment.” *Id.*, 8:39-42. Duvall therefore performs pre-processing on the server database to allow the filtering system to search more efficiently at runtime. *Id.*, 8:42-67.

117. This pre-processing generates various tables that are used for searching. *Id.*, 8:39-61. It is incompatible with dynamic updates because the tables must be generated by the database whenever there is a change to the filters. With Duvall’s suggested search algorithm, the system cannot search the database before generating a new probe table. During the time when the database is pre-processing the filters to generate the probe table, Duvall’s filtering system would be inoperable.

118. Finally, Duvall discloses patching code for its content filtering

system. *Id.*, 9:14-11:20. Duvall’s method for patching is designed to prevent end-users from bypassing Duvall’s censorship tool. *See id.*, 9:65-10:63; *see also id.*, 2:1-11.

B. Chu

119. Chu discloses a blacklist/whitelist filtering policy that filters URLs in the context of trust management. *See* EX1005 at 11, 43. The policy will return a response of either “*true*”, “*false*”, or “*unknown*,” indicating whether the URL associated with a file is trustworthy. *See id.*, 31-32; 43-44.

120. Trustworthiness for Chu’s sample policy is based on the file’s source (*e.g.*, URL). *See id.*, 24; 43-44. If the source is trustworthy, then the information itself will be treated as trustworthy, and vice versa. *See id.*, 24. When the policy returns “*unknown*” (*i.e.*, trustworthiness of the source cannot be determined) control is returned to the application seeking to access the file, and the end-user is prompted for attention by the application. *Id.*, 43-44. No definitive trust/do-not-trust determination is made, nor is any analysis conducted on the “*unknown*” results. Instead, “*unknown*” simply reflects the lack of any determination or analysis. *Id.*, 43-44.

1. Chu Focuses on Trust Management

121. Instead of focusing on analysis techniques to identify potential threats

(e.g., analyzing the content), trust management—according to Chu—considers the trustworthiness of third parties providing data. *See id.*; *see also id.*, 32. Chu does not determine whether a file is safe, only whether the party distributing the file is trustworthy. *See id.*, 11; *see also id.*, 32. Chu therefore explains that trust management is a “distinct aspect” of security. *Id.*, 11.

122. Chu initially observes various concerns related to downloading a game from the web, including whether the game has a virus. *Id.*, 23. Instead of addressing these concerns by providing techniques to find viruses, Chu treats them as trust management issues. *Id.* Indeed, one of the concerns listed by Chu is whether the game is fun to play. *Id.* Just like the question of whether the game has a virus, Chu does not actually address that question head on. *Id.*

123. For Chu, these concerns exist because the user cannot trust that the provider of the code is who the provider purports to be, especially in the Internet age. *See id.*, 23-24. Chu lists three possible ways to address these concerns from a trust management perspective. *See id.* The first approach limits code functionality using a sandbox so no trust or analysis is needed. *Id.* But “[n]o matter how much effort is put into building a robust sandbox, hackers can always find security holes to compromise the sandbox.” *Id.* Notably, this sandbox approach disclosed by Chu does not involve analyzing the data with a sandbox. *Id.* Instead, by limiting

what the code can do, the sandbox eliminates the need to analyze the data. *Id.*

124. The second approach requires additional code to prove the original code is safe. *Id.* This approach also is impractical because there are too many different types of executables, which all need different solutions. *Id.*

125. Ultimately, Chu prefers the third approach—code-signing. *See id.*, 24-25. Code-signing “uses cryptography to establish authentication and integrity of an untrusted piece of code.” *Id.*, 24. A third-party can “provide digitally signed metadata to express trust,” and applications can then decide whether to trust the code, based on whether they trust the third-party. *Id.* In other words, Chu looks at who the sender is.

126. To that end, Chu teaches a “trust management infrastructure” facilitating this process. *See id.*, 25. Rather than identifying potential security risks by directly analyzing the data, Chu’s infrastructure determines whether any third parties associated with the data are trustworthy. Notably, in the context of network security, it is unrealistic for the source of malware to sign that malware. Likewise, even if signed, it does not mean the data itself is safe. In this way, a POSA would not look to Chu for techniques to confirm whether a potential threat is actually a threat.

2. Chu’s REFEREE Trust Management System

127. According to Chu, the “trust management infrastructure” has four key components: metadata format, trust protocol, trust policy language, and execution environment. *Id.*, 13. Together, they form Chu’s “trust management system.” *Id.* *Id.* According to Chu, its execution environment—REFEREE—“is the heart of a trust management system.” *Id.*, 28. The various policies that Petitioner relies upon from Chu are disclosed in the context of REFEREE.

128. REFEREE uses a tri-valued logic operand, which has three possible outcomes: *true*, *false*, and *unknown*. *Id.*, 31-32. Importantly, *unknown* means that “REFEREE was unable to find sufficient credentials.” *Id.* When returned, “REFEREE forces the host application to decide what action (if any) should be taken.” *Id.*, 32. In other words, Chu merely indicates that it has failed to identify the source as trusted when *unknown* is returned.

3. Chu’s Sample REFEREE Policy

129. Chu discloses sample policies for REFEREE. *Id.*, 43-48. These policies “use code-signing as the target trust management problem.” *Id.*, 43. Therefore, Chu’s concern is about whether a URL corresponds to a trusted entity (*e.g.*, whether “**Error! Hyperlink reference not valid.**www.harverd.edu” is trusted), and not about identifying potential security risks. *See id.*, at 44. One sample policy from Chu uses a whitelist and blacklist to filter URLs. *Id.* Chu

observes that whitelist and blacklist filters require the user to know beforehand what sites should or should not be trusted. *Id.* Therefore, Chu explains that such a policy “is useful if the application knows *a priori* a list of sites or directories it should not download codes from.” *Id.*

130. Chu also explains that REFEREE will return “*unknown*” when the requested URL is in neither the blacklist nor the whitelist. *Id.* And when “*unknown*” is returned by REFEREE, the application will “prompt [the end-user] for [her] attention.” *Id.*, 44. Notably, REFEREE only returns “*unknown*,” and defers control back to the calling application. *Id.*, 32. The application prompts the user in response to receiving a response of “*unknown*.” *Id.*, 43-44.

131. The “blacklist and whitelist ensure good automation of the trust decision process,” because the end-user need not be prompted to intervene so long as the URL is in either list. *Id.* “User intervention is needed only when the given URL is in neither the blacklist nor the whitelist.” *Id.* End-users are left to choose whether or not to accept the risk of proceeding with code from an unknown source. But that choice has no bearing on the (future) trustworthiness of the URL or determining whether the data/URL itself is actually problematic. Nor does it suggest the end-user could perform any further analysis to make such a determination. Indeed, the end-user is in a poor position to perform this analysis

given they are unlikely to be trained professionals.

C. Cogger

132. Cogger discloses an interface for customers to remotely access a telecommunications service provider's pre-existing trouble ticketing system. In "conventional customer enabled maintenance systems, a connection is made with a large legacy system via a dial-up connection from a customer owned personal computer." EX1013, 1:26-28. Notably, "dial-up access requires custom software on the customer workstation to provide dial-up services, communication services, emulation and/or translation services and generally some resident custom form of the legacy application to interface with the mid-range or mainframe computer running the legacy system." *Id.*, 1:32-36. Therefore, it is desirable to "provide connectivity to enterprise legacy systems over the public Internet, as the Internet provides access connectivity [to the] world wide [web] via the TCP/IP protocol, without need to navigate various telephone exchanges, dialing standards or signal standards." *Id.*, 2:1-5.

133. Cogger focuses on a "Service Inquiry" ("SI") application, which is part of a suite of "telecommunications network applications" allowing customers to "manage their telecommunication assets, quickly and securely." *Id.*, 13:66-14:4; 4:65-5:1. The suite of applications are interoperable and "leverage existing

infrastructure services.” *See id.*, 13:66-14:17.

134. With respect to SI, Cogger explains that trouble tickets were already being used in the telecommunications industry’s fault management systems. *Id.*, 2:6-11. Trouble tickets “represent[] the set of ongoing and future repair efforts of the service provider.” *Id.*, 2:26-28. But customers want “an open access route to this information” and “to access their data over the internet.” *Id.*, 2:44-47. Cogger therefore provides a “system and method for allowing a customer to remotely access a service provider’s trouble ticketing system.” *Id.*, 2:39-43.

135. For SI, Cogger discloses an architecture including a client (Fig. 2, 20), a Demilitarized Zone (DMZ) (Fig. 2, 17), a MCI Intranet Dispatcher Server (Fig. 2, 26), MCI Intranet applications servers (Fig. 2, 30), and data warehouses and legacy systems (Fig. 2, 40). *Id.*, 6:8-14.

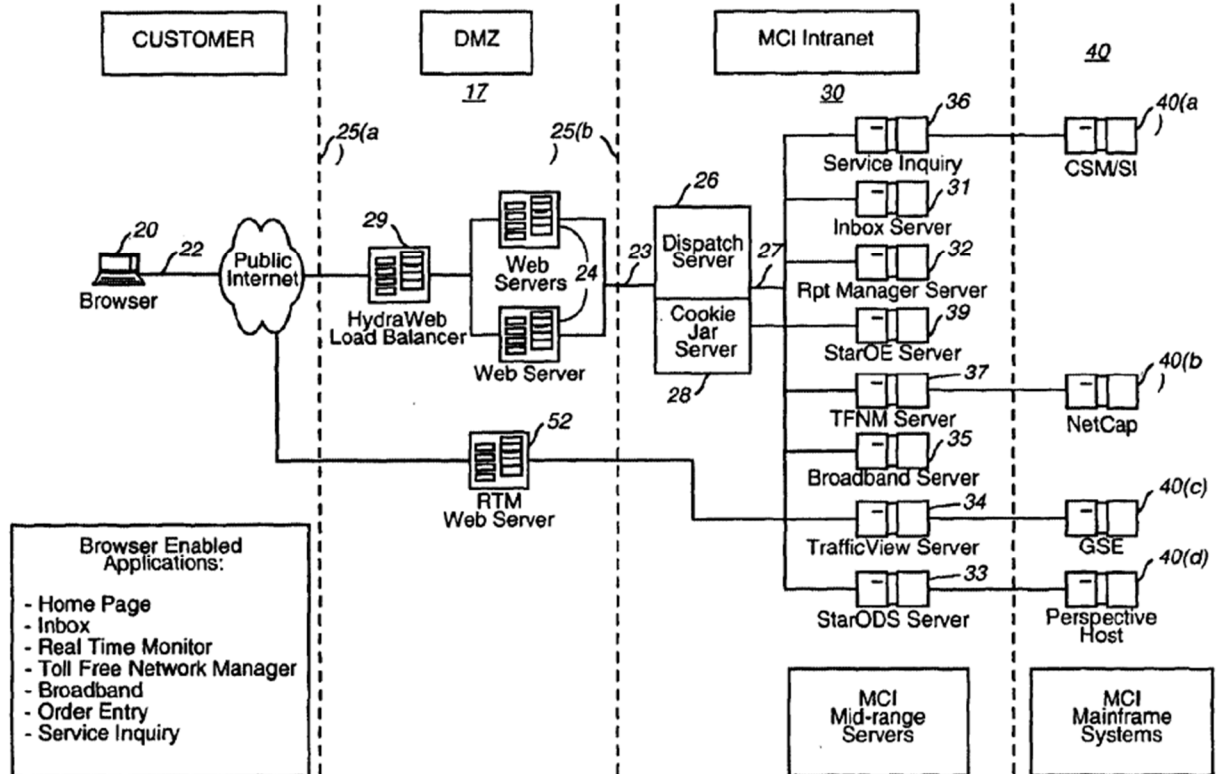


FIG. 2

Id., Fig. 2.

136. This architecture translates SI requests into requests processable by the legacy trouble ticketing system—*i.e.*, the Customer Service Management System (CSM). *Id.*, 15:58-16:31. Translation relies on various components within the architecture and the existing suite of telecommunications applications. *Id.*, 15:58-16:31. Once translated, an actual trouble request can be issued to the legacy CSM. *Id.*, 16:28-30. The legacy CSM processes the request and returns the

response data—passing it back to SI. *Id.*, 16:32-42. Because the legacy system already exists, Cogger does not actually disclose the legacy CSM’s features and functionality in detail.

137. Instead, Cogger only discloses the front-end interface for SI—explaining how it can be used to interact with legacy CSM. For example, a user may click a “Create” button to create a new ticket in SI. *Id.*, 21:55-57. Because the existing legacy CSM system typically requires an “Event-Type/Call-Type” for most transactions, SI provides the interface that “asks the customer the problem/identifier type and then narrow[s] down the problem by having the customer choose from a list of Product types, Service types and Trouble Descriptions.” *Id.*, 21:23-31. These data elements are passed on to the CSM legacy system to form the appropriate CSM request to open a Trouble Ticket. *See id.*, 21:41-57. Thus Cogger does not disclose how to implement the legacy CSM itself, nor how to use such a system to update filters in a completely different environment.

VII. THE PETITION FAILS TO ESTABLISH A REASONABLE LIKELIHOOD THAT THE ASSERTED REFERENCES RENDER ANY CLAIM UNPATENTABLE

A. Multiple Features Present in Every Claim Are Missing from Petitioner’s Asserted References

138. The claims of Schneier contemplate two separate levels of analysis of status data after that status data has been positively and negatively filtered. These two separate levels of analysis help determine whether unknown and new threats are reflected in the status data that did not match positive and negative filters.

139. Duvall and Chu fail to teach even one level of analysis after filtering, let alone the two-level process required by the claims (first at the probe to identify potential events, then transmitting to an analyst system, *e.g.*, at a SOC for a confirmation step).

140. Both references are also completely silent as to any dynamic update capability.

1. Duvall and Chu Lack Analysis of Post-Filter Residue Status Data to Identify Potential Security-Related Events

141. Neither Duvall nor Chu performs a first analysis, after filtering, on post-filtering residue, to flag potential events, at the probe, before transmission on for more detailed analysis.

142. This first-level of analysis to flag potential events takes place at the probe. EX1001, 3:17-21.

143. The claim language appears in element b of claim 1 as follows:

analyze status data to identify potentially security-related events represented in the status data, wherein the analysis includes filtering followed by an analysis of post-filtering residue, wherein the post-filtering residue is data neither discarded nor selected by filtering[.]

EX1001, 33:30-35 (claim 1) (emphasis added).

It can also be found within the first step of independent method claim 18 as follows:

creating an event record for information received about an identified potentially security-related event occurring on the network, wherein the potentially security-related event is identified by filtering followed by an analysis of post-filtering residue, wherein the post-filtering residue is neither discarded nor selected by the filtering[.]

EX1001, 34:25-30 (emphasis added).

144. Based on these limitations, the independent claims of Schneier all require the same thing. After filtering, the post-filtering residue status data, which was neither discarded nor selected (by positive and negative filtering), is analyzed, at the probe, to identify a potentially security-related event.

145. Neither reference cited by Petitioner provides even the slightest suggestion to do that. Instead, both Duvall and Chu only teach filtering for things that are already known. Neither of them analyzes any data to identify *potential* threats after filtering (*i.e.*, previously unknown threats from the post-filtering residue).

146. Duvall never “identifies potentially security-related events.”

Petitioner improperly equates Duvall’s subjective-criteria based filtering, which only matches to content known by the user to be subjectively objectionable, to an analysis that identifies events that are potentially objectively malicious. *See* Petition 33-36. But Duvall uses filters to censor data based on the user’s subjective preferences; it never analyzes any data for potentially security-related events. EX1004, 2:12-16.

147. Whether an identified event is in-fact potentially security-related is an objective question relevant to network security. Objective analysis is needed to elevate the data not matching any known filtering criteria (*i.e.*, unknown) to “potentially” security-related. But that sort of objective determination makes no sense for Duvall where the question of censorship is purely subjective. Duvall is limited to filtering out the known objectionable/non-objectionable material, which is determined based on user preferences. The status of Duvall’s data (including data “neither ALLOWED nor BLOCKED”) is always known to the user because the user subjectively pre-determines whether it is objectionable. *Id.* A POSA would not use Duvall to detect unknown threats. There is no teaching in Duvall as to how any data could be elevated from unknown to the level of a potential security-related event.

148. Petitioner argues that Duvall contemplates “unknown” data because Duvall’s system has yet to know which filter will apply before it has finished the filtering process. But that sort of “unknown” would not call for any additional objective analysis—Duvall’s system simply needs to be allowed to finish its filtering process. Duvall never contemplates events that are unknown to the user, such that the user would not be able to capture that content with filters. Duvall never suggests an objective analysis could elevate the unknown data into “potential” events at a first-level or be confirmed at a second-level. Indeed, Duvall never needs that sort of analysis because the user subjectively makes that determination instead.

149. Notably, Petitioner argues that certain filtered content “can carry viruses or malware.” Petition, 26. But that still does not mean Duvall teaches or even suggests “**identifying**” any “potential security-related events.” And even if a security-related event could tag along with subjectively objectionable content, filtering that content would not amount to identifying the security-related events.

150. Indeed, security-related events could tag along with virtually any content—even non-objectionable content. For example, a web interface for accessing an email with no objectionable content could contain an attachment that had malware. Whether the content is subjectively objectionable has no connection

to whether the content actually carries a virus or malware. The two are not directly related. Therefore, this would not motivate a POSA to alter Duvall as Petitioner proposes.

151. Petitioner also argues that it would be obvious to use Duvall for security to block malware. Petition, 26. But a POSA would recognize that Duvall's filtering system is generally unsuitable to monitor network security.

152. In addition to the above, Duvall does not analyze post-filtering residue. The data PAN identifies in Duvall is partially filtered and is still being processed by a common and pre-defined sequence of filters (*i.e.*, mid-filtering data). Duvall discloses a single filtering process that is always applied in a pre-defined sequence, and arbitrarily dividing that sequence mid-filtering does not somehow create the required distinct positive filtering, negative filtering, and post-filtering analysis steps. Before that filtering process is complete there cannot be post-filtering residue. And rather than filtering **out** (positively and negatively) data matching what is known, Duvall just searches for the single filter that should act.

153. Duvall applies a single set of filters from its filtering database to each data transmission—acting based on the first match. *See* EX1004, 4:22-37. Duvall suggests efficiently ordering filters with direct-actions first, followed by those with deferred-actions. But filter order is primarily dictated by the user's desired

censorship. For example, a user may want to block access to all but a few sites.

Id. 7:7-8. In that case the user may use a default BLOCK filter. But if the user also wants to replace certain content from sites while allowing the rest of the content, then the filters may need to be ordered so that an ALLOW direct-action filter is preceded by a deferred-action filter. This way, the deferred-action filter can replace certain content from a site. And content from the site not matching the deferred-action filters can be allowed by the direct-action filter rather than blocked by the default filter.

154. Therefore, Petitioner’s characterization of Duvall’s deferred-action filter as analysis of post-filtering residue is strained because it only applies when the filters happen to be ordered a specific way—and even then, Duvall is still in the process of searching for a matching filter, meaning filtering is not complete. *See id.*, 4:65-5:7. Any mid-filtering data still being considered is not yet “post-filtering” data, much less analysis of post-filtering residue status data.

155. Petitioner mischaracterizes the deferred-action filter’s filtering criteria as “techniques for analyzing residue data.” Petition, 18. (citing EX1004, Fig. 4). As explained above, the data passing the direct-action filters in Duvall is not unknown residue to be analyzed. *See id.* And the latter filters are only finishing Duvall’s process for filtering what is already known.

2. Duvall and Chu Lack a Second-level of Analysis By an Analyst System

156. Duvall and Chu also fail to disclose the second-level of analysis by at analyst system of potential events identified in the first-level of analysis. Having a first-level analysis to identify potential threats is a prerequisite for the second-level of analysis, which confirms those threats. Indeed, if a potential event is never identified at the first-level, there is nothing to confirm at a second-level.

157. The relevant language is in, element c of claim 1:

transmit information about the identified events to an analyst system associated with the security monitoring system[.]

EX1001, 33:36-38 (claim 1) (emphasis added); EX2001, ¶¶18, 67.

158. In my view, neither asserted reference provides even the slightest suggestion to transmit information about previously identified events from a first-level of analysis to an analyst system where the analyst can determine whether the potential events are actual events and/or resolve them. *Id.* To the extent there is any sort of analysis taught by the references, analysis could only relate to the first-level analysis—and not the second. The first-level analysis occurs after filtering, while the second-level analysis considers the results of the first-level analysis. Since both Duvall and Chu only disclose filtering, any analysis thereafter would—at best—relate to the first-level analysis that occurs after filtering. Neither

reference teaches transmitting information about the results of the first-level analysis to an analyst system.

159. Duvall makes no mention of anything that could be considered a second-level of analysis. Petitioner relies on Chu to teach a second level of analysis (Petition, 38-41), but, like Duvall, Chu never discloses any second-level analysis. Indeed, Chu cannot teach “transmitting information about **said identified events**” from a first-level analysis when there was no first-level analysis to begin with. EX1001, 33:36-38 (claim 1).

160. Chu does not “transmit[]” any information to an analyst system. Nor does Chu examine any information about potential events (*e.g.*, for a confirmatory step) that would suggest transmitting that information to an analyst system. Chu only filters URLs with whitelists and blacklists to automate a trust decision process. *See* EX1005, 44. If the trust decision cannot be automated because the URL is neither in the whitelist nor the blacklist, then Chu simply returns control back to the end-user—alerting them that no decision could be made. *Id.*

161. Chu characterizes this as “user intervention” because the system cannot automate the decision here—instead, giving control back to the end-user. *See id.* Chu’s “user intervention” is not the same as “transmitting information about said identified events to an analyst system.” *See* Petition 22-24; 38-39.

Alerting an end-user about an inability to make a trust decision is different than transmitting information about identified potentially security-related events to an analyst system (*e.g.*, for an independent confirmatory analysis). *See, e.g.*, EX1001, 2:43-64. Indeed, Chu only discloses REFEREE returning “*unknown*.” EX1005, 43-44. Instead, it is the application that prompts the user because the “unknown value returned by REFEREE forces the host application to decide what action (if any) should be taken.” *Id.* 31-32.

162. It would not even make sense to redirect the alert away from the end-user to a (non-existent) analyst system for any reason. “REFEREE is recommendation-based; the burden to enforce ... is on the application itself.” *Id.*, 51. Chu simply returns “*true*,” “*false*,” or “*unknown*,” and the application either automatically allows, automatically denies, or prompts the end-user to let the end-user decide. Redirecting one of the return values to an analyst system would force the end-user to wait for someone else (or another system) to respond to the prompt whenever accessing something not listed in the filters—something not taught by Chu. This would cause an enormous amount of delay making the resulting system effectively unusable. This makes no sense and is not taught by Chu.

- a. Chu and Duvall Never Teach or Suggest Any Previously “Identified Events” from a First-level Analysis

163. Schneier recites “transmitting information about said identified events.” EX1001, 33:36-38 (emphasis added).

164. Chu does not identify potentially security-related events. Chu only determines whether a file is trustworthy based on known criteria. When REFEREE returns “*unknown*,” Chu explains this merely means that REFEREE’s analysis was inconclusive. *See* EX1005, 43-44. But a POSA would not equate informing the user that an analysis was inconclusive by returning “unknown” to suggest any further analysis. This is especially true because control is being relinquished to the end-user rather than someone trained in network security. Chu holds little interest in the “unknown” data, instead focusing on preventing the data from ending up in that category (*i.e.*, Chu is just focused on positive and negative filtering).

165. As Chu itself explains, its trustworthiness determination is not the same as identifying a potential security-related event or even identifying an actual security event. *Id.*, 11. Petitioner effectively treats the data that triggered the “*unknown*” response by Chu as an identified potential event. But that makes little sense because Chu only returns an “unknown” response but does not identify anything out of that “*unknown*” data that could rise to the level of a “potential” security-related event.

b. Chu Never “Transmit[s] Information” After Filtering
Besides REFEREE’s “unknown” Response

166. Even if the “unknown” data (*i.e.*, data triggering REFEREE’s response of “unknown”) somehow was associated with an identified potential security event, Chu’s “user intervention” still would not involve transmitting any information about that event to anyone. *See* Petition 22-24; 38-39. Petitioner mischaracterizes Chu’s “user intervention” as somehow teaching the transmission of data to a human for a confirmatory analysis. *See* EX1005, 40. But at best, Chu’s sample policy only returns the response “unknown” and defers control back to the original application, which only alerts the end-user of the lack of any determination. *See* EX1005, 43-44.

167. Chu never discloses that REFEREE would ever transmit the “unknown” data itself to any human at an analyst system. *Id.* Chu only discloses that REFEREE would return a response of “true,” “false,” or “unknown.” *Id.* While the end-user provides data to REFEREE, REFEREE only returns the response “unknown” for the data not matching any filters. That response cannot be characterized as “information about said identified events” when no potential events were identified out of the data triggering the “unknown” response.

168. Therefore, Chu’s “user intervention” alert does not transmit

information about the “*unknown*” data to a human—let alone information about previously identified potential events.

c. Neither Duvall Nor Chu Discusses an Analyst System to Which Information Could Be Transmitted

169. At best, Chu’s “user intervention” teaches prompting the end-user for attention. *See* EX1005, 44. Prompting the end-user for attention merely lets the end-user decide whether to access the file. No analyst system is involved whatsoever. The end-user is not a trained security professional and therefore could not be the analyst, nor would the original application be an analyst system. *See, e.g.,* EX1001, 2:43-50; 7:10-23. Indeed, the end-user’s application, which includes the probe, cannot be the claimed analyst system. There is no teaching—whatsoever—in Chu or Duvall about an analyst system.

170. Petitioner suggests that a trained network administrator from a specialized third-party network-security service could serve as the analyst system. *See* Petition, 23. In support, Petitioner cites to a reference not actually relied upon in the proposed grounds for unpatentability. *See* Petition, 23 (citing EX1037). But neither the Petition nor this third reference discloses how to introduce the analyst or associated analyst system into either Duvall or Chu, much less *why* it would have been obvious to (1) shift the prompts away from the end-user to the

administrator, (2) modify those alerts to include information about identified potential security-related events rather than returning “*unknown*”, or (3) have the administrator that handles filters do double-duty to analyze security-related events. Petitioner simply makes this up using Schneier as a template. Similarly, the Petition never explains how these modifications would be made, which combinations would yield a predictable result, or how the modified system would even operate.

3. The Limitations Regarding Receipt of Feedback at the Probe to Dynamically Enhance Its Analytical Capabilities Are Missing from the Asserted References

171. Independent claim 1 of Schneier contains a limitation directed to updating the analytical capabilities of the probe, dynamically, through the receipt of feedback at the probe. The relevant claim language appears in elements d and e of Schneier claim 1 as follows:

- d) a receiver for receiving feedback at the probe based on empirically-derived information reflecting operation of the security monitoring system; and
- e) a modification control system for dynamically modifying an analysis capability of the probe during operation thereof based on the received feedback.

EX1001, 33:39-45 (reciting claim 1).

172. Duvall and Chu both fail to teach this dynamic feedback loop.

Importantly, the term “dynamically” was previously construed to mean “during actual operation, rather than offline.” EX1013, 2. Petitioner wrongly argues that Duvall teaches a dynamic feedback loop, *see* Petition 41-43, but Duvall never discloses “dynamically” modifying any analysis capability.

173. Petitioner speculates that a POSA would have recognized that the analysis capability of Duvall is modified dynamically “because taking the system offline each time an update was required would be disadvantageous.” Petition, 43. But just because taking the system offline would be disadvantageous does not mean Duvall teaches modifying its analysis capability “during actual operation, rather than offline.” *See* EX1013, 2. Indeed, Duvall contains no such teachings.

174. Petitioner argues that Duvall “avoid[s] this disadvantage” by “provid[ing] updates online so that the system can adapt as new sites and servers are added to the Internet.” *Id.* (citing EX1004, 2:16-18). But the word “online” in Duvall means “over the Internet”—not while the system is operational. *See* EX1004, 2:16-18; *see also id.*, 7:17-29 (explaining that “[b]ecause Internet sites are being added to the Internet at a fast rate” an update mechanism is provided to “locate a particular ... update server [] over the Internet”). Updating over the

Internet (*i.e.*, online) neither teaches nor suggests “dynamically” modifying an analysis capability “during actual operation, rather than offline.” *Compare, id.*, 7:17-29, *with*, EX1013 at 2.

175. Petitioner also argues Duvall’s filtering system accommodates dynamic updates because it uses a database. But a database does not inherently allow for dynamic updates. And Duvall actually teaches that the database cannot be updated during actual operation. *See* EX1004, 8:12-14. Duvall states that “during a session over the Internet, a user can copy URLs for **later editing**.” *Id.* (emphasis added). Duvall therefore recognizes that the filters cannot be editing while it is in operation. Instead, the user would need to wait until after the browsing session—when the system can be taken offline—to update the filters the user copied.

176. Duvall also teaches pre-processing the database in a way that makes it incompatible with dynamic updates. The process requires “analy[sis] to determine the frequency of combinations of characters” and the creation of two additional tables for searching. *Id.*, 8:39-61.

177. Duvall’s server-based system in fact teaches against “dynamic” modification because Duvall consciously sacrifices the potential to have dynamic updates to make the server-based system efficient enough to work. Indeed,

whenever the filters are edited, the system would need to perform this analysis and create a new pair of tables before searching is possible using this modified algorithm. During this time, the system would necessarily be inoperable until the tables are generated.

178. Modifications to Duvall's filters are also based on subjective views; not on "feedback at the probe based on **empirically-derived information** reflecting operation of said security monitoring system." Duvall's filters are modified based on the user's "belie[f] that the filtering system is filtering too much or too little." EX1004, 8:7. But subjective beliefs and the user's "individual tastes and sensitivities" are not objectively verifiable. *See id.*, 2:12-16. Duvall's feedback is therefore based on the exact opposite of "empirically-derived information." In fact, the subjective nature of Duvall precludes the empirically-driven approach disclosed by Schneier.

B. There Is No Basis for Combining Duvall and Chu, No Details How to Do So, and No Reasonable Expectation of Success

179. There is no reason a POSA would be motivated to combine Duvall and Chu to arrive at anything begins to approach the claimed invention. Not only are Petitioner's suggested motivations expressly contradicted by the disclosures of Duvall and Chu, but Chu does not even teach the features (*i.e.*, techniques for

determining whether to block or allow residue) Petitioner attempts to combine into Duvall.

180. Duvall and Chu have completely different focuses: They attempt to solve different problems in different ways compared to each other and Schneier.

Duvall focuses on censorship, whereas Chu focuses on trust management.

Compare, EX1004, 1:10-29, *with*, EX1005, 11-12. The mere fact that Duvall and Chu both use filters is insufficient to show why or how a POSA would have combined their teachings, much less in a manner involving a network security approach as claimed in Schneier.

181. Petitioner never explains why a POSA would combine a censorship mechanism with a trust management system to create a security monitoring device. Duvall and Chu filter different things for different purposes in different ways from both each other, and from Schneier's claims. Accordingly, Petitioner's proposed combination is flawed for the many reasons discussed below.

1. Duvall Is Unsuitable for Use as a Network Security Monitoring System

182. Petitioner argues that Duvall recognizes its own system could "filter too much or too little." Petition, 20, 22-23, 38, 40-41. Petitioner's expert argues that a desire to "ensure that the system correctly filters data on a consistent basis"

would provide the motivation to “seek additional techniques to more accurately resolve the status of residue status data transmissions to ensure transmissions are correctly blocked or allowed.” EX1003, ¶103. But this argument already presumes that Duvall would be used for security in the first place, which it is not.

183. For security monitoring systems, like that in Schneier, the goal is to detect and respond to network threats. *See* EX1001, 1:18-20. Schneier detects threats using objective analyses and improves using empirically-derived information, which is fundamentally unlike Duvall’s methods for determining whether something is subjectively objectionable.

184. On the other hand, Duvall is only concerned with censorship. Duvall’s user is not uncertain about the status of any data matched by the default filter such that further confirmatory analysis would be needed. *See* EX1004, 5:1-7. Instead, Duvall intentionally relies on the default filter to allow or block transmissions (*e.g.*, where the user wants to block access to all but a few sites). *See id.*, 7:7-8. It is nonsensical to seek objective techniques to resolve the status of this data when what matters is subjective in nature and is pre-determined in advance.

- a. Duvall Is Unsuitable for Network Security Because It Is Predicated on Subjective Preferences Predetermined by the User

185. Petitioner’s proffered desire to “seek additional techniques to more accurately resolve the status of residue” actually suggests Duvall should not be used for security in the first place. Security requires analysis that uses objective criteria (*i.e.*, is it a threat?)—which is exactly what Duvall lacks. Duvall never teaches or discloses “ensur[ing] content that implicates security concerns does not reach recipients.” *Compare* Petition, 14, *with*, EX1004. Duvall never mentions security at all. Far from it, Duvall explicitly teaches that its filtering system filters content that is “personally objectionable” based on the user’s subjective “tastes and sensitivities.” *Id.*, 2:12-16.

186. Petitioner never explains why Duvall should be altered for use in such an unintended way. Petitioner asserts that a “POSA would have recognized that data transmissions from unknown or questionable sources can carry viruses or malware.” Petition, 33. But that still does not explain why Duvall should be transformed to detect and block those viruses instead of subjectively objectionable content. Indeed, a POSA would also have recognized that data from known and apparently trustworthy sources could also carry viruses or malware. Petitioner’s rationale is invalid. This would not motivate a POSA to alter Duvall into a security monitoring system as Duvall would be both under and overinclusive. A POSA cannot simply fix those problems with a reasonable expectation of success

without more. Moreover, rather than teaching how to identify threats—Duvall focuses on censoring content in different ways (*e.g.*, by replacing specific objectionable words).

187. Duvall notes in passing that its “implementing software ... can [] be incorporated into a firewall,” but never suggests it would become a security monitoring system. EX1004, 1:61-63. Instead, Duvall acknowledges that it could be convenient to enforce the censorship from a central point through which internet traffic flows—the firewall.

188. Because Duvall is not focused on security, Duvall also never contemplates an objective analysis that could identify potential security-related events. Not only is Petitioner’s asserted motivation contrary to Duvall’s explicit disclosure, but this supposed desire to seek additional techniques only arises because Petitioner is attempting to transform Duvall into something Duvall itself never intended. Duvall is primarily focused on what happens after a transmission is matched to its filter (*i.e.*, enabling different ways of blocking) but not on different ways to detect threats that are otherwise unknown to the user.

189. Petitioner suggests that a POSA would be motivated to look for Chu to add a confirmatory step to Duvall after fundamentally transforming the nature of Duvall into a security device that needs Chu. This reasoning is circular and clearly

driven by hindsight.

b. Duvall Is Unsuitable for Computer Network Security
Because It Is Not Concerned with Security Threats

190. Duvall demonstrates that a censorship-based system operates fundamentally differently than a security-focused system. For example, Duvall only suggests monitoring transmissions related to a particular type of transport protocol commonly used to receive content over the Internet—Transmission Control Protocol (TCP). See EX1004, 2:42-62; 3:50-63.

191. Duvall never mentions any other transport protocol, where network security threats also exist. Data is not exclusively transmitted over the Internet with TCP. Network security systems often require inspecting raw packet data or at least a consideration of data from other alternative transport protocols—e.g., UDP. Yet Duvall expresses no interest in monitoring anything else. Indeed, it would not make sense to use Duvall for that purpose either given its focus on ways to censor material (e.g., replacing objectionable words in received content) rather than detecting security threats. As a security monitoring system Duvall leaves much to be desired, and a POSA cannot simply transform Duvall into a security monitoring system with a reasonable expectation of success based on what is disclosed in Duvall and Chu.

192. Duvall is unconcerned with the analysis of raw packet data and is only meant to operate on the presentation and application layers (layers 6/7). *See* EX1004, 2:34-3:19; 6:10-28 (showing how Duvall operates on HTTP data). This makes Duvall wholly unsuitable for detecting network threats from any other layers. While this is sufficient to censor content, it is greatly inadequate for monitoring network security.

193. Duvall also never contemplates a need for any sort of objective analysis to identify threats—which is the essence of network security. This is because Duvall assumes that the status of data as good or bad is subjectively predetermined by the user. EX1004, 2:12-16; *see also id.*, 3:25-29. So, instead of focusing on different analysis techniques or ways to detect unknown threats, Duvall focuses on different ways to censor the material already captured by its filters (*i.e.*, different blocking actions). *See, e.g., id.*, 3:20-41. Therefore, Duvall does not filter as part of any analysis related to security; it only filters to censor. EX1004, 1:12-29.

194. Moreover, Duvall's disclosure is more concerned about an end-user bypassing the censorship than it is with any network security concerns (*i.e.*, it secures the network against internal users not external threats). *See, e.g., id.*, 11:19-20. Indeed, Duvall is completely silent about any potential malware,

viruses, or other network security-related concerns. Instead, Duvall devotes much of its disclosure to teaching how to implement patching code that cannot be easily disabled by the end-user. *Id.*, 10:35-11:19. But those concerns about the end-user are not the threats that a network security monitoring device is concerned of.

195. From a network security monitoring point of view, Duvall's concerns are backwards—in a sense treating the end-user as the primary threat instead of actual threats, such as malware. EX1004, 10:34-11:19. Duvall's system is therefore undesirable if the goal is to provide network security monitoring system. A POSA would not have been motivated to look for Chu nor combine it with Duvall to arrive at the security monitoring system claimed by Schneier.

2. Petitioner's Motivation Relies on a Mischaracterization of Duvall's Concerns about Subjectively "Filtering Too Much or Too Little"

196. Although Duvall discloses "filtering too much or too little," (*i.e.*, not capturing all content that would be objectionable to a user or capturing non-objectionable content) this is only because the filtering criteria was not adequately configured. Duvall never suggests—as Petitioner implies—that its filtering systems would be inaccurate because the user was uncertain whether he believed something was objectionable. *See* Petition, 20.

197. Duvall never contemplates that any technique or analysis could assist

with that determination. Instead, Duvall’s concern is purely subjective—based on whether the “**user believes** that the filtering system is filtering too much or too little.” EX1004, 8:6-8 (emphasis added). Duvall only recognizes that the filters may need to be tuned to accommodate individual tastes and provides the solution accordingly. Duvall’s concern is whether its filters can accurately capture everything the user knows (*i.e.*, subjectively predetermined)—not identifying or resolving anything potential, uncertain, or unknown to the user.

198. Accordingly, Duvall provides an “editing manager” for users to tune the filters according to their subjective preferences. EX1004, 8:6-10. Duvall presents this as the solution to the problem of “filtering too much or too little.” *Id.*

199. Given that Duvall already provides the solution, a POSA would not be motivated to seek any objective analysis techniques as Petitioner suggests. *See* Petition, 23-24, 38-41. Rather than introducing additional analysis techniques (which do not exist given the subjective nature of the determination), Duvall’s solution embraces the subjective nature of its filtering system by simply allowing users to modify the filters accordingly.

200. For Duvall, additional objective analysis makes no sense and would not resolve the problem actually contemplated by Duvall.

3. Duvall and Chu Use Filters in Different Ways and for Different

Purposes

201. Duvall and Chu use their filters for different purposes. The sample Chu policy Petitioner relies on is meant to address issues of code-signing within trust management. EX1005, 43. Unlike Duvall, Chu’s REFEREE objectively determines the status of incoming data as trustworthy or untrustworthy using filters to determine whether an associated URL corresponds to a known trusted/untrusted entity. *See id.*, 43-44. In contrast, Duvall merely filters to capture material that is “personally objectionable, whether that material is sexually explicit, violent, politically extreme, or otherwise, depending on that user’s individual tastes and sensitivities.” EX1004, 2:13-16.

202. So, whereas Chu uses filters to determine trust, Duvall only teaches a filtering system for censoring—but makes no objective determinations. Duvall never goes as far as using or suggesting how the filtering system could be used to perform any sort of analysis like identifying potential security-related events. Instead, Duvall focuses on providing different ways to censor the content after a match with its filters (*i.e.*, different blocking actions).

4. A POSA Would Not Combine Chu with Duvall Given the Differences In Concerns, Underlying Assumptions, and Goals.

203. Duvall and Chu’s perspectives on “trust” are diametrically opposed.

To the extent that Duvall has trust concerns, they are primarily with end-users bypassing the censorship. Duvall thus assumes the end-user is the primary “threat” and suggests several counter-measures such as password protecting the filter database and ensuring that even “if a[n] [end-]user kills the program by re-booting, the blocking mechanisms are left intact.” *Id.* 11:18-20. In contrast, Chu trusts the end-user—deferring ultimate decisions about whether to access a file back to the end-user when REFEREE cannot automatically decide. *See* EX1005, 43-44. Because Chu is focused on trust management, its primary concern is whether to trust a third party. *See id.* Duvall lacks any shared concerns about trust and, likewise, Chu lacks the concerns found in Duvall.

204. In my view, given these differences, a POSA would not have been motivated to combine Chu with Duvall.

5. Even If a POSA Was Motivated to Search for Techniques to Resolve the Status of Residue, Chu Does Not Teach Any Such Techniques.

205. Petitioner asserts that a POSA would have been motivated to “seek techniques for more accurately resolving the status of residue data transmissions.” Petition, 20. But even that were the case, Chu does not teach any such techniques. *See* EX1005, 43-44.

206. Even applying Petitioner’s own rationale, a POSA still would not

have been motivated to look to Chu to address this purported desire. Petitioner's expert contends that Chu provides "techniques for determining whether to block or allow residue data." EX1003 at 104. But that statement is unsupported by Chu. At best, Chu provides techniques to determine the trustworthiness of incoming data using whitelists and blacklists. *See* EX1005, 43-44. But Chu is silent about any techniques to analyze residue from those filters. *See id.*

207. Petitioner also argues the supposed desire for Chu comes from a desire to ensure Duvall's filters are accurate—*i.e.*, to tune and edit the filters. *See* Petition, 20. But the policies Chu discloses are never used for that purpose. *See* EX1005, 43-44. Nor does Chu apply "user intervention" for that purpose. *See id.* Instead, Chu teaches letting an end-user decide (*i.e.*, user intervention) whether to access a file. *See id.*

6. A POSA Would Not Have Arrived at the Claims Given Duvall and Chu.

208. Even with Duvall and Chu before him, there is simply no way for a POSA to arrive at the claimed invention. Petitioner's rationale is internally inconsistent resulting in a proposed combination that makes little sense. Petitioner's expert suggests a POSA would be motivated to "incorporate Chu's user intervention into Duvall's system for data transmissions not matching any of

Duvall’s ALLOW or BLOCK (i.e., positive or negative) filters, nor resolved by Duvall’s post-filtering residue analysis.” EX1003, ¶107.

209. Therefore, Petitioner modifies Duvall in at least three ways: to include an analyst system, transmit information to that analyst system, and then have that analyst system provide feedback to the probe based on the analysis. But Chu does not teach those features. Chu never discloses the type of “user intervention” that Petitioner is attempting to introduce into Duvall and Petitioner’s resulting combination would be inoperable.

a. Chu Does Not Disclose the Type of “User Intervention” that Petitioner Seeks to Introduce into Duvall

210. The “user intervention” Chu discloses is not as Petitioner represents. *Compare* Petition, 22-24, *with* EX1005, 44. Petitioner suggests Chu’s “user intervention” somehow teaches transmitting residue to an analyst system for further analysis. *See* Petition, 22. But that is not what Chu means by “user intervention.” *See* EX1005, 44. According to Chu, when no filters are matched, REFEREE just returns control back to the application—prompting the end-user for attention. *See id.*, 32; 44. Chu merely recognizes the common-sense notion that filters can only make automatic decisions about data matched by the filters. Chu is not suggesting any further analysis or any intervention with respect to REFEREE.

b. Petitioner Applies “User Intervention” in Ways Never Contemplated by Chu

211. Chu never applies “user intervention” the way Petitioner attempts to modify Duvall. Petitioner applies “user intervention” on different data, for a different purpose, by a different human compared to what is taught by Chu.

212. Chu never uses “user intervention” on any identified potential events (*i.e.*, results of a first-level analysis). Instead, Chu’s “user intervention” only returns “*unknown*” after filtering the URL, before any first-level analysis could be performed to identify potential events. Therefore, Chu only makes a single trust/do-not-trust/unknown decision and is far from any second-level analysis by an analyst system.

213. Even under Petitioner’s formulation of Duvall-Chu, this combination still makes no sense. Petitioner suggests Chu’s “user-intervention” should be applied to data “not matching any of Duvall’s positive/negative filters at step 104, nor resolved by Duvall’s post-filtering residue analysis.” Petition, 22. Petitioner also asserts that “information is only provided to the trained professional if the data is unresolved residue data—meaning that the data did not match any allow/block filters and passed through subsequent residue analysis (*e.g.*, keywords and pattern matching) without any resolution.” Petition 41; *see also* EX1003, ¶146.

214. Even under this formulation, the “unresolved residue data” cannot also be “an identified potential event.” *See* Petition, 38-41. Though there is an amount of uncertainty tied to both types of data, no analysis has been performed on the purported “unresolved residue data” to elevate it from entirely unknown to the level of an identified potential event.

215. Chu also does not tune filters with “user intervention.” Chu merely lets the end-user decide whether to access the file. *See* EX1005, 44. At best, this would relate to blocking or allowing but not tuning. If anything, Chu only properly replaces Duvall’s default filters by letting the end-user decide what to see when no other filters match with the content. But that would undermine Duvall’s system (effectively letting the child control what should be censored as opposed to the parent). Therefore, there is no proper way to combine Duvall and Chu to arrive at Schneier.

216. Petitioner incorrectly argues Chu’s “user intervention” is intended to help an administrator tune the filters more accurately. Petition, 20; 41. But this is not supported by Chu. Chu never teaches how this could be used to update filters—let alone to help an administrator (or analyst system) tune them. On the contrary, the end-user’s discretion to access a file is a very poor basis for tuning filters if security is truly a concern. *See also* EX1003, 108. The end-user is not a

trained security personnel and—like in Duvall—the end-user may just be a child.

Basing the filters based on the end-user’s preference rather than the user’s preference would undermine the system.

217. Schneier discloses remote analyst systems at the SOC confirming whether the previously identified potential events are security-related, which makes sense because an analyst is better positioned to determine whether a potential event is actually a security threat compared to an end-user. EX1001, 2:18-24. Schneier provides the remote analyst system with access to more information gathered across various probes. *Id.*, 10:16-30.

218. In contrast, Duvall’s criteria are subjectively pre-defined, meaning trained security analysts (and analyst systems) are not even needed or appropriate. EX1004, 2:12-16. Given Duvall’s concerns about inefficiencies related to a remote database (*i.e.*, delays caused by moving the database to a remote server)—remote administrators would make even less sense. *See id.*, 8:39-42. Indeed, this would only cause more delays in Duvall’s system. Moreover, because Petitioner argues “information is only provided to the trained professional if the data is unresolved residue data,” Petition 41, that means information about what was matched is not sent to administrators. Without information about what was actually matched, such administrators would not even have the information

necessary to tune pre-existing filters. At best, new filters could be created to capture what was not already matched, but existing filters that captured too much would not be tunable.

219. Petitioner asserts that “user intervention” should be combined into Duvall such that it is performed by an analyst rather than the end-user. Petition, 23-24; 39-40. But this is inconsistent with and not taught by Chu. Chu does not mention any analyst or analyst system, and thus Petitioner therefore argues Duvall’s filtering system would be controlled by a remote administrator. *See* Petition 23-24. But Petitioner never explains how “user intervention”—as taught by Chu—could assist administrators.

220. Instead, Chu’s intervention is performed by the end-user. *See* EX1004, 44. Chu does not support making this sort of modification to Duvall because it is inconsistent with Chu. If “user intervention” involved an administrator instead, Chu’s invention would fall apart due to the flood of prompts faced by the administrator. Therefore, requiring intervention by an administrator is inconsistent with and never suggested by Chu. Indeed, if every end-user needed to wait for the administrator to respond to the prompt before accessing something not matched by a filter, this would lead to an unreasonable amount of delay when browsing the internet. The record provides no support for the modification to

Duvall suggested by Petitioner and appears to be driven purely by hindsight.

c. Petitioner's Resulting Combination Would Be Inoperable

221. Petitioner overlooks the importance of limiting what is sent to an analyst in Schneier—resulting in an inoperable combination. Schneier recognizes that the volume of residue data is unmanageable without first identifying the data that rises to the level of a potential events (to narrow what should be sent to the analyst system). *See, e.g.*, EX1001, 2:5-12, 1:36-39. This step significantly narrows down what is sent to the analyst system. Schneier does not simply transmit all the unknown and leftover data to an analyst system, but specifically identifies potential events and only transmits information about those potential events.

222. Yet Petitioner's resulting combination sends an administrator information about every transmission “not matching any of Duvall's positive/negative filters at step 104, nor resolved by Duvall's post-filtering residue analysis.” Petition, 22. Because Duvall never narrows down what should be sent, like Schneier does, Duvall's system results in the same flood of information that Schneier is designed to prevent.

223. Regardless of how Petitioner characterizes Duvall, its system only filters based on known criteria. Petitioner suggests transmitting all the information

that would reach the default filters to an analyst system. But the amount of data that would reach Duvall's default filters still far exceeds what is manageable. Schneier recognized that prior art systems filtering based on what was already known left millions of lines of audit data. EX1001, 2:8-10. Duvall's system falls squarely into this defect of the prior art and its deferred-action filters would not reduce that unmanageable volume. At best, it would just be layering additional filters of known criteria. So, regardless of whether Duvall-Chu can actually be combined or characterized as Petitioner suggests, the resulting combination Petitioner proposes would still flood the analyst system with an unmanageable amount of information.

224. As a practical matter, this combination would be inoperable. *Id.* It faces the exact problem that Schneier's secondary considerations addressed. *Id.* A POSA would therefore not be motivated to send an analyst system all the data from Duvall that would otherwise match the default filter.

225. There is no way to combine these references to arrive at these aspects of the claimed invention.

C. There Is No Way to Combine Cogger into Duvall-Chu To Arrive At Independent Claim 18

226. There is no motivation to combine Cogger into Duvall-Chu. Nor is

there a logical reason to use a remote interface for a pre-existing telecommunications trouble ticketing system for Duvall-Chu. Cogger simply allows customers to remotely access a pre-existing system through its disclosed interface. *See* EX1033, 2:39-43. But accessing a system remotely is not the same as using the underlying system to remotely manage a filtering database. Yet Petitioner modifies Duvall-Chu to do just that. Petition, 65-67.

227. Cogger is concerned with a legacy telecommunications trouble ticketing system that Duvall-Chu does not already have. A POSA would not be motivated or know how to adapt Cogger for use with Duvall-Chu, much less in a network security context. Cogger does not disclose the specifics of the ticketing system so that a POSA could implement one for Duvall-Chu. Instead, Cogger only teaches the interface to access that telecommunications ticketing system's pre-existing features. *See* EX1033, 2:39-43.

228. Petitioner's combination first requires Duvall-Chu to utilize remote management before a trouble ticketing system for that remote management even begins to make sense. But there is no motivation to do that in Duvall, Chu, or Cogger. Petitioner never provides insight into how this could be done, and also offers no motivation to do so.

229. Petitioner also provides no details about how Cogger could even be

combined into Duvall-Chu with a reasonable expectation of success. Cogger relies on a complex pre-existing architecture and a suite of other pre-existing telecommunications applications. *See id.*, 1:32-36; 6:8-14; 13:66-14:17. None of these systems exist in Duvall-Chu.

230. A POSA could not just adapt the pre-existing telecommunications trouble ticketing system—nor the entire suite of applications Cogger requires—into Duvall-Chu with any reasonable expectation of success. Merely implementing the user interface taught by Cogger would accomplish nothing without the underlying systems and applications. It would also not make sense to implement these telecommunications systems and applications into Duvall-Chu given how unrelated they are.

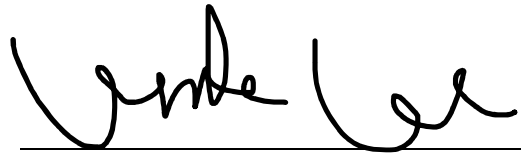
VIII. CONCLUSION

231. Petitioner's references therefore fail to disclose multiple claim limitations. The references cannot be combined into an operable whole, much less with a reasonable expectation of success and there is no motivation to combine them absent the use of impermissible hindsight.

I declare under penalty of perjury under the laws of the United States of America that all statements made herein based on my own personal knowledge are true and correct, that statements made on the basis of information and belief are believed to be true, and the opinions expressed in this declaration are my own. If called to testify as to the contents of this declaration, I could and would do so, competently.

These statements are made with knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under section 1001 of title 18 of the United States Code.

Dated: __August 17, 2023_____

A handwritten signature in black ink, appearing to read 'Wenke Lee', is written over a horizontal line.

Wenke Lee

Exhibit A

Wenke Lee
Professor and John P. Imlay Jr. Chair
School of Cybersecurity and Privacy and School of Computer Science
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280
wenke@cc.gatech.edu
<http://www.cc.gatech.edu/~wenke/>

Contents

EDUCATIONAL BACKGROUND	2
EMPLOYMENT HISTORY	2
CURRENT FIELDS OF INTEREST	2
I. TEACHING	4
A. Courses Taught	4
B. Curriculum Development	6
C. Individual Student Guidance	8
II. RESEARCH AND CREATIVE SCHOLARSHIP	11
A. Theses	11
B. Published Journal Papers	11
C. Published Books and Parts of Books	12
D. Edited Proceedings and Books	13
E. Conference Presentations (refereed)	13
E.1 with Proceedings (refereed)	13
F. Workshop Presentations (refereed)	28
F.1 with Proceedings	28
G. Other	29
H. Research Proposals and Grants	31
I. Honors and Awards	38
III. SERVICE	40
A. Professional Activities	40
A.1 Membership and Activities in Professional Societies	40
A.2 Journal Edit-oral Board Activities	40
A.3 Conference Committee Activities	40
B. On-campus Georgia Tech Committees	44
IV. NATIONAL AND INTERNATIONAL PROFESSIONAL RECOGNITION	46
A. Patents	46
V. OTHER CONTRIBUTIONS	48
A. Seminar Presentations	48

EDUCATIONAL BACKGROUND

Degree	Year	University	Field
Ph.D.	1999	Columbia University	Computer Science
M.S.	1990	The City College of New York	Computer Science
B.S.	1988	Sun Yat-Sen University, China	Computer Science

EMPLOYMENT HISTORY

Title	Organization	Years
John P. Imlay Jr. Chair	College of Computing	3/2016–present
Director	Institute for Information Security & Privacy (IISP)	7/2015–7/2021
	Georgia Institute of Technology	
Director	Georgia Tech Information Security Center (GTISC)	8/2012–6/2015
	Georgia Institute of Technology	
Professor	College of Computing	2/2009–present
	Georgia Institute of Technology	
Associate Professor	College of Computing	2/2005–2/2009
	Georgia Institute of Technology	
Assistant Professor	College of Computing	8/2001–2/2005
	Georgia Institute of Technology	
Assistant Professor	Department of Computer Science	8/1999–8/2001
	North Carolina State University	
Research Assistant	Department of Computer Science	1994–1999
	Columbia University	
Research Staff Member	IBM T.J. Watson Research Center	5/1997–8/1997
Member of Technical Staff	AT&T Research	5/1996–8/1996
Senior Software Analyst	Intergraph Corporation	1991–1994
Software Analyst	Intergraph Corporation	1990–1991

CURRENT FIELDS OF INTEREST

Information Security

Main research thrusts are: “Malware analysis and detection” – developing “live” and “safe” environment to uncover malware behaviors and machine-learning based techniques to automatically generate detection models and forensic analysis rules; “Adversarial Machine Learning” – studying the vulnerabilities of machine learning based systems due to adversarial manipulations and developing countermeasures; “Software debloating and protocol subsetting” – developing program and protocol analysis and rewriting techniques to remove unnecessary features and code from program binaries; “Privacy-preserving biometric-based authentication and search” – researching efficient methods to protect the privacy of biometric data while enabling important security applications including re-

mote authentication and video surveillance; “Adaptive analysis framework for advanced persistent threats” – developing game-theoretic approaches to model and analyze advanced persistent threats.

I. TEACHING

A. Courses Taught

<u>Semester/Year</u>	<u>Course</u>	<u>Number of Students</u>	<u>Comments</u>
Fall 2022	CS 4235/6035 Introduction to Information Security	184	At CoC
Fall 2022	OMS CS/Cybersecurity 6262 Network Security	251	Online
Fall 2022	OMS CS/Cybersecurity 6035 Introduction to Information Security	622	Online
Fall 2022	OMS CS/Cybersecurity 6264 Information Security Labs: System and Network Defenses	18	Online
Summer 2022	OMS CS/Cybersecurity 6262 Network Security	258	Online
Summer 2022	OMS CS/Cybersecurity 6035 Introduction to Information Security	334	Online
Spring 2022	OMS CS/Cybersecurity 6262 Network Security	300	Online
Spring 2022	OMS CS/Cybersecurity 6035 Introduction to Information Security	635	Online
Spring 2022	OMS CS/Cybersecurity 6264 Information Security Labs: System and Network Defenses	17	Online
Fall 2021	CS 6262 Network Security	40	At CoC
Fall 2021	OMS CS/Cybersecurity 6262 Network Security	360	Online
Fall 2021	OMS CS/Cybersecurity 6035 Introduction to Information Security	980	Online
Fall 2021	OMS CS/Cybersecurity 6264 Information Security Labs: System and Network Defenses	17	Online
Summer 2021	OMS CS/Cybersecurity 6262 Network Security	200	Online
Summer 2021	OMS CS/Cybersecurity 6035 Introduction to Information Security	460	Online
Spring 2021	OMS CS/Cybersecurity 6264 Information Security Labs: System and Network Defenses	25	Online
Spring 2021	OMS CS/Cybersecurity 6262 Network Security	300	Online
Spring 2021	OMS CS/Cybersecurity 6035 Introduction to Information Security	800	Online
Fall 2020	CS 6262 Network Security	40	At CoC
Fall 2020	OMS CS/Cybersecurity 8803 Information Security Labs: System and Network Defenses	25	Online
Fall 2020	OMS CS/Cybersecurity 6262 Network Security	260	Online
Fall 2020	OMS CS/Cybersecurity 6035 Introduction to Information Security	850	Online
Summer 2020	OMS CS/Cybersecurity 8803 Information Security Labs: System and Network Defenses	20	Online
Summer 2020	OMS CS/Cybersecurity 6262 Network Security	150	Online
Summer 2020	OMS CS/Cybersecurity 6035 Introduction to Information Security	450	Online
Spring 2020	OMS CS/Cybersecurity 8803 Information Security Labs: System and Network Defenses	40	Online
Spring 2020	OMS CS/Cybersecurity 6262 Network Security	360	Online
Spring 2020	OMS CS/Cybersecurity 6035 Introduction to Information Security	884	Online

<u>Semester/Year</u>	<u>Course</u>	<u>Number of Students</u>	<u>Comments</u>
Fall 2019	CS 6262 Network Security	43	At CoC
Fall 2019	OMS CS/Cybersecurity 6262 Network Security	156	Online
Fall 2019	OMS CS/Cybersecurity 6035 Introduction to Information Security	770	Online
Summer 2019	OMS CS/Cybersecurity 6262 Network Security	129	Online
Summer 2019	OMS CS/Cybersecurity 6035 Introduction to Information Security	456	Online
Spring 2019	OMS CS/Cybersecurity 6262 Network Security	275	Online
Spring 2019	OMS CS/Cybersecurity 6035 Introduction to Information Security	938	Online
Fall 2018	CS 6262 Network Security	47	At CoC
Fall 2018	OMSCS 6262 Network Security	154	Online
Fall 2018	OMSCS 6035 Introduction to Information Security	525	Online
Summer 2018	OMSCS 6262 Network Security	191	Online
Summer 2018	OMSCS 6035 Introduction to Information Security	338	Online
Spring 2018	OMSCS 6262 Network Security	220	Online
Spring 2018	OMSCS 6035 Introduction to Information Security	428	Online
Fall 2017	CS 4235/6035 Introduction to Information Security	120	At CoC
Fall 2017	OMSCS 6262 Network Security	300	Online
Fall 2017	OMSCS 6035 Introduction to Information Security	350	Online
Summer 2017	OMSCS 6262 Network Security	300	Online
Summer 2017	OMSCS 6035 Introduction to Information Security	350	Online
Spring 2017	OMSCS 6262 Network Security	300	Online
Spring 2017	OMSCS 6035 Introduction to Information Security	350	Online
Fall 2016	CS 4235/6035 Introduction to Information Security	80	At CoC
Fall 2016	OMSCS 6262 Network Security	300	Online
Fall 2016	OMSCS 6035 Introduction to Information Security	350	Online
Summer 2016	OMSCS 6035 Introduction to Information Security	350	Online
Spring 2016	OMSCS 6035 Introduction to Information Security	350	Online
Fall 2015	CS 4235/6035 Introduction to Information Security	80	At CoC
Fall 2015	OMSCS 6035 Introduction to Information Security	230	Online
Fall 2014	CS 4235/6035 Introduction to Information Security	80	At CoC
Spring 2013	CS 4235 Introduction to Information Security	50	At CoC
Fall 2012	CS 4235 Introduction to Information Security	41	At CoC
Spring 2012	CS 6262 Network Security	35	At CoC

<u>Semester/Year</u>	<u>Course</u>	<u>Number of Students</u>	<u>Comments</u>
Fall 2010	CS 3210 Operating System Design	25	At CoC
Spring 2010	CS 6262 Network Security	45	At CoC
Fall 2009	CS 4237 Computer and Network Security	15	At CoC
Spring 2009	CS 4237 Computer and Network Security	15	At CoC
Fall 2008	CS 6262 Network Security	55	At CoC
Spring 2008	CS 8803ANS Advanced Systems and Network Security	25	At CoC
Fall 2007	CS 4237 Computer and Network Security	20	At CoC
Spring 2007	CS 6262 Network Security	45	At CoC
Fall 2006	CS 6265 Information Security Lab (w/ Mustaque)	18	At CoC
Fall 2006	CS 4803 Computer and Network Security	30	At CoC
Spring 2006	CS 6262 Network Security	45	At CoC
Fall 2005	CS 4803 Computer and Network Security	30	At CoC
Spring 2005	CS 6262 Network Security	30	At CoC
Fall 2004	CS 4803K Computer and Network Security	30	At CoC
Spring 2004	CS 6262 Network Security	30	At CoC
Fall 2003	CS 4803K Computer and Network Security	35	At CoC
Spring 2003	CS 8803K Advanced Systems and Network Security	16	At CoC
Fall 2002	CS 6262 Network Security	28	At CoC
Spring 2002	CS 6262 Network Security	30	New at CoC
Fall 2000	CSC 591Q Special Topics: Network Security	65	At NC State
Fall 1999	CSC 591Q Special Topics: Network Security	45	At NC State
Summer 1998	CS 3139 Data Structures and Algorithms	20	At Columbia

B. Curriculum Development

CS 6264/OMS CS/Cybersecurity 6264: Information Security Labs: System and Network Defenses. This graduate-level course helps students develop both in-depth knowledge and hands-on skills in a number of important cybersecurity areas, including software security, malware, and threat analysis, end-point security, network security, web security, mobile security, and machine learning-based security analytics. The lecture materials of each topic area are drawn from the latest research papers and prototypes, and a comprehensive project is designed to help students master each area. The lecture materials explain the design principles of cutting-edge security tools, and the projects are designed to let students extend these tools. Most of the tools are in the open-source, and therefore, students can continue to build on and use these tools beyond this course. Recorded videos for on-line offering as part of OMSCS as well as OMS Cybersecurity.

CS 6262/OMS CS/Cybersecurity 6262: Network Security. Graduate-level course in network security with topics including large-scale attacks and impacts, penetration testing and security assessments, security of Internet protocols (IP, TCP, DNS, and BGP), advanced web security, advanced malware analysis, advanced network monitoring, Internet-scale threat analysis, bitcoins and cryptocurrencies, big data and security, cloud security, and attack-tolerant systems. Completely revamped

in 2016 with new course materials drawing from research papers and projects developed by my Ph.D. students. Recorded videos for on-line offering via Udacity as part of OMSCS as well as OMS Cybersecurity.

CS 4235/OMS CS/Cybersecurity 6035: Introduction to Information Security. Cross-listed undergraduate and graduate introductory course in information security. It teaches the basic concepts, principles, and fundamental approaches to secure computers and networks. Its main topics include: security basics, security management and risk assessment, software security, operating systems security, database security, cryptography algorithms and protocols, network authentication and secure network applications, malware, network threats and defenses, web security, mobile security, legal and ethical issues, and privacy. Completely revamped in 2015 with new materials covering the up-to-date threat models, attack methods, and new technologies and policy considerations. Recorded videos for on-line offering via Udacity as part of OMS CS as well as OMS Cybersecurity.

CS 6262: Network Security. Graduate-level course focusing on the fundamental principles as well as advanced techniques in network security. This course covers cryptography and its application to network and operating system security; authentication; security for electronic mail; Java security; Firewalls and intrusion detection.

CS 4237: Computer and Network Security. New undergraduate course (first offered as CS 4803 in Fall 2003) focusing on the fundamental principles and techniques in computer and network security.

CS 8803: Advanced Systems and Network Security. Graduate course on advanced topics such as network and host-based intrusion detection, botnet detection, software security, malware analysis, and virtual machine monitoring. Students are required to study research papers and work on research projects.

C. Individual Student Guidance

1. Postdoctoral Fellows Supervised

Bo Feng: 2022 - present

Erkam Uzun: 2021 - 2022; Software Engineer, ByteDance/TikTok

Yisroel Mirsky: 2019 - 2021; Tenure-track Lecturer in the Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Israel

Guangliang Yang: 2019 - 2021; Assistant Professor in the Department of Computer Science, Fudan University, Shanghai, China

Hong Hu: 2017-2019; Tenure-track Assistant Professor in the College of Information Sciences and Technology, Pennsylvania State University

Sangho Lee: 2015-2018; Researcher in Microsoft Research (MSR) Redmond

Tielei Wang: 2011 - 2014; Start-up in China

Simon Pak Ho Chung: 2011 -2014; Research Scientist in SCS at Georgia Tech

Roberto Perdisci: 2009 - 2010; Professor with tenure in the Department of Computer Science, University of Georgia

Daniel Xiapu Luo: 2008 - 2010; Associate Professor with tenure at Hong Kong Polytechnic University

2. Ph.D. Students Supervised

Xinzhou Qin: Graduated in Summer 2005; Senior Engineering Manager, VmWare

Yian Huang: Graduated in Summer 2006; Senior Engineering Manager, LinkedIn

Prahlad Fogla: Graduated in Summer 2007; Software Engineer, Google

Guofei Gu: Graduated in Summer 2008; Professor with tenure in the Department of Computer Science, Texas A&M University

Bryan Payne: Graduated in Summer 2010; Chief Information Security Officer, BetterUp

Monirul Sharif: Graduated in Fall 2010; Director of Engineering, Google

Kapil Singh: Graduated in Summer 2011; Principal Research Scientist, IBM T.J. Watson Research Center

Manos Antonakakis: Graduated in Spring 2012; Associate Professor with tenure in ECE at Georgia Tech

Martim Carbone: Graduated in Summer 2012; Systems Software Engineer, VMware

Junjie Zhang: Graduated in Summer 2012; Associate Professor with tenure in the Department of Computer Science and Engineering, Wright State University

Long Lu: Graduated in Summer 2013; Chief Scientist & Global Head of Digital Safety & Security, NIO; Associate Professor with tenure in the College of Computer and Information Science, Northeastern University

Brendan Dolan-Gavitt: Graduated in Summer 2014; tenure-track Assistant Professor in the Department of Computer Science and Engineering, NYU Tandon School of Engineering

Yacin Nadji: Graduated in Summer 2015; Senior Security Researcher, Corelight

Xinyu Xing: Graduated in Summer 2015; Associate Professor of Computer Science, Northwestern University

Chengyu Song: Graduated in Summer 2016; Associate Professor (with tenure) in the Department of Computer Science and Engineering, UC Riverside

Byoungyoung Lee: Graduated in Summer 2016; Associate Professor in the Department of Electrical and Computer Engineering, Seoul National University, South Korea

Yeongjin Jang: Graduated in Summer 2017; Tenure-track Assistant Professor in the Department of Electrical Engineering and Computer Science, Oregon State University

Kangjie Lu: Graduated in Summer 2017; Tenure-track Assistant Professor in the Department of Computer Science and Engineering, University of Minnesota

Wei Meng: Graduated in Summer 2017; Tenure-track Assistant Professor in the Department of Computer Science and Engineering, The Chinese University of Hong Kong

Yizheng Chen: Graduated in Fall 2017; Tenure-track Assistant Professor in the Department of Computer Science, The University of Maryland, College Park, Maryland

Yang Ji: Graduated in Fall 2019; Manager of Research, Palo Alto Networks

Ruian Duan: Graduated in Fall 2019; Staff Security Researcher, Palo Alto Networks

Erkam Uzun: Graduated in Summer 2021; Software Engineer, ByteDance/TikTok

Chenxiong Qian: Graduated in Summer 2021; Tenure-track Assistant Professor in the Department of Computer Science, The University of Hong Kong

Carter Yagemann: Graduated in Summer 2022; Tenure-track Assistant Professor in the Department of Computer Science and Engineering, The Ohio State University

Evan Downing: In Progress

Joey Allen: In Progress

Moses Ike: In Progress

Matthew Landen: In Progress

Yongheng Chen: In Progress

Feng Xiao: In Progress

Haoran Wang: In Progress

Ding Zhang: In Progress

Zhang Yang: In Progress

Yupeng Yang: In Progress

Jizhou Chen: In Progress

II. RESEARCH AND CREATIVE SCHOLARSHIP

A. Theses

Ph.D. Thesis

Title: A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems.

Date Completed: June 1999

Advisor: Professor Salvatore J. Stolfo.

University: Columbia University

B. Published Journal Papers (refereed)

1. Yisroel Mirsky, Ambra Demontis, Jaidip Kotak, Ram Shankar, Deng Gelei, Liu Yang, Xianguyu Zhang, Maura Pintor, Wenke Lee, Yuval Elovici, and Battista Biggio. The Threat of Offensive AI to Organizations. *Computer Security*. Vol 124, 2023.
2. Yisroel Mirsky and Wenke Lee. The Creation and Detection of Deepfakes: A Survey. *ACM Computing Surveys*. 54(1), 2022.
3. Kangjie Lu, Meng Xu, Chengyu Song, Taesoo Kim, and Wenke Lee. Stopping Memory Disclosures via Diversification and Replicated Execution. *IEEE Transactions on Dependable and Secure Computing (TDSC)*. 18(1). 2021.
4. Shana Moothedath, Dinuka Sahabandu, Joey Allen, Andrew Clark, Linda Bushnell, Wenke Lee, and Radha Poovendran. A Game-Theoretic Approach for Dynamic Information Flow Tracking to Detect Multistage Advanced Persistent Threats. *IEEE Transactions on Automatic Control*. 65(12): 5248-5263, 2020.
5. Junjie Zhang, Roberto Perdisci, Wenke Lee, Unum Sarfraz, and Xiapu Luo. Building a Scalable System for Stealthy P2P-Botnet Detection. *IEEE Transactions on Information Forensics and Security*, January 2014.
6. Roberto Perdisci, Davide Ariu, Prahlad Fogla, Giorgio Giacinto, and Wenke Lee. McPAD: A Multiple Classifier System for Accurate Payload-Based Anomaly Detection. *Computer Networks*, Vol. 53, 2009.
7. Roberto Perdisci, Andrea Lanzi, and Wenke Lee. Classification of Packed Executables for Accurate Computer Virus Detection. *Pattern Recognition Letters* Vol. 29, No. 14 (October 2008).
8. Prahlad Fogla and Wenke Lee. q-Gram Matching Using Tree Models. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 4 (April 2006).
9. W. Fan, M. Miller, S. Stolfo, Wenke Lee, and P. Chan. Using Artificial Anomalies to Detect Unknown and Known Network Intrusions. *Knowledge and Information Systems*, Vol. 6, No. 5 (September 2004), Springer.

10. Yongguang Zhang, Wenke Lee, and Yian Huang. Intrusion Detection Techniques for Mobile Wireless Networks. *ACM/Kluwer Wireless Networks Journal (ACM WINET)*, 9(5), September 2003.
11. Joao B.D. Cabrera, Lundy Lewis, Xinzhou Qin, Wenke Lee, and Raman K. Mehra. Proactive Intrusion Detection and Distributed Denial of Service Attacks - A Case Study in Security Management. *Journal of Network and Systems Management*. 10(2), June 2002.
12. Wenke Lee, Wei Fan, Matt Miller, Sal Stolfo, and Erez Zadok. Toward Cost-Sensitive Modeling for Intrusion Detection and Response. *Journal of Computer Security*, 10(1,2), 2002.
13. Wenke Lee and Sal Stolfo. A Framework for Constructing Features and Models for Intrusion Detection Systems. *ACM Transactions on Information and System Security*, 3(4), November 2000.
14. Wenke Lee, Sal Stolfo, and Kui Mok. Adaptive Intrusion Detection: A Data Mining Approach. *Artificial Intelligence Review*, Kluwer Academic Publishers, 14(6):533-567, December 2000.
15. Wenke Lee and Gail E. Kaiser. Interfacing Oz with the PCTE OMS: A Case Study of Integrating a Legacy System with a Standard Object Management System. *Journal of Systems Integration*, Kluwer Academic Publishers, 9(4):329-358, December 1999.

C. Published Books and Parts of Books

1. Xinzhou Qin and Wenke Lee. Discovering Novel Attack Strategies from INFOSEC Alerts. *Data Warehousing and Data Mining Techniques for Cyber Security*. Anoop Singhal (eds), Springer, 2007.
2. Yongguang Zhang and Wenke Lee. Security in Mobile Ad-Hoc Networks. *Ad Hoc Networks: Technologies and Protocols*. P. Mohapatra and S. Krishnamurthy (eds), Springer, 2004.
3. Xinzhou Qin, Wenke Lee, Lundy Lewis, Joao B.D. Cabrera. Using MIB II Variables for Network Intrusion Detection. *Applications of Data Mining in Computer Security*. D. Barbara and S. Jajodia (eds), Kluwer Academic Publishers, May 2002.
4. Joao B.D. Cabrera, Lundy Lewis, Xinzhou Qin, Wenke Lee, Raman K. Mehra. Proactive Intrusion Detection - A Study on Temporal Data Mining. *Applications of Data Mining in Computer Security*. D. Barbara and S. Jajodia (eds), Kluwer Academic Publishers, May 2002
5. Wenke Lee, Sal Stolfo, and Kui Mok. Algorithms for Mining System Audit Data. *Data Mining, Rough Sets, and Granular Computing*. T. Y. Lin, Y. Y. Yao, and L. A. Zadeh (eds), Physica-Verlag, 2002
6. Wenke Lee and Naser Barghouti. Jadve: An Extensible Data Visualization Environment. In *Object-Oriented Applications Frameworks*, M. Fayad, D. Schmidt, and R. Johnson (eds), John Wiley & Sons, 1999.

D. Edited Proceedings and Books

1. *Botnet Detection: Countering the Largest Security Threat (Advances in Information Security)*, Wenke Lee, Cliff Wang, and David Dagon (Eds.), Springer, 2007.
2. *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection (RAID 2001)*, Wenke Lee, Ludovic Me, and Andreas Wespi (Eds.), Lecture Notes in Computer Science, Vol. 2212, Springer, 2001.

E. Conference Presentations (refereed)

E.1. Conference Presentations with Proceedings

1. Zheng Yang, Joey Allen, Matthew Landen, Roberto Perdisci, and Wenke Lee. TRIDENT: Towards Detecting and Mitigating Web-based Social Engineering Attacks. In *Proceedings of the 32nd USENIX Security Symposium (USENIX)*. Anaheim, California. 2023. (to appear)
2. Yisroel Mirsky, George Macon, Michael Brown, Carter Yagemann, Matthew Pruett, Evan Downing, Sukarno Mertoguno, and Wenke Lee. VulChecker: Graph-based Vulnerability Localization in Source Code. In *Proceedings of the 32nd USENIX Security Symposium (USENIX)*. Anaheim, California. 2023. (to appear)
3. Carter Yagemann, Simon Chung, Brendan Saltaformaggio, and Wenke Lee. PUMM: Preventing Use-After-Free Using Execution Unit Partitioning. In *Proceedings of the 32nd USENIX Security Symposium (USENIX)*. Anaheim, California. 2023. (to appear)
4. Moses Ike, Kandy Phan, Keaton Sadoski, Romuald Valme, and Wenke Lee. SCAPHY: Detecting Modern ICS Attacks by Correlating Behaviors in SCADA and PHYsical. In *Proceedings of the 2023 IEEE Symposium on Security and Privacy*. San Francisco, California. 2023. (to appear)
5. Matthew Landen, Keywhan Chung, Moses Ike, Sarah Mackay, Jean-Paul Watson, and Wenke Lee. DRAGON: Deep Reinforcement Learning for Autonomous Grid Operation and Attack Detection. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*. 2022.
6. ChangSeok Oh, Sangho Lee, Chenxiong Qian, Hyungjoon Koo, and Wenke Lee. DeView: Confining Progressive Web Applications by Debloating Web APIs. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*. 2022.
7. Feng Xiao, Zheng Yang, Joey Allen, Guangliang Yang, Grant Williams, and Wenke Lee. Understanding and Mitigating Remote Code Execution Vulnerabilities in Cross-platform Ecosystem. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2022.
8. Carter Yagemann, Simon Chung, Brendan Saltaformaggio, and Wenke Lee. Automated Bug Hunting with Data-Driven Symbolic Root Cause Analysis. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2021.

9. Carter Yagemann, Mohammad Nouredine, Wajih Hassan, Simon Chung, Adam Bates, and Wenke Lee. Validating the Integrity of Audit Logs Against Execution Repartitioning Attacks. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2021.
10. Erkam Uzun, Simon P. Chung, Vladimir Kolesnikov, Alexandra Boldyreva, and Wenke Lee. Fuzzy Labeled Private Set Intersection with Applications to Private Real-Time Biometric Search. In *Proceedings of the 2021 USENIX Security Symposium*. 2021.
11. Evan Downing, Kyuhong Park, Yisroel Mirsky, and Wenke Lee. DeepReflect: Discovering Malicious Functionality through Binary Reconstruction. In *Proceedings of the 2021 USENIX Security Symposium*. 2021.
12. Carter Yagemann, Matthew Pruett, Simon P. Chung, Kennon Bittick, Brendan Saltaformaggio, and Wenke Lee. ARCUS: Symbolic Root Cause Analysis of Exploits in Production Systems. In *Proceedings of the 2021 USENIX Security Symposium*. 2021.
13. Feng Xiao, Jianwei Huang, Yichang Xiong, Guangliang Yang, Hong Hu, Guofei Gu, and Wenke Lee. Abusing Hidden Properties to Attack the Node.js Ecosystem. In *Proceedings of the 2021 USENIX Security Symposium*. 2021.
14. Kyuhong Park, Burak Sahin, Yongheng Chen, Jisheng Zhao, Evan Downing, Hong Hu, and Wenke Lee. Identifying Behavior Dispatchers for Malware Analysis. In *Proceedings of the 16th ACM ASIA Conference on Computer and Communications Security (ACM AsiaCCS 2021)*.
15. Erkam Uzun, Carter Yagemann, Simon P. Chung, Vladimir Kolesnikov, and Wenke Lee. Cryptographic Key Derivation from Biometric Inferences for Remote Authentication. In *Proceedings of the 16th ACM ASIA Conference on Computer and Communications Security (ACM AsiaCCS 2021)*.
16. Dongsong Yu, Guangliang Yang, Guozhu Meng, Xiaorui Gong, Xiu Zhang, Xiaobo Xiang, Xiaoyu Wang, Yue Jiang, Kai Chen, Wei Zou, Wenke Lee, and Wenchang Shi. SEPAL: Towards a Large-scale Analysis of SEAndroid Policy Customization. In *Proceedings of The Web Conference 2021 (WWW 2021)*.
17. Yongheng Chen, Rui Zhong, Hong Hu, Hangfan Zhang, Yupeng Yang, Dinghao Wu, and Wenke Lee. One Engine to Fuzz 'em All: Generic Language Processor Testing with Semantic Validation. In *Proceedings of the 41st IEEE Symposium on Security and Privacy (Oakland)*. 2021.
18. Ruian Duan, Omar Alrawi, Ranjita Pai Kasturi, Ryan Elder, Brendan Saltaformaggio, and Wenke Lee. Towards Measuring Supply Chain Attacks on Package Managers for Interpreted Languages. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. 2021.
19. Kangjie Lu, Meng Xu, Chengyu Song, Taesoo Kim, and Wenke Lee. Stopping Memory Disclosures via Diversification and Replicated Execution. *IEEE Transactions on Dependable and Secure Computing (TDSC)*. 18(1), 2021.

20. Carter Yagemann, Simon P. Chung, Erkam Uzun, Sai Ragam, Brendan Saltaformaggio, and Wenke Lee. On the Feasibility of Automating Stock Market Manipulation. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*. 2020.
21. Chenxiong Qian, Hyungjoon Koo, ChangSeok Oh, Taesoo Kim, and Wenke Lee. Slimium: Debloating the Chromium Browser with Feature Subsetting. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2020.
22. Joey Allen, Zheng Yang, Matthew Landen, Raghav Bhat, Harsh Grover, Andrew Chang, Yang Ji, Roberto Perdisci, and Wenke Lee. Mnemosyne: An Effective and Efficient Postmortem Watering Hole Attack Investigation System. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2020.
23. Rui Zhong, Yongheng Chen, Hong Hu, Hangfan Zhang, Wenke Lee, and Dinghao Wu. SQUIRREL: Testing Database Management Systems with Language Validity and Coverage Feedback. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2020.
24. Dinuka Sahabandu, Joey Allen, Shana Moothedath, Linda Bushnell, Wenke Lee, and Radha Poovendran. Quickest Detection of Advanced Persistent Threats: A Semi-Markov Game Approach. In *Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*. 2020.
25. D. Sahabandu, S. Moothedath, J. Allen, A. Clark, L. Bushnell, Wenke Lee, and R. Poovendran. Dynamic Information Flow Tracking Games for Simultaneous Detection of Multiple Attackers. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, Nice, France, December 2019.
26. S. Misra, S. Moothedath, H. Hosseini, J. Allen, L. Bushnell, Wenke Lee, and R. Poovendran. Learning Equilibria in Stochastic Information Flow Tracking Games with Partial Knowledge. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, Nice, France, December 2019.
27. Dinuka Sahabandu, Shana Moothedath, Joey Allen, Linda Bushnell, Wenke Lee, and Radha Poovendran. Stochastic Dynamic Information Flow Tracking Game with Reinforcement Learning. In *Proceedings of the 2019 Conference on Decision and Game Theory for Security*, Stockholm, Sweden, October 2019.
28. Chenxiong Qian, Hong Hu, Mansour Alharthi, Pak Ho Chung, Taesoo Kim, and Wenke Lee. RAZOR: A Framework for Post-deployment Software Debloating. In *Proceedings of the 28th USENIX Security Symposium*. Santa Clara, CA, August 2019.
29. Dinuka Sahabandu, Shana Moothedath, Linda Bushnell, Radha Poovendran, Joey Aller, Wenke Lee, and Andrew Clark. A Game Theoretic Approach for Dynamic Information Flow Tracking with Conditional Branching. In *Proceedings of the 2019 American Control Conference (ACC)*. Philadelphia, PA, July 2019.

30. Ruian Duan, Ashish Bijlani, Yang Ji, Omar Alrawi, Yiyuan Xiong, Moses Ike, Brendan Saltaformaggio, and Wenke Lee. Automating Patching of Vulnerable Open-Source Software Versions in Application Binaries. In *Proceedings of the 2019 Network and Distributed System Security Symposium (NDSS)*. San Diego, CA, February 2019.
31. Joey Allen, Matthew Landen, Sanya Chaba, Yang Ji, Simon Pak Ho Chung, and Wenke Lee. Improving Accuracy of Android Malware Detection with Lightweight Contextual Awareness. In *Proceedings of the 34th Annual Computer Security Applications Conference (ACSAC)*. December, 2018.
32. Dinuka Sahabandu, Baicen Xiao, Andrew Clark, Sangho Lee, Wenke Lee, and Radha Poovendran. DIFT Games: Dynamic Information Flow Tracking Games for Advanced Persistent Threats. In *Proceedings of The 57th IEEE Conference on Decision and Control (CDC)*. Miami Beach, FL, December 2018.
33. Shana Moothedath, Dinuka Sahabandu, Andrew Clark, Sangho Lee, Wenke Lee, and Radha Poovendran. Multi-Stage Dynamic Information Flow Tracking Game. In *Proceedings of The 9th Conference on Decision and Game Theory for Security (GameSec)*. Seattle, WA, October 2018.
34. Hong Hu, Chenxiong Qian, Carter Yagemann, Simon Pak Ho Chung, Bill Harris, Taesoo Kim, and Wenke Lee. Enforcing Unique Code Target Property for Control-Flow Integrity. In *Proceedings of The 25th ACM Conference on Computer and Communications Security (CCS 2018)*. Toronto, Canada, October 2018.
35. Andrea Possemato, Andrea Lanzi, Simon Pak Ho Chung, Wenke Lee, and Yanick Fratantonio. ClickShield: Are You Hiding Something? Towards Eradicating Clickjacking on Android. In *Proceedings of The 25th ACM Conference on Computer and Communications Security (CCS 2018)*. Toronto, Canada, October 2018.
36. Yang Ji, Sangho Lee, Mattia Fazzini, Joey Allen, Evan Downing, Taesoo Kim, Alessandro Orso, and Wenke Lee. Enabling Refinable Cross-Host Attack Investigation with Efficient Data Flow Tagging and Tracking. In *Proceedings of the 27th USENIX Security Symposium*. Baltimore, MD, August 2018
37. Wei Meng, Chenxiong Qian, Shuang Hao, Kevin Borgolte, Giovanni Vigna, and Christopher Kruegel, and Wenke Lee. Rampart: Protecting Web Applications from CPU-Exhaustion Denial-of-Service Attacks. In *Proceedings of the 27th USENIX Security Symposium*. Baltimore, MD, August 2018
38. Erkam Uzun, Simon Pak Ho Chung, Irfan Essa, and Wenke Lee. rtCaptcha: A Real-Time CAPTCHA Based Liveness Detection System. In *Proceedings of The 2018 Network and Distributed System Security Symposium (NDSS)*. San Diego, CA, February 2018.
39. Antonio Bianchi, Yanick Fratantonio, Aravind Machiry, Christopher Kruegel, Giovanni Vigna, Simon Pak Ho Chung, and Wenke Lee. Broken Fingers: On the Usage of the Fingerprint API in Android. In *Proceedings of the 2018 Network and Distributed System Security Symposium (NDSS)*. San Diego, CA, February 2018.

40. Ruian Duan, Ashish Bijlani, Meng Xu, Taesoo Kim, and Wenke Lee. Identifying Open-Source License Violation and 1-day Security Risk at Large Scale. In *Proceedings of The 24th ACM Conference on Computer and Communications Security (CCS 2017)*. Dallas, Texas, October 2017.
41. Yang Ji, Sangho Lee, Evan Downing, Weiren Wang, Mattia Fazzini, Taesoo Kim, Alessandro Orso, and Wenke Lee. RAIN: Refinable Attack Investigation with On-demand Inter-Process Information Flow Tracking. In *Proceedings of The 24th ACM Conference on Computer and Communications Security (CCS 2017)*. Dallas, Texas, October 2017.
42. Ren Ding, Chenxiong Qian, Chengyu Song, Bill Harris, Taesoo Kim, and Wenke Lee. Efficient Protection of Path-Sensitive Control Security. In *Proceedings of the 26th USENIX Security Symposium*. Vancouver, BC, Canada, August 2017.
43. Meng Xu, Kangjie Lu, Taesoo Kim, and Wenke Lee. Bunshin: Compositing Security Mechanisms through Diversification. In *Proceedings of the 2017 USENIX Annual Technical Conference*. Santa Clara, CA, July 2017.
44. Yanick Fratantonio, Chenxiong Qian, Pak Chung, and Wenke Lee. Cloak and Dagger: From Two Permissions to Complete Control of the UI Feedback Loop. In *Proceedings of The 2017 IEEE Symposium on Security and Privacy*. San Jose, CA, May 2017 (**Distinguished Practical Paper Award**).
45. Kangjie Lu, Marie-Therese Walter, David Pfaff, Stefan Nuernberger, Wenke Lee, and Michael Backes. Unleashing Use-Before-Initialization Vulnerabilities in the Linux Kernel Using Targeted Stack Spraying. In *Proceedings of The 2017 Network and Distributed System Security Symposium (NDSS)*. San Diego, CA, February 2017.
46. Le Guan, Jun Xu, Shuai Wang, Xinyu Xing, Lin Lin, Heqing Huang, Peng Liu, and Wenke Lee. From Physical to Cyber: Escalating Protection for Personalized Auto Insurance. In *Proceedings of The 14th ACM Conference on Embedded Networked Sensor Systems (SenSys 2016)*. Stanford, CA, November 2016.
47. Kangjie Lu, Chengyu Song, Taesoo Kim, and Wenke Lee. UniSan: Proactive Kernel Memory Initialization to Eliminate Data Leakages. In *Proceedings of The 23rd ACM Conference on Computer and Communications Security (CCS 2016)*. Vienna, Austria, October 2016.
48. Yizheng Chen, Panagiotis Kintis, Manos Antonakakis, Yacin Nadji, David Dagon, Wenke Lee, and Michael Farrell. Financial Lower Bounds of Online Advertising Abuse - A Four Year Case Study of the TDSS/TDL4 Botnet. In *Proceedings of The 13th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2016)*. Sebastian, Spain, July 2016.
49. Chengyu Song, Hyungon Moon, Monjur Alam, Insu Yun, Byoungyoung Lee, Taesoo Kim, and Wenke Lee, Yunheung Paek. HDFI: Hardware-Assisted Data-Flow Isolation. In *Proceedings of The 2016 IEEE Symposium on Security and Privacy*. San Jose, CA, May 2016.

50. Wei Meng, Byoungyoung Lee, Xinyu Xing, and Wenke Lee. TrackMeOrNot: Enable Flexible Control on Web Tracking. In *Proceedings of The 25th International World Wide Web Conference (WWW)*, April 2016. San Diego, CA, February 2016.
51. Wei Meng, Ren Ding, Simon P. Chung, Steven Han, and Wenke Lee. The Price of Free: Privacy Leakage in Personalized Mobile In-Apps Ads. In *Proceedings of The 2016 Network and Distributed System Security Symposium (NDSS)*. San Diego, CA, February 2016.
52. Kangjie Lu, Wenke Lee, Stefan Nrnberger, and Michael Backes. How to Make ASLR Win the Clone Wars: Runtime Re-Randomization. In *Proceedings of The 2016 Network and Distributed System Security Symposium (NDSS)*. San Diego, CA, February 2016.
53. Chengyu Song, Byoungyoung Lee, Kangjie Lu, William Harris, Taesoo Kim and Wenke Lee. Enforcing Kernel Security Invariants with Data Flow Integrity. In *Proceedings of The 2016 Network and Distributed System Security Symposium (NDSS)*. San Diego, CA, February, 2016.
54. Meng Xu, Yeongjin Jang, Xinyu Xing, Taesoo Kim, and Wenke Lee. UCognito: Private Browsing without Tears. In *Proceedings of The 22nd ACM Conference on Computer and Communications Security (CCS)*. Denver, CO, October 2015.
55. Kangjie Lu, Chengyu Song, Byoungyoung Lee, Simon P. Chung, Taesoo Kim, and Wenke Lee. ASLR-Guard: Stopping Address Space Leakage for Code Reuse Attacks. In *Proceedings of The 22nd ACM Conference on Computer and Communications Security (CCS)*. Denver, CO, October 2015.
56. Byoungyoung Lee, Chengyu Song, Taesoo Kim, and Wenke Lee. Type Casting Verification: Stopping an Emerging Attack Vector. In *Proceedings of The 24th USENIX Security Symposium*. Washington, D.C., August 2015. (Awarded the **Internet Defense Prize** by Facebook and USENIX)
57. Xinyu Xing, Wei Meng, Byoungyoung Lee, Udi Weinsberg, Anmol Sheth, Roberto Perdisci, and Wenke Lee. Unraveling the Relationship Between Ad-Injecting Browser Extensions and Malvertising. In *Proceedings of The 24th International World Wide Web Conference (WWW)*. Florence, Italy, May 2015.
58. Chengyu Song, Chao Zhang, Tielei Wang, Wenke Lee, and David Melski. Exploiting and Protecting Dynamic Code Generation. In *Proceedings of The 2015 Network and Distributed System Security Symposium (NDSS)*. San Diego, CA, February, 2015.
59. Byoungyoung Lee, Chengyu Song, Yeongjin Jang, Tielei Wang, Taesoo Kim, Long Lu, and Wenke Lee. Preventing Use-after-free with Dangling Pointers Nullification. In *Proceedings of The 2015 Network and Distributed System Security Symposium (NDSS)*. San Diego, CA, February, 2015.
60. Kangjie Lu, Zhichun Li, Vasileios P. Kemerlis, Zhenyu Wu, Long Lu, Cong Zheng, Zhiyun Qian, Wenke Lee, and Guofei Jiang. Checking More and Alerting Less: Detecting Privacy Leakages via Enhanced Data-flow Analysis and Peer Voting. In *Proceedings of The 2015 Network and Distributed System Security Symposium (NDSS)*. San Diego, CA, February, 2015.

61. Wei Meng, Xinyu Xing, Anmol Sheth, Udi Weinsberg, and Wenke Lee. Your Online Interests Pwned! A Pollution Attack Against Targeted Advertising. In *Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS)*. Scottsdale, AZ, November 2014.
62. Yeongjin Jang, Chengyu Song, Simon Chung, Tielei Wang, and Wenke Lee. A11y Attacks: Exploiting Accessibility in Operating Systems. In *Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS)*. Scottsdale, AZ, November 2014.
63. Billy Lau, Pak Ho Chung, Chengyu Song, Yeongjin Jang, Wenke Lee, and Alexandra Boldyreva. Mimesis Aegis: A Mimicry Privacy Shield - A System's Approach to Data Privacy on Public Cloud. In *Proceedings of the 23rd USENIX Security Symposium*. San Diego, CA, August 2014.
64. Tielei Wang, Yeongjin Jang, Yizheng Chen, Pak Ho Chung, Billy Lau, and Wenke Lee. On the Feasibility of Large-Scale Infections of iOS Devices. In *Proceedings of the 23rd USENIX Security Symposium*. San Diego, CA, August 2014.
65. Byoungyoung Lee, Long Lu, Tielei Wang, Taesoo Kim, and Wenke Lee. From Zygote to Morula: Fortifying Weakened ASLR on Android. In *Proceedings of the 35th IEEE Symposium on Security and Privacy (Oakland)*, San Jose, CA, May 2014.
66. Xinyu Xing, Wei Meng, Dan Doozan, Nick Feamster, Wenke Lee, and Alex C. Snoeren. Exposing Inconsistent Web Search Results with Bobble. In *Proceedings of The 2014 Passive and Active Measurement Conference (PAM)*, Los Angeles, CA, March 2014.
67. Yeongjin Jang, Simon P. Chung, Bryan D. Payne, and Wenke Lee. Gyrus: A Framework for User-Intent Monitoring of Text-Based Networked Applications. In *Proceedings of the 21st Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2014.
68. Yacin Nadji, Manos Antonakakis, Roberto Perdisci, and Wenke Lee. Beheading Hydras: Performing Effective Botnet Takedowns. In *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS)*, Berlin, Germany, November 2013.
69. Brendan Dolan-Gavitt, Tim Leek, Josh Hodosh, and Wenke Lee. Tappan Zee (North) Bridge: Mining Memory Accesses for Introspection. In *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS)*, Berlin, Germany, November 2013.
70. Xinyu Xing, Wei Meng, Dan Doozan, Alex C. Snoeren, Nick Feamster, and Wenke Lee. Take this Personally: Pollution Attacks on Personalized Services. In *Proceedings of the 22nd USENIX Security Symposium*, Washington, D.C., August 2013.
71. Tielei Wang, Kangjie Lu, Long Lu, Simon Chung, and Wenke Lee. Jekyll on iOS: When Benign Apps Become Evil. In *Proceedings of the 22nd USENIX Security Symposium*, Washington, D.C., August 2013.

72. Yacin Nadji, Manos Antonakakis, Roberto Perdisci, and Wenke Lee. Connected Colors: Unveiling the Structure of Criminal Networks. In *Proceedings of the 16th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, St. Lucia, October 2013 (to appear).
73. Junjie Zhang, Yinglian Xie, Fang Yu, David Soukal, and Wenke Lee. Intention and Origination: An Inside Look at Large-Scale Bot Queries. In *Proceedings of The 20th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2013.
74. Charles Lever, Manos Antonakakis, Bradley Reaves, Patrick Traynor and Wenke Lee. The Core of the Matter: Analyzing Malicious Traffic in Cellular Carriers. In *Proceedings of The 20th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2013.
75. Long Lu, Zhichun Li, Zhenyu Wu, Wenke Lee, and Guofei Jiang. CHEX: Statically Vetting Android Apps for Component Hijacking Vulnerabilities. In *Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS)*, Raleigh, NC. October 2012.
76. Martim Carbone, Matthew Conover, Bruce Montague, and Wenke Lee. Secure and Robust Monitoring of Virtual Machines through Guest-Assisted Introspection. In *Proceedings of the 15th International Symposium on Research in Attacks, Intrusions, and Defenses (RAID)*. Amsterdam, The Netherlands. September, 2012.
77. Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee, and David Dagon. From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware. In *Proceedings of the 21st USENIX Security Symposium*. Bellevue, WA. August 2012.
78. Kapil Singh, Helen Wang, Alexander Moshchuk, Collin Jackson, and Wenke Lee. Practical End-to-End Web Content Integrity. In *Proceedings of the 21st International World Wide Web Conference (WWW)*, Lyon, France, April 2012.
79. Xiapu Luo, Peng Zhou, Junjie Zhang, Roberto Perdisci, Wenke Lee, and Rocky K.C. Chang. Exposing Invisible Timing-Based Traffic Watermarks with BACKLIT. In *Proceedings of The 27th Annual Computer Security Applications Conference (ACSAC 2011)*, Orlando, FL, December 2011.
80. Yacin Nadji, Manos Antonakakis, Roberto Perdisci, and Wenke Lee. Understanding the Prevalence and Use of Alternative Plans in Malware with Network Games. In *Proceedings of The 27th Annual Computer Security Applications Conference (ACSAC 2011)*, Orlando, FL, December 2011.
81. Long Lu, Roberto Perdisci, and Wenke Lee. SURF: Detecting and Measuring Search Poisoning. In *Proceedings of The 18th ACM Conference on Computer and Communications Security (CCS)*. Chicago, IL, October 2011.
82. Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nick Vasiloglou, and David Dagon. Detecting Malware Domains at the Upper DNS Hierarchy. In *Proceedings of The 20th USENIX Security Symposium*, San Francisco, CA August 2011.

83. Junjie Zhang, Roberto Perdisci, Wenke Lee, Unam Sarfraz, and Xiapu Luo. Detecting Stealthy P2P Botnets Using Statistical Traffic Fingerprints. In *Proceedings of The 41st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2011)*, Hong Kong, China, June 2011.
84. Xiapu Luo, Peng Zhou, Edmond W. W. Chan, Rocky K. C. Chang, and Wenke Lee. A Combinatorial Approach to Network Covert Communications with Applications in Web Leaks. In *Proceedings of The 41st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2011)*, Hong Kong, China, June 2011.
85. Brendan Dolan-Gavitt, Tim Leek, Michael Zhivich, Jon Giffin, and Wenke Lee. Virtuoso: Narrowing the Semantic Gap in Virtual Machine Introspection. In *Proceedings of The 2010 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2011.
86. Junjie Zhang, Jay Stokes, Christian Seifert, and Wenke Lee. ARROW: Generating Signatures to Detect Drive-By Downloads. In *Proceedings of The 20th International World Wide Web Conference (WWW)*, Hyderabad, India, March 2011.
87. Junjie Zhang, Xiapu Luo, Roberto Perdisci, Guofei Gu, Wenke Lee, and Nick Feamster. Boosting the Scalability of Botnet Detection Using Adaptive Traffic Sampling. In *Proceedings of The 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, Hong Kong, March 2011.
88. Xiapu Luo, Peng Zhou, Edmond W.W. Chan, Wenke Lee, Rocky K. C. Chang, and Roberto Perdisci. HTTPoS: Sealing Information Leaks with Browser-side Obfuscation of Encrypted Flows. In *Proceedings of The 18th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2011.
89. Qing Hui, Xiapu Luo, and Wenke Lee. Control of Low-Rate Denial-of-Service Attacks on Web Servers and TCP Flows. In *Proceedings of The 49th IEEE Conference on Decision and Control (CDC)*, Atlanta, GA, December 2010.
90. Long Lu, Vinod Yegneswaran, Phil Porras, and Wenke Lee. BLADE: An Attack-Agnostic Approach for Preventing Drive-By Malware Infections. In *Proceedings of The 17th ACM Conference on Computer and Communications Security (CCS)*, Chicago, IL, October 2010.
91. Xiapu Luo, Junjie Zhang, Roberto Perdisci, and Wenke Lee. On the Secrecy of Spread-Spectrum Flow Watermarks. In *Proceedings of The 15th European Symposium on Research in Computer Security (ESORICS)*, Athens, Greece, September 2010.
92. Manos Antonakakis, David Dagon, Xiapu Luo, Roberto Perdisci, and Wenke Lee. A Centralized Monitoring Infrastructure for Improving DNS Security. In *Proceedings of The 13th International Symposium on Recent Advances in Intrusion Detection (RAID)*, Ottawa, Ontario, Canada, September 2010.
93. Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. Building a Dynamic Reputation System for DNS. In *Proceedings of The 19th USENIX Security Symposium*, Washington, DC, August 2010.

94. Kapil Singh, Samrit Sangal, Nehil Jain, Patrick Traynor, and Wenke Lee. Evaluating Bluetooth as a Medium for Botnet Command and Control. In *Proceedings of The 7th Conference on Detection of Intrusions and Malware Vulnerability Assessment (DIMVA)*, Bonn, Germany, July 2010.
95. Kapil Singh, Alexander Moshchuk, Helen J. Wang, and Wenke Lee. On the Incoherencies in Web Browser Access Control Policies. In *Proceedings of The 2010 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2010.
96. Roberto Perdisci, Wenke Lee, and Nick Feamster. Behavioral Clustering of HTTP-based Malware and Signature Generation using Malicious Network Traces. In *Proceedings of The 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, San Jose, CA, April 2010.
97. Roberto Perdisci, Igino Corona, David Dagon, and Wenke Lee. Detecting Malicious Flux Service Networks through Passive Analysis of Recursive DNS Traces. In *Proceedings of The 25th Annual Computer Security Applications Conference (ACSAC 2009)*, Honolulu, HI, December 2009.
98. Guofei Gu, Vinod Yegneswaran, Phillip Porras, Jennifer Stoll, and Wenke Lee. Active Botnet Probing to Identify Obscure Command and Control Channels. In *Proceedings of The 25th Annual Computer Security Applications Conference (ACSAC 2009)*, Honolulu, HI, December 2009.
99. Monirul Sharif, Wenke Lee, Weidong Cui, and Andrea Lanzi. Secure In-VM Monitoring Using Hardware Virtualization. In *Proceedings of The 16th ACM Conference on Computer and Communications Security (CCS 2009)*, Chicago, IL, November, 2009.
100. Martim Carbone, Weidong Cui, Long Lu, Wenke Lee, Marcus Peinado, and Xuxian Jiang. Mapping Kernel Objects to Enable Systematic Integrity Checking. In *Proceedings of The 16th ACM Conference on Computer and Communications Security (CCS 2009)*, Chicago, IL, November, 2009.
101. Kapil Singh, Sumeer Bhola, and Wenke Lee. xBook: Redesigning Privacy Control in Social Networking Platforms. In *Proceedings of The 18th USENIX Security Symposium*, Montreal, Canada, August, 2009.
102. Roberto Perdisci, Manos Antonakakis, Xiapu Luo, and Wenke Lee. WSEC DNS: Protecting Recursive DNS Resolvers from Poisoning Attacks. In *Proceedings of The 39th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2009)*, Lisbon, Portugal, June 2009.
103. Monirul Sharif, Andrea Lanzi, Jon Giffin, and Wenke Lee. Automatic Reverse Engineering of Malware Emulators. In *Proceedings of The 2009 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2009. **(Best Student Paper Award)**
104. Andrea Lanzi, Monirul Sharif, and Wenke Lee. K-Tracer: A System for Extracting Kernel Malware Behavior. In *Proceedings of The 16th Annual Network and Distributed System Security Symposium (NDSS 2009)*, San Diego, CA, February 2009.

105. David Dagon, Manos Antonakakis, Kevin Day, Xiapu Luo, Christopher P. Lee, and Wenke Lee. Recursive DNS Architectures and Vulnerability Implications. In *Proceedings of The 16th Annual Network and Distributed System Security Symposium (NDSS 2009)*, San Diego, CA, February 2009.
106. Roberto Perdisci, Andrea Lanzi, and Wenke Lee. McBoost: Boosting Scalability in Malware Collection and Analysis Using Statistical Classification of Executables. In *Proceedings of The 24th Annual Computer Security Applications Conference (ACSAC 2008)*, Anaheim, CA, December 2008.
107. Artem Dinaburg, Paul Royal, Monirul Sharif, and Wenke Lee. Ether: Malware Analysis via Hardware Virtualization Extensions. In *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS 2008)*, Alexandria, VA, October 2008.
108. David Dagon, Manos Antonakakis, Paul Vixie, Tatuya Jinmei, and Wenke Lee. Increased DNS Forgery Resistance Through 0x20-Bit Encoding. In *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS 2008)*, Alexandria, VA, October 2008.
109. Monirul Sharif, Vinod Yegneswaran, Hassen Saidi, Phillip Porras, and Wenke Lee. Eureka: A Framework for Enabling Static Malware Analysis. In *Proceedings of the 13th European Symposium on Research in Computer Security (ESORICS)*, Malaga, Spain, October 2008.
110. Guofei Gu, Roberto Perdisci, Junjie Zhang, and Wenke Lee. BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection. In *Proceedings of The 17th USENIX Security Symposium (Security'08)*, San Jose, CA, July 2008.
111. Kapil Singh, Abhinav Srivastava, Jon Giffin, and Wenke Lee. Evaluating Email's Feasibility for Botnet Command and Control. In *Proceedings of the 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2008)*, Anchorage, Alaska, June 2008.
112. Bryan D. Payne, Martim Carbone, Monirul Sharif, and Wenke Lee. Lares: An Architecture for Secure Active Monitoring Using Virtualization. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2008.
113. Guofei Gu, Alvaro A. Cardenas, and Wenke Lee. Principled Reasoning and Practical Applications of Alert Fusion in Intrusion Detection Systems. In *Proceedings of the ACM Symposium on InformAction, Computer and Communications Security (ASIACCS'08)*, Tokyo, Japan, March 2008.
114. David Dagon, Niels Provos, Chris Lee, and Wenke Lee. Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority. In *Proceedings of The 15th Annual Network and Distributed System Security Symposium (NDSS 2008)*, San Diego, CA, February 2008.
115. Guofei Gu, Junjie Zhang, and Wenke Lee. BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic. In *Proceedings of The 15th Annual Network and Distributed System Security Symposium (NDSS 2008)*, San Diego, CA, February 2008.

116. Monirul Sharif, Andrea Lanzi, Jonathon Giffin, and Wenke Lee. Impeding Malware Analysis using Conditional Code Obfuscation. In *Proceedings of The 15th Annual Network and Distributed System Security Symposium (NDSS 2008)*, San Diego, CA, February 2008.
117. Bryan D. Payne, Martim Carbone, and Wenke Lee. Secure and Flexible Monitoring of Virtual Machines. In *Proceedings of The 23rd Annual Computer Security Applications Conference (ACSAC 2007)*, Miami Beach, FL, December 2007.
118. David Dagon, Guofei Gu, Chris Lee and Wenke Lee. A Taxonomy of Botnet Structures. In *Proceedings of The 23rd Annual Computer Security Applications Conference (ACSAC 2007)*, Miami Beach, FL, December 2007.
119. Guofei Gu, Zesheng Chen, Phillip Porras and Wenke Lee. Misleading and Defeating Importance-Scanning Malware Propagation. In *Proceedings of The 3rd International Conference on Security and Privacy in Communication Networks (SecureComm'07)*, Nice, France, September 2007.
120. Takehiro Takahashi and Wenke Lee. An Assessment of VoIP Covert Channel Threats. In *Proceedings of The 3rd International Conference on Security and Privacy in Communication Networks (SecureComm'07)*, Nice, France, September 2007.
121. Monirul Sharif, Kapil Singh, Jonathon Giffin and Wenke Lee. Understanding Precision in Host Based Intrusion Detection: Formal Analysis and Practical Models. In *Proceedings of The 10th International Symposium on Recent Advances in Intrusion Detection (RAID)*, Surfers Paradise, Australia, September 2007.
122. Guofei Gu, Phillip Porras, Vinod Yegneswaran, Martin Fong, Wenke Lee. BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation. In *Proceedings of The 16th USENIX Security Symposium (Security'07)*, Boston, MA, August 2007.
123. David Cash, Yan Zong Ding, Yevgeniy Dodis, Wenke Lee, Richard Lipton, and Shabsi Wal-fish. Intrusion-Resilient Key Exchange in the Bounded Retrieval Model. In *Proceedings of The Fourth IACR Theory of Cryptography Conference (TCC 2007)*, Amsterdam, The Netherlands, February 2007.
124. Roberto Perdisci, Guofei Gu, and Wenke Lee. Using an Ensemble of One-Class SVM Classifiers to Harden Payload-based Anomaly Detection Systems. In *Proceedings of The 2006 IEEE International Conference on Data Mining (ICDM '06)*, Hong Kong, China, December 2006.
125. Paul Royal, Mitch Halpin, David Dagon, Robert Edmonds, and Wenke Lee. PolyUnpack: Automating the Hidden-Code Extraction of Unpack-Executing Malware. In *Proceedings of The 22th Annual Computer Security Applications Conference (ACSAC 2006)*, Miami Beach, FL, December 2006.
126. Prahlad Fogla and Wenke Lee. Evading Network Anomaly Detection Systems: Formal Reasoning and Practical Techniques. In *Proceedings of The 13th ACM Conference on Computer and Communications Security (CCS 2006)*, Alexandria, VA, October 2006.

127. Guofei Gu, Prahlad Fogla, David Dagon, Wenke Lee, and Boris Skoric. Towards an Information-Theoretic Framework for Analyzing Intrusion Detection Systems. In *Proceedings of The 11th European Symposium Research Computer Security (ESORICS 2006)*, Hamburg, Germany, September 2006.
128. Prahlad Fogla, Monirul Sharif, Roberto Perdisci, Oleg Kolesnikov, and Wenke Lee. Polymorphic Blending Attacks. In *Proceedings of The 15th USENIX Security Symposium (SECURITY '06)*, Vancouver, B.C., Canada, August 2006.
129. Collin Mulliner, Giovanni Vigna, David Dagon, and Wenke Lee. Using Labeling to Prevent Cross-Service Attacks Against Smart Phones. In *Proceedings of The 3rd Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2006)*, Berlin, Germany, July 2006.
130. Hongmei Deng, Roger Xu, Jason H. Li, Frank Zhang, Renato Levy, and Wenke Lee. Agent-Based Cooperative Anomaly Detection for Wireless Ad Hoc Networks. In *Proceedings of The 12th International Conference on Parallel and Distributed Systems (ICPADS 2006)*, Minneapolis, Minnesota, July 2006.
131. Guofei Gu, Prahlad Fogla, Wenke Lee, and Douglas Blough. DSO: Dependable Signing Overlay. In *Proceedings of The 4th International Conference on Applied Cryptography and Network Security (ACNS '06)*, Singapore, June 2006.
132. Roberto Perdisci, David Dagon, Wenke Lee, Prahlad Fogla, and Monirul Sharif. Misleading Worm Signature Generators Using Deliberate Noise Injection (full paper). In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2006.
133. Guofei Gu, Prahlad Fogla, David Dagon, Wenke Lee, and Boris Skoric. Measuring Intrusion Detection Capability: An Information-Theoretic Approach. In *Proceedings of ACM Symposium on InformAction, Computer and Communications Security (ASIACCS '06)*, Taipei, Taiwan, March 2006.
134. David Dagon, Cliff Zou, and Wenke Lee. Modeling Botnet Propagation Using Time Zones. In *Proceedings of The 13th Annual Network and Distributed System Security Symposium (NDSS 2006)*, San Diego, CA, February 2006.
135. Yongguang Zhang, Yi-an Huang, and Wenke Lee. An Extensible Environment for Evaluating Secure MANET. In *Proceedings of The 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm 2005)*, Athens, Greece, September 2005.
136. Jonanthon T. Giffin, David Dagon, Somesh Jha, Wenke Lee, and Barton P. Miller. Environment-Sensitive Intrusion Detection. In *Proceedings of The 8th International Symposium on Recent Advances in Intrusion Detection (RAID 2005)*, Seattle, WA, September 2005.
137. David Dagon, Wenke Lee, and Richard Lipton. Protecting Secret Data from Insider Attacks. *Proceedings of The Ninth International Conference on Financial Cryptography and Data Security (FC'05)*, Roseau, Dominica, February, 2005.

138. Guofei Gu, David Dagon, Xinzhou Qin, Monirul I. Sharif, Wenke Lee, and George F. Riley. Worm Detection, Early Warning, and Response Based on Local Victim Information. *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC 2004)*, Tucson, Arizona, December 2004.
139. Xinzhou Qin and Wenke Lee. Attack Plan Recognition and Prediction Using Causal Networks. *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC 2004)*, Tucson, Arizona, December 2004.
140. Joao B.D. Cabrera, Jaykumar Gosar, Wenke Lee, and Raman K. Mehra. On the Statistical Distribution of Processing Times in Network Intrusion Detection. *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC 2004)*, Bahamas, December 2004.
141. George F. Riley, Monirul I. Sharif, and Wenke Lee. Simulating Internet Worms. *Proceedings of the 12th Annual Meeting of the IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Volendam, The Netherlands, October 2004.
142. Yian Huang and Wenke Lee. Attack Analysis and Detection for Ad Hoc Routing Protocols. *Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID 2004)*, Sophia Antipolis, France, September 2004.
143. David Dagon, Xinzhou Qin, Guofei Gu, Wenke Lee, Julian Grizzard, John Levin, and Henry Owen. HoneyStat: Local Worm Detection Using Honeypots. *Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID 2004)*, Sophia Antipolis, France, September 2004.
144. Xinzhou Qin and Wenke Lee. Discovering Novel Attack Strategies from INFOSEC Alerts. *Proceedings of the 9th European Symposium on Research in Computer Security (ESORICS 2004)*, Sophia Antipolis, France, September 2004.
145. H. Feng, Jonathon T. Giffin, Yong Huang, Somesh Jha, Wenke Lee, and Barton P. Miller. Formalizing Sensitivity in Static Analysis for Intrusion Detection. *Proceedings of the 2004 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2004.
146. Xinzhou Qin and Wenke Lee. Statistical Causality Analysis of INFOSEC Alert Data. In *Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID 2003)*, Pittsburgh, PA, September 2003.
147. H. Feng, Oleg Kolesnikov, Prahlad Fogla, Wenke Lee, and Weibo Gong. Anomaly Detection Using Call Stack Information. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2003.
148. Yi-an Huang, Wei Fan, Wenke Lee, and Philip S. Yu. Cross-Feature Analysis for Detecting Ad-Hoc Routing Anomalies. In *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS)*, Providence, RI, May 2003.

149. Joao B.D. Cabrera, Wenke Lee, R.K. Prasanth, Lundy Lewis, and Raman K. Mehra. Optimization and Control Problems in Real Time Intrusion Detection. *Proceedings of the 41st IEEE Conference on Decision and Control (CDC 2002)*, Las Vegas, December, 2002.
150. Wenke Lee, Joao B.D. Cabrera, Ashley Thomas, Niranjana Balwalli, Sunmeet Saluja, and Yi Zhang. Performance Adaptation in Real-Time Intrusion Detection Systems. In *Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection (RAID 2002)*, Zurich, Switzerland, October 2002.
151. Xinzhou Qin, Wenke Lee, Lundy Lewis, and Joao B.D. Cabrera. Integrating Intrusion Detection and Network Management. *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2002)*, Florence, Italy, May 2002.
152. Wei Fan, Matt Miller, Sal Stolfo, Wenke Lee, and Phil Chan. Using Artificial Anomalies to Detect Unknown and Known Network Intrusions. *Proceedings of The First IEEE International Conference on Data Mining (ICDM)*, San Jose, CA, November 2001.
153. Wenke Lee, Sal Stolfo, Phil Chan, Eleazar Eskin, Wei Fan, Matt Miller, Shlomo Hershtik, and Junxin Zhang. Real Time Data Mining-based Intrusion Detection. *Proceedings of the 2001 DARPA Information Survivability Conference and Exposition (DISCEX II)*, Anaheim, CA, June 2001.
154. Wenke Lee and Dong Xiang. Information-Theoretic Measures for Anomaly Detection. *Proceedings of The 2001 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2001.
155. Joao B.D. Cabrera, L. Lewis, X. Qin, Wenke Lee, Ravi Prasanth, B. Ravichandran, and Raman Mehra. Proactive Detection of Distributed Denial of Service Attacks Using MIB Traffic Variables - A Feasibility Study. *Proceedings of The Seventh IFIP/IEEE International Symposium on Integrated Network Management (IM 2001)*, Seattle, WA, May 2001.
156. Yongguang Zhang and Wenke Lee. Intrusion Detection in Wireless Ad-Hoc Networks. *Proceedings of The Sixth International Conference on Mobile Computing and Networking (MobiCom 2000)*, Boston, MA, August 2000.
157. Wei Fan, Wenke Lee, Sal Stolfo, and Matt Miller. A Multiple Model Cost-Sensitive Approach for Intrusion Detection. *Proceedings of The Eleventh European Conference on Machine Learning (ECML 2000)*, LNAI 1810, Barcelona, Spain, May 2000.
158. Sal Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, and Phil Chan. Cost-based Modeling for Fraud and Intrusion Detection: Results from the JAM Project. *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition (DISCEX '00)*, Hilton Head, SC, January 2000.
159. Wenke Lee, Sal Stolfo, and Kui Mok. Mining in a Data-flow Environment: Experience in Network Intrusion Detection. *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '99)*, San Diego, CA, August 1999.

160. Wenke Lee, Sal Stolfo, and Kui Mok. A Data Mining Framework for Building Intrusion Detection Models. *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, Oakland, CA, May 1999.
161. Wenke Lee, Sal Stolfo, and Kui Mok. Mining Audit Data to Build Intrusion Detection Models. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD '98)*, New York, NY, August 1998.
162. Wenke Lee and Sal Stolfo. Data Mining Approaches for Intrusion Detection. *Proceedings of the Seventh USENIX Security Symposium (SECURITY '98)*, San Antonio, TX, January 1998.
163. Sal Stolfo, Andreas Prodromidis, Shelley Tselepis, Wenke Lee, Wei Fan, and Phil Chan. JAM: Java Agents for Meta-learning over Distributed Databases. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD '97)*, Newport Beach, CA, August 1997.
164. Naser S. Barghouti, John Mocenigo, and Wenke Lee. Grappa: A GRAPh PAckage in Java. *Proceedings of the Fifth Annual Symposium on Graph Drawing (Graph Drawing '97)*, Rome, Italy, September 1997.
165. Wenke Lee, Gail Kaiser, Paul Clayton, and Eric Sherman. OzCare: A Workflow Automation System for Care Plans. *Proceedings of the American Medical Informatics Association Annual Fall Symposium*, Washington DC, October 1996.

F. Workshop Presentations (refereed)

F.1. Workshop Presentations with Proceedings

1. Yi-an Huang and Wenke Lee. Hotspot-Based Traceback for Mobile Ad Hoc Networks. In *Proceedings of The ACM Workshop on Wireless Security (WiSe 2005)*, Cologne, Germany, September 2005.
2. Monirul Sharif, George Riley, and Wenke Lee. Comparative Study between Analytical Models and Packet-Level Worm Simulations. In *Proceedings of The 19th Workshop on Parallel and Distributed Simulation (PADS 2005)*, Monterey, CA, June 2005.
3. Chris Clark, Wenke Lee, David Schimmel, Didier Contis, Mohamed Kone, and Ashley Thomas. A Hardware Platform for Network Intrusion Detection and Prevention. *The 3rd Workshop on Network Processors and Applications (NP3)*, Madrid, Spain, February 2004.
4. Yian Huang and Wenke Lee. A Cooperative Intrusion Detection System for Ad Hoc Networks. *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, Fairfax VA, October 2003.
5. Mustaque Ahamad, Wenke Lee, Ling Liu, Leo Mark, Edward Omicienski, Calton Pu, and Andre dos Santos. Guarding the Next Internet Frontier: Countering Denial of Information Attacks. *Proceedings of the 2002 New Security Paradigms Workshop*, Virginia Beach, Virginia, September 2002.

6. Xinzhou Qin, Wenke Lee, Lundy Lewis, and Joao B.D. Cabrera. Using MIB III Variables for Network Anomaly Detection - A Feasibility Study. *ACM Workshop on Data Mining for Security Applications*, Philadelphia, PA, November 2001.
7. Yongguang Zhang, Harrick Vin, Lorenzo Alvisi, Wenke Lee, and Son K. Dao. Heterogeneous Networking: A New Survivability Paradigm. *Proceedings of the 2001 New Security Paradigms Workshop*, Cloudcroft, New Mexico, September 2001.
8. Wenke Lee, Wei Fan, Matt Miller, Sal Stolfo, and Erez Zadok. Toward Cost-Sensitive Modeling for Intrusion Detection and Response. *ACM Workshop on Intrusion Detection Systems*, Athens, Greece, November 2000.
9. Wenke Lee, Rahul Nimbalkar, Kam Yee, Sunil Patil, Pragnesh Desai, Thuan Tran, and Sal Stolfo. A Data Mining and CIDF Based Approach for Detecting Novel and Distributed Intrusions. *Proceedings of The Third International Workshop on Recent Advances in Intrusion Detection (RAID 2000)*, LNCS 1907, Toulouse, France, October 2000.
10. Wenke Lee, Chris Park, and Sal Stolfo. Towards Automatic Intrusion Detection using NFR. *1st USENIX Workshop on Intrusion Detection and Network Monitoring*, April 1999.
11. Wenke Lee, Sal Stolfo, and Phil Chan. Learning Patterns from Unix Process Execution Traces for Intrusion Detection. *AAAI Workshop: AI Approaches to Fraud Detection and Risk Management*, July 1997.
12. Sal Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, and Phil Chan. Credit Card Fraud Detection Using Meta-Learning: Issues and Initial Results. *AAAI Workshop: AI Approaches to Fraud Detection and Risk Management*, July 1997.
13. Gail Kaiser and Wenke Lee. Pay No Attention to the Man Behind the Curtain. *NSF Workshop on Workflow and Process Automation*, May 1996.
14. Wenke Lee. Data Modeling and Management for Large Spatial Databases. *The Third International Workshop in Geographic Information Systems*, Beijing, China, August 1993.

G. Other

1. Matt Blaze, Sampath Kannan, Insup Lee, Oleg Sokolsky, Jonathan Smith, Angelos Keromytis, and Wenke Lee. Dynamic Trust Management. *IEEE Computer*, February 2009.
2. Martim Carbone, Diego Zamboni, and Wenke Lee. Taming Virtualization. *IEEE Security & Privacy*, vol. 6, no. 1, January/February 2008.
3. Bryan D. Payne, Reiner Sailer, Ramon Caceres, Ronald Perez, and Wenke Lee. A Layered Approach to Simplified Access Control in Virtualized Systems. *ACM SIGOPS Operating Systems Review*, 4(2), July 2007.
4. Wenke Lee. 2002. Applying Data Mining to Intrusion Detection: The Quest for Automation, Efficiency, and Credibility. *SIGKDD Explorations*, 4(2), December 2002.

5. Wenke Lee and Wei Fan. 2001. Mining System Audit Data: Opportunities and Challenges. *SIGMOD Record*, 30(4), December 2001.
6. Salvatore J. Stolfo, Wenke Lee, Philip K. Chan, Wei Fan, and Eleazar Eskin. 2001. Data Mining-Based Intrusion Detectors: An Overview of the Columbia IDS Project. *SIGMOD Record*, 30(4), December 2001.

H. Research Proposals and Grants

a. Approved and Funded

1. AI Support for Cybersecurity

David Wagner (PI; UC Berkeley); co-PIs at UC Berkeley: Michael Jordan, Vern Paxson, Jacob Steinhardt, and Nina Haghtalab; co-PI at Georgia Tech: Wenke Lee
UC Berkeley via c3.ai
Funded at \$140,000 to Georgia Tech, 7/1/2022-6/30/2023.

2. Proof-driven Refinement of Infrastructure Software Modules (PRISM)

Brendan Saltaformaggio (PI); co-PIs: Wenke Lee, Taesoo Kim, Alessandro Orso, Qirun Zhang, Yisroel Mirsky, Martin Osterloh, Jason Li, Michael Brown, Sukarno Mertoguno, Kevin Hamlen, and Nicholas Evancich
DARPA I2O
Funded at \$10,647,759, 7/1/2021-12/1/2025.

3. D3: Debloating, Dialecting and Diversification for Attack Resilient Software with Real-time Constraints

Taesoo Kim (PI); co-PI: Wenke Lee
Technology Innovation Institute Sole Proprietorship LLC
Funded at \$1,575,000, 9/1/2020-8/31/2023.

4. HECTOR: Halting Emergent Computation Through Oracle Reinforcement

Wenke Lee (PI); co-PIs: Sukarno Mertoguno (GTRI) and Clayton Kerce (GTRI)
DARPA I2O
Funded at \$1,000,000, 1/15/2020-7/15/2021.

5. Interactive Editing Techniques for Subsetting and Dialecting Network Protocols

Taesoo Kim (PI); co-PIs: Wenke Lee, Alex Orso, and Brendan Saltaformaggio
ONR
Funded at \$2,976,781, 8/1/2018-7/31/2021

6. SaTC: CORE: Medium: Understanding and Fortifying Machine Learning Based Security Analytics

Polo Chau (PI); co-PIs: Wenke Lee, Le Song, and Taesoo Kim
NSF
Funded at \$1,200,000, 8/1/2017-7/31/2021

7. Comprehensive System Debloating via Path-Based Learning and Late-Stage OS Composition

Wenke Lee (for Bill Harris) (PI); co-PIs: Bill Harris, Taesoo Kim, Alessandro Orso, and Santosh Pande
ONR
Funded at \$7,500,000, 8/1/2017-7/31/2022

8. **CyberCorps Scholarship-for-Service (SFS)**
Mustaque Ahamad (PI); co-PIs: Wenke Lee, Taesoo Kim, Sy Goodman, and Manos Antonakakis
NSF
Funded at \$5,000,000, 9/1/2016-10/31/2021.
9. **Multi-Layer Adaptive and Proactive Strategic Cyber Defense**
Radha Poovendran (PI, University of Washington); co-PIs: Wenke Lee (Georgia Tech) and Tamer Basar (UIUC)
ARO
Funded at \$1,000,000, 8/1/2016-7/31/2019.
10. **Intel Science & Technology Center for Adversary-Resilient Security Analytics (ISTC-ARSA)**
Wenke Lee (PI); co-PIs: Taesoo Kim, Polo Chau (CSE), and Le Song (CSE)
Funded as a \$1,500,000 gift.
11. **THEIA: Tagging and Tracking of Multi-Level Host Events for Transparent Computing and Information Assurance**
Wenke Lee (PI); co-PIs: Taesoo Kim, Alex Orso, and Simon Chung
DARPA
Funded at \$4,193,126, 6/26/2015-6/26/2019.
12. **ADAPT: Analytical Framework for Actionable Defense against Advanced Persistent Threats**
Radha Poovendran (PI, University of Washington); co-PIs: Wenke Lee (Georgia Tech), Shankar Sastry and Anthony Joseph (UC Berkeley), Joao Hespanha (UCSB), Tamer Basar (UIUC), and Maryam Fazel and Linda Bushnell (University of Washington)
ONR MURI
Funded at \$7,500,000, 6/1/2016-5/31/2021.
13. **Embedasploit: A “Pen-Test in a Box” for Industrial Control Systems**
Wenke Lee (PI); co-PIs: Sal Stolfo at Columbia, Brendan Dolan-Gavitt at NYU Tandon School of Engineering
Office of Naval Research
Funded at \$1,250,351, 5/1/2015-4/30/2018.
14. **BFT++: Attack Tolerance in Hard Real-Time Systems**
Taesoo Kim (PI); co-PI: Wenke Lee
Office of Naval Research
Funded at \$1,245,720, 4/1/2015-3/31/2018.
15. **TWC SBE: TTP Option: Medium: Collaborative: EPICA: Empowering People to Overcome Information Controls and Attacks**
Wenke Lee (PI); co-PIs: Nick Feamster, Hongyuan Zha, and Hans Klein
NSF
Funded at \$1,100,000, 8/1/14-7/31/17.

16. **Comprehensive Understanding of Malicious Overlay Networks**
 Homeland Security Advanced Project Research Agency (HSARPA of DHS)
 Wenke Lee (PI); co-PIs: David Dagon at Georgia Tech, Chris Smoak of GTRI, Roberto Perdisci (UGA), April Lorenzen of Dissect Cyber, Paul Vixie of Internet Systems Consortium, Jody Westby of Global Cyber Risk, and Matt Jonkman of Open Information Security Foundation.
 Funded at \$1,873,078 for project total, with \$673,078 to CoC/GT, 10/1/12–9/30/15.
17. **EAGER: The Conceptual Landscape of Information Manipulation**
 Wenke Lee (PI); co-PIs: Nick Feamster, and Hans Klein (Public Policy)
 NSF
 Funded at \$200,000, 8/1/12–7/31/14.
18. **Integrating Security Concepts in an Undergraduate Computer Sciences Curriculum**
 Intel
 Wenke Lee (PI); co-PIs: Mustaque Ahamad, Nick Feamster, and Paul Royal
 Funded as unrestricted gift at \$50,000, 11/2011;
 additional, unrestricted gift at \$35,000, 3/2012;
 additional, unrestricted gift at \$50,000, 11/2012;
 a total of \$135,000.
19. **Monitoring Free and Open Access to Information on the Internet**
 Google
 Nick Feamster and Wenke Lee (co-PIs)
 Funded as unrestricted gift: \$1,000,000 base for 12/15/2010–12/14/2012;
 \$500,000 option for 12/15/2012–12/14/2013;
 additional gift at \$500,000, for 12/15/13–12/14/2014;
 a total of \$2,000,000.
20. **Dimensions of Network Intelligence**
 HSARPA of DHS
 Wenke Lee (PI) and Paul Royal (co-PI)
 Funded at \$1,022,000, 9/2012–9/2017.
21. **Dynamic Reputation for DNS**
 Google
 Wenke Lee (PI)
 Funded as unrestricted gift at \$75,000, 7/2010.
22. **PEASOUP: Preventing Exploits Against Software Of Uncertain Provenance**
 IARPA/STONESOUP Program (GrammaTech subcontract)
 Wenke Lee (co-PI), with David Melski (PI) at GrammaTech, Jack Davidson and John Knight (co-PIs) at University of Virginia, and Tom Bracewell (co-PI) at Raytheon
 Funded at \$13,000,000 for project total, with \$1,695,887 to Georgia Tech, 6/1/2010–5/31/2014.
23. **Research in Virtual Machine Monitoring Techniques**
 Sandia National Labs/Sandia Corp.

Wenke Lee (PI)
Funded at \$50,000, 9/15/2010–9/15/2011.

24. **TC: SMALL: A Foundational and Practical Platform for Host Security Applications**
NSF CNS/Trustworthy Computing
Wenke Lee (PI)
Funded at \$429,571, 7/15/2010–7/15/2013.
25. **Botnet Attribution and Removal: From Axioms to Theory to Practice**
ONR MURI
Wenke Lee (PI); Co-PIs: Nick Feamster, and Jon Giffin at Georgia Tech; Farnam Jahanian, Mike Bailey, and Kang Shin at University of Michigan, Ann Arbor; Giovanni Vigna and Chris Kruegel at University of California, Santa Barbara; John Mitchell at Stanford University
Funded: \$7,500,000 project total, \$2,600,000 for Georgia Tech, 5/1/2009–4/30/2014.
26. **Collaborative Research: CT-L: CLEANSE: Cross-Layer Large-Scale Efficient Analys**
of Network Activities to Secure the Internet
NSF CNS/Cyber Trust
Wenke Lee (PI); Co-PIs: Nick Feamster, Jon Giffin, Mustaque Ahamad, and Xiaoming Huo (ISyE) at Georgia Tech; Mike Reiter and Fabian Monrose at University of North Carolina at Chapel Hill; Farnam Jahanian and Mike Bailey at University of Michigan; Phil Porras and Vinod Yegneswaran at SRI International; and Paul Vixie at Internet Software Consortium.
Funded: \$1,829,297 project total, \$758,297 for Georgia Tech, 9/1/2008–8/31/2012.
27. **SMITE: Scalable Monitoring in the Extreme**
DARPA
Wenke Lee and Nick Feamster (co-PIs at Georgia Tech); BBN subcontract
Funded: \$574,999 for Georgia Tech, 4/1/2008–3/31/2010
28. **Countering Botnets: Anomaly-Based Detection, Comprehensive Analysis, and Efficient Mitigation**
DHS HSARPA
Wenke Lee (PI); with Nick Feamster and Jon Giffin at Georgia Tech, David Dagon at Damballa, Rick Wesson and Adam Waters at Support Intelligence, Paul Vixie at Internet Systems Consortium, and Jody Westby at Global Cyber Risk
Funded: \$1,050,730 project total, \$300,000 for Georgia Tech, 4/1/2008–3/31/2010.
Add-on: \$305,066 project total, \$190,066 to Georgia Tech, 7/1/2009–9/30/2010.
29. **CRI-IAD: Collaborative Research: Enabling Security and Network Management Research for Future Networks**
NSF CRI
Nick Feamster (co-PI), Alex Gray, and Wenke Lee at Georgia Tech, and Z. Morley Mao (co-PI) and Farnam Jahanian at University of Michigan
Funded: \$397,426 for Georgia Tech (3/1/2008–2/28/2011)
30. **Collaborative Research: CT-T: Logic and Data Flow Extraction for Live and Informed Malware Execution**

NSF CNS/CyberTrust

Wenke Lee (co-PI); co-PIs: Paul Barford at University of Wisconsin at Madison, and Phil Porras at SRI International

Funded: \$220,000 for Georgia Tech (9/1/2007–8/31/2009)

31. **Foundational and Systems Support for Quantitative Trust Management**

Office of Naval Research MURI (University of Pennsylvania subcontract)

Sampath Kannan (PI); co-PIs: Insup Lee, Oleg Sokolsky, Matt Blazes, and Jonathan Smith at University of Pennsylvania, Wenke Lee at Georgia Tech, and Angelos Keromytis at Columbia University.

Funded: \$1,125,000 for Georgia Tech (project total: \$4,000,000) (5/1/2007–4/30/2012)

32. **Collaborative Research: CT-ISG: Modeling and Measuring Botnets**

NSF CNS/CyberTrust

Wenke Lee (PI) and Cliff C. Zou (co-PI) at University of Central Florida

Funded: \$350,000 for project total, Georgia Tech at \$175,000, (8/15/2006–8/15/2009)

33. **CT-ISG: Trusted Passage: Managing Distributed Trust to Meet the Needs of Emerging Applications**

NSF CNS/CyberTrust

Mustaque Ahamad (PI), Wenke Lee (co-PI), and Karsten Schwan (co-PI)

Funded: \$350,000, (8/15/2006–8/15/2009)

34. **Cyber Threat Analytics**

ARO via subcontract from SRI International

Funded: \$80,000, (7/15/2006–7/15/2007)

35. **ARL CTA**

ARL via subcontract from SPARTA/Telcordia

Funded: \$75,000, (7/15/2006–7/15/2007)

36. **Computer Network Defenses Technologies**

AFRL via subcontract from Scientific Systems Co. Funded: \$30,000, (9/13/2006–3/13/2007)

37. **ARO Workshop on Research in Botnets**

ARO

Funded: \$25,000, (4/1/2006–6/1/2007)

38. **Next-Generation Botnet Detection and Response**

DARPA (via ARO)

Funded: \$291,346, (12/15/2005–6/14/07)

39. **An Information-Theoretic Framework for Evaluating and Optimizing Intrusion Detection Performance**

Army Research Office

Funded: \$199,745, (4/1/2005–1/30/07)

40. **Preventing SQL Code Injection by Combining Static and Runtime Analysis, funded by Department of Homeland Security**
Department of Homeland Security
Alex Orso (PI) and Wenke Lee (co-PI)
Funded: \$200,000, (7/1/2005-12/31/2006)
41. **Anomaly and Misuse Detection in Network Traffic Streams – Checking and Machine Learning Approaches**
Office of Naval Research MURI (University of Pennsylvania subcontract)
Sampath Kannan (PI), Insup Lee, Oleg Sokolsky, and Linda Zhao at University of Pennsylvania, Wenke Lee at Georgia Tech, and Diana Spears and William Spears at University of Wyoming.
Funded: \$400,000 for Georgia Tech (\$2,000,000 project total) (7/15/2004–6/1/2006)
42. **Intrusion Detection and Security Management Technologies for Mobile Ad Hoc Networks**
Army Research Lab (Scientific Systems Company Inc. subcontract)
Funded: \$65,000 (2/1/2005-2/1/2006)
43. **A Course Module on Wireless Security**
Microsoft, Curriculum Development Grant
Funded: \$30,000 (unrestricted gift)
44. **Intrusion Detection Techniques for Mobile Ad Hoc Networks**
NSF, CISE/CCR/Trusted Computing
Funded: \$275,000 (8/15/2003-7/31/2006)
REU Funded: \$15,000 (9/10/2004-7/31/2006)
45. **High-Speed Intrusion Detection Sensors**
Army Research Lab (Scientific Systems Company Inc. subcontract)
Funded: \$65,000 (8/7/2003-8/6/2005)
46. **Distributed Intrusion Detection Feasibility Study**
U.S. Government (Columbia University subcontract)
Funded: \$45,000 (6/1/2003-6/30/2004)
47. **Agile Security for Storing Sensitive and Critical Information**
NSF, CISE/Trusted Computing
Mustaque Ahamad (PI), and co-PIs: Wenke Lee, Doug Blough, and H. Venkateswaran
Funded: \$460,000 (8/15/2002-7/31/2005)
48. **CAREER: Adaptive Intrusion Detection Systems**
NSF, CISE/CAREER
Funded: \$350,000 (8/15/2002–7/31/2007)
49. **ITR/SI: Guarding the Next Internet Frontier: Countering Denial of Information**
NSF, ITR
Mustaque Ahamad (PI), and co-PIs: Wenke Lee, Andre dos Santos, Ling Liu, Leo Mark, Ed

Omiecinski, and Calton Pu

Funded: \$1,694,769 (9/1/2001–8/31/2006)

50. Vulnerability Assessment Tools for Complex Information Networks

Army Research Office MURI (Harvard University subcontract)

Yu-Chi Ho (PI) at Harvard University, co-PIs: Avrom Pfeffer at Harvard University, Christos G. Cassandras at Boston University, Wei-Bo Gong at University of Massachusetts, Amherst, and Wenke Lee at Georgia Tech

Funded: \$400,000 for Georgia Tech (\$2,000,000 project total) (5/1/2001–4/30/2006)

51. A Data Mining Approach for Building Cost-sensitive and Light Intrusion Detection Models.

DARPA, ATO

Wenke Lee (PI), and co-PIs: Salvatore J. Stolfo at Columbia University and Philip Chan at Florida Institute of Technology

Funded: \$854,000 for Georgia Tech (\$2,000,001 project total) (6/1/2000–8/31/2003)

I. Honors and Awards

1. ACSAC Test of Time Award, The 35th Annual Computer Security Applications Conference (ACSAC), for the paper “Paul Royal, Mitch Halpin, David Dagon, Robert Edmonds, and Wenke Lee. PolyUnpack: Automating the Hidden-Code Extraction of Unpack-Executing Malware” published in ACSAC 2006 that have had enduring significance and impact to the security community, 2022.
2. Distinguished Service Award, IEEE Technical Committee on Security and Privacy, 2022.
3. IEEE Fellow, 2021.
4. Keynote at the ACM Conference on Computer and Communications Security (CCS), 2020.
5. ACM SIGSAC (Special Interest Group on Security, Audit and Control) Outstanding Innovation Award for “contributions to network and systems security, in particular, machine-learning based approaches to security analytics, including tackling intrusion and botnet detection”, 2019.
6. CCS Test of Time Award, The 25th ACM Conference on Computer and Communications Security (CCS), for the paper “Artem Dinaburg, Paul Royal, Monirul Sharif, and Wenke Lee. Ether: Malware Analysis via Hardware Virtualization Extensions” published in CCS 2008 that have had the greatest impact on security research and practice over the past decade, 2018.
7. ACM Fellow, 2017.
8. Distinguished Practical Paper Award, for the paper “Yanick Fratantonio, Chenxiong Qian, Pak Chung, and Wenke Lee. Cloak and Dagger: From Two Permissions to Complete Control of the UI Feedback Loop” in Proceedings of The 2017 IEEE Symposium on Security and Privacy, 2017.
9. Internet Defense Prize, awarded by Facebook and USENIX, for the paper “Byoungyoung Lee, Chengyu Song, Taesoo Kim, and Wenke Lee. Type Casting Verification: Stopping an Emerging Attack Vector” in Proceedings of The 24th USENIX Security Symposium, 2015.
10. Cyber Is A Global Sport Award (awarded to David Dagon and Wenke Lee), Department of Homeland Security, Science and Technology Directorate, Cyber Security Division, 2015.
11. Outstanding Achievement in Research Program Development Award, (awarded to Mustaque Ahamad, Wenke Lee, Paul Royal, and David Dagon), Georgia Institute of Technology, 2015.
12. Outstanding Community Service Award, the IEEE Technical Committee on Security and Privacy, 2013
13. Outstanding Faculty Leadership for the Development of GRAs Award, Georgia Institute of Technology, 2012.
14. Sigma Xi Faculty Best Paper Award, Georgia Institute of Technology, 2010.

15. Best Student Paper Award, for the paper “Monirul Sharif, Andrea Lanzi, Jon Giffin, and Wenke Lee. Automatic Reverse Engineering of Malware Emulators” in Proceedings of The 2009 IEEE Symposium on Security and Privacy, 2009.
16. Outstanding Senior Faculty Research Award, College of Computing, Georgia Institute of Technology, 2009.
17. NSF CAREER Award, 2002.
18. Best Paper Award, Applied Research Category, the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '99), August 1999 (with Kui Mok and Sal Stolfo).
19. Honorable mention (runner-up) for Best Paper Award, Applied Research Category, the 4th International Conference on Knowledge Discovery and Data Mining (KDD '98), August 1998.
20. Honorable mention (runner-up) for Best Paper Award, Applied Research Category, the 3rd International Conference on Knowledge Discovery and Data Mining (KDD '97), August 1997.

III. SERVICE

A. Professional Activities

A.1 Membership and Activities in Professional Societies

1. Fellow, Association for Computing Machinery (ACM)
2. Fellow, Institute of Electrical and Electronics Engineers (IEEE)

A.2 Journal Editorial Board Activities

1. IEEE Transactions on Dependable and Secure Computing (TDSC) 2010 - 2011.
2. ACM Transactions on Information and System Security (TISSEC) 2005 - 2011.

A.3 Conference Committee Activities

1. The 32nd USENIX Security Symposium (2023).
2. The 31st USENIX Security Symposium (2022).
3. The 2022 ACM Conference on Computer and Communications Security (2022).
4. The 2020 ACM Conference on Computer and Communications Security.
5. The 2020 IEEE Symposium on Security & Privacy.
6. The 30th USENIX Security Symposium (2021).
7. The 2019 ACM Conference on Computer and Communications Security.
8. The 2019 IEEE Symposium on Security & Privacy.
9. The 2018 ACM Conference on Computer and Communications Security.
10. The 2018 IEEE Symposium on Security & Privacy.
11. The 2018 IEEE European Symposium on Security & Privacy.
12. The 2017 ACM Conference on Computer and Communications Security.
13. The 26th USENIX Security Symposium (Security '17).
14. Co-chair, Security and Privacy Track, WWW 2017.
15. Annual Network and Distributed System Security Symposium (NDSS) 2017.
16. The 25th USENIX Security Symposium (Security '16).
17. The 23rd ACM Conference on Computer and Communications Security (CCS 2016).

18. The 2016 IEEE European Symposium on Security & Privacy.
19. Annual Network and Distributed System Security Symposium (NDSS) 2016.
20. The 22nd ACM Conference on Computer and Communications Security (CCS 2015).
21. Co-chair, Security and Privacy Track, WWW 2015.
22. Annual Network and Distributed System Security Symposium (NDSS) 2015.
23. European Symposium on Research in Computer Security (ESORICS) 2014.
24. The 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2014).
25. Security and Privacy Track, WWW 2014.
26. *PC Chair*, The 2013 IEEE Symposium on Security and Privacy.
27. The 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2013).
28. The 20th ACM Conference on Computer and Communications Security (CCS 2013).
29. The 16th International Symposium on Recent Advances in Intrusion Detection (RAID 2013).
30. The 19th ACM Conference on Computer and Communications Security (CCS 2012).
31. The 9th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DMIVA 2012).
32. The 7th USENIX Workshop on Hot Topics in Security (HotSec '12).
33. *PC co-Chair*, The 2012 IEEE Symposium on Security and Privacy.
34. Financial Cryptography and Data Security (FC 2012).
35. The 19th Annual Network and Distributed System Security Symposium (NDSS 2012).
36. The 8th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2011).
37. The 41st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2011).
38. The 20th USENIX Security Symposium (Security '11).
39. The 6th USENIX Workshop on Hot Topics in Security (HotSec '11).
40. The 2011 IEEE Symposium on Security and Privacy.
41. The 13th International Symposium on Recent Advances in Intrusion Detection (RAID 2010).

42. The 15th European Symposium on Research in Computer Security (ESORICS 2010).
43. The 7th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2010).
44. Security and Privacy Track, ICDCS 2010.
45. The 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2010).
46. Security and Privacy Track, WWW 2010.
47. *PC Chair*, The 17th Annual Network and Distributed System Security Symposium (NDSS 2010).
48. Financial Cryptography and Data Security (FC 2010).
49. The 25th Annual Computer Security Applications Conference (ACSAC 2009).
50. The 16th ACM Conference on Computer and Communications Security (CCS 2009).
51. 14th European Symposium on Research in Computer Security (ESORICS 2009).
52. The 18th USENIX Security Symposium (Security '09).
53. *PC Chair*, The Second USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), 2009.
54. *PC Co-Chair*, The Second ACM Conference on Wireless Network Security (WiSec), 2009.
55. The 2009 IEEE Symposium on Security and Privacy.
56. The 16th Annual Network and Distributed System Security Symposium (NDSS 2009).
57. The 24th Annual Computer Security Applications Conference (ACSAC 2008).
58. Internet Measurement Conference (IMC) 2008.
59. 13th European Symposium on Research in Computer Security (ESORICS 2008).
60. The 15th ACM Conference on Computer and Communications Security (CCS 2008).
61. The 17th USENIX Security Symposium (Security '08).
62. 3rd USENIX Workshop on Hot Topics in Security (HotSec '08).
63. The 2008 IEEE Symposium on Security and Privacy.
64. The 15th Annual Network and Distributed System Security Symposium (NDSS 2008).
65. The 23rd Annual Computer Security Applications Conference (ACSAC 2007).
66. The 14th ACM Conference on Computer and Communications Security (CCS 2007).

67. The 10th International Symposium on Recent Advances in Intrusion Detection (RAID 2007).
68. The 16th USENIX Security Symposium (Security '07).
69. The 2007 IEEE Symposium on Security and Privacy.
70. The 14th Annual Network and Distributed System Security Symposium (NDSS 2007).
71. The 2nd International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm 2006).
72. The 15th USENIX Security Symposium (Security '06).
73. The 13th Annual Network and Distributed System Security Symposium (NDSS 2006).
74. The 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm 2005).
75. The 2005 IEEE Symposium on Security and Privacy.
76. The 2005 ACM Workshop on Wireless Security (WiSe 2005).
77. The 2005 IEEE International Conference on Data Mining (ICDM 2005).
78. The Fourteenth International World Wide Web Conference (WWW 2005).
79. The 2005 International Conference on Distributed Computing Systems (ICDCS).
80. The 2004 ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC 2004).
81. The 2004 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2004).
82. The 2004 ACM Workshop on Wireless Security (WiSe 2004).
83. Steering Committee and the 7th International Symposium on Recent Advances in Intrusion Detection (RAID 2004).
84. The 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004.
85. The 2004 SIAM International Conference on Data Mining, 2004.
86. The 2004 IEEE International Conference on Data Mining (ICDM 2004).
87. The 2004 IEEE Information Assurance Workshop.
88. Co-Chair of the the ICDM Workshop on Data Mining for Computer Security (DMSEC 2003).
89. The 2003 IEEE International Conference on Data Mining (ICDM 2003).
90. The 2003 ACM Workshop on Wireless Security (WiSe 2003).

91. Steering Committee and the 6th International Symposium on Recent Advances in Intrusion Detection (RAID 2003).
92. The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003.
93. The 2003 IEEE Information Assurance Workshop.
94. Steering Committee for the 5th International Symposium on Recent Advances in Intrusion Detection (RAID 2002).
95. The 2002 ACM New Security Paradigms Workshop.
96. The 2002 IEEE Symposium on Security and Privacy.
97. *PC Co-Chair*, The 4th International Symposium on Recent Advances in Intrusion Detection (RAID 2001).
98. Program Committee and Organizational Committee for the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000.

B. On-campus Georgia Tech Committees

1. Chair, Faculty Recruiting Committee, School of Computer Science, 2023.
2. Chair, Faculty Recruiting Committee, School of Computer Science, 2021.
3. 5-Year Review Committee for the Chair of School of Computer Science, College of Computing, 2016.
4. Faculty Recruiting Committee, School of Computer Science, 2014.
5. Faculty Recruiting Committee, School of Computer Science, 2012-13.
6. RPT Committee, College of Computing, 2010-2012.
7. College of Computing Dean Search Committee, 2009-2010, College of Computing.
8. Chair, Faculty Recruiting Committee, 2008-2009, School of Computer Science, College of Computing.
9. Chair Advisory Committee, 2008-2009, School of Computer Science, College of Computing.
10. Co-Chair, Faculty Recruiting Committee, 2007-2008, School of Computer Science, College of Computing.
11. RPT Executive Committee, 2007-2008, College of Computing.
12. Faculty Recruiting Committee, 2006-2007, College of Computing.
13. Faculty Recruiting Committee, 2005-2006, College of Computing.

14. Undergraduate Curriculum Task Force, 2004-2005, College of Computing.
15. Ph.D. Admissions Committee, 2003-2004, College of Computing.
16. Faculty Recruiting Committee, 2001-2003, College of Computing.

E. Consulting and Advisory Appointments

1. External Review Committee of the Computer Science Department at the University of California, Santa Barbara. 2022.
2. Advisory Board of the NUS-Singtel Cyber Security Research & Development Laboratory, National University of Singapore. 2019-2021.
3. Secure, Accessible & Fair Elections (SAFE) Commission, State of Georgia, 2018-2019
4. Advisory Board of the Faculty of Engineering, The Chinese University of Hong Kong. 2017-
5. Board of Trustees, Pace Academy. 2017-

IV. NATIONAL AND INTERNATIONAL PROFESSIONAL RECOGNITION

A. Patents

1. “Privacy preserving face-based authentication”. Pak Ho Chung, Wenke Lee, Erkam Uzun, and Carter Yagemann. U.S. Patent Number: 11,494,476, November 2022.
2. “Devices, systems, and methods of program identification, isolation, and profile attachment”. Ruian Duan, Ashish Bijlani, Taesoo Kim, and Wenke Lee. U.S. Patent Number: 11,036,479, June 2021.
3. “Systems and methods for using video for user and message authentication”. Pak Ho Chung, Wenke Lee, and Yeongjin Jang. U.S. Patent Number: 10,476,888, November 2019.
4. “Method and System for Detecting Malware”. Manos Antonakakis, Roberto Perdisci, Wenke Lee, and Gunter Ollmann. U.S. Patent Number: 10,257,212, April 2019.
5. “Methods and systems for detecting compromised computers”, David Dagon, Nick Feamster, Wenke Lee, Robert Edmonds, Richard Lipton, and Anirudh Ramachandran. U.S. Patent Number: 10,044,748, August 2018.
6. “Method and system for detecting malicious and/or botnet-related domain names”. Roberto Perdisci, Wenke Lee, and Gunter Ollmann. U.S. Patent Number: 10,027,688, July 2018.
7. “Method and system for network-based detecting of malware from behavioral clustering”. Roberto Perdisci and Wenke Lee. U.S. Patent Number: 9,948,671, April 2018.
8. “Method and system for detecting DGA-based malware”. Manos Antonakakis, Roberto Perdisci, Wenke Lee, and Nikolaos Vasiloglou. U.S. Patent Number: 9,922,190, March 2018.
9. “Method and system for detecting malicious domain names at an upper DNS hierarchy”. Manos Antonakakis, Roberto Perdisci, Wenke Lee, and Nikolaos Vasiloglou. U.S. Patent Number: 9,686,291, June 2017.
10. “Systems and methods of safeguarding user information while interacting with online service providers”. Wenke Lee, Sasha Boldyreva, Simon Chung, Billy Lau, and Chengyu Song. U.S. Patent Number: 9,659,189, May 2017.
11. “Method and System for Detecting Malware”. Manos Antonakakis, Roberto Perdisci, Wenke Lee, and Gunter Ollmann. U.S. Patent Number: 9,525,699, December 2016.
12. “Method and system for determining whether domain names are legitimate or malicious”, Manos Antonakakis, Roberto Perdisci, David Dagon, and Wenke Lee. U.S. Patent Number: 9,516,058, December 2016.
13. “Method and systems for detecting compromised networks and/or computers”, David Dagon, Nick Feamster, Wenke Lee, Robert Edmonds, Richard Lipton, and Anirudh Ramachandran. U.S. Patent Number: 9,306,969, April 2016.

14. "Systems and methods for secure in-VM monitoring", Monirul Sharif and Wenke Lee. U.S. Patent Number: 9,129,106, September 2015.
15. "Method and system for network-based detecting of malware from behavioral clustering". Roberto Perdisci, Wenke Lee, and Gunter Ollmann. U.S. Patent Number: 8,826,438, September 2014.
16. "Method and system for detecting malicious domain names at an upper DNS hierarchy". Manos Antonakakis, Roberto Perdisci, Wenke Lee, and Nikolaos Vasiloglou. U.S. Patent Number: 8,631,489, January 2014.
17. "Method and System for Detecting Malware". Manos Antonakakis, Roberto Perdisci, Wenke Lee, and Gunter Ollmann. U.S. Patent Number: 8,578,497, November 2013.
18. "Method and System for Detecting and Responding to Attacking Networks". David Dagon, Nick Feamster, Wenke Lee, Robert Edmonds, Richard Lipton, and Anirudh Ramachandran. U.S. Patent Number: 8,566,928, October 2013.
19. "Methods for Cost-Sensitive Modeling for Intrusion Detection and Response". Wei Fan, Wenke Lee, Matt Miller, and Sal Stolfo. U.S. Patent Number: 7,818,797, October 2010.
20. "Method and System for Using Intelligent Agents for Financial Transactions, Services, Accounting, and Advice". Dan Schutzer, Will Foster, Huanrui Hu, Wenke Lee, Sal Stolfo, and Wei Fan. U.S. Patent Number: 5,920,848, July 1999.

V. OTHER CONTRIBUTIONS

A. Seminar Presentations

1. “Machine Learning and Security: The Good, The Bad, and The Ugly”, Distinguished Lecture at Dartmouth College, April, 2021.
2. “Machine Learning and Security: The Good, The Bad, and The Ugly”, Distinguished Lecture at Ohio State University, January, 2021.
3. “Machine Learning and Security: The Good, The Bad, and The Ugly”, Keynote at the 2020 ACM Conference on Computer and Communications Security, November, 2021.
4. “Security Overlay”, Stony Brook University, March 2016.
5. “Security Overlay”, Princeton University, October 2015.
6. “Internet Monitoring via DNS Traffic Analysis”, Google-UMD Cybersecurity Seminar, November 2012.
7. “Botnet Detection and Response: Challenges and Opportunities”, Carnegie Mellon University, September 2007.
8. “Botnet Detection and Response”, Indiana University, March 2007.
9. “Botnet Detection and Response”, University of Pittsburgh, October 2006.
10. “Architecture Considerations for Anomaly Detection”, University of Texas at San Antonio, November, 2005
11. “Advanced Intrusion Detection Techniques”, National Security Research Institute, Daejeon, South Korea, February, 2005
12. “U.S. Policy and Recent Detection Techniques for Cyber Threats and Attacks”, National Cyber Security Center, Seoul South Korea, February, 2005
13. “Architecture Considerations for Anomaly Detection”, Purdue University, February, 2005
14. “Architecture Considerations for Anomaly Detection”, Graduate Center of The City University of New York, February, 2005
15. “Architecture Considerations for Anomaly Detection”, The University of Alabama, November, 2004
16. “Architecture Considerations for Anomaly Detection”, Carnegie Mellon University, September, 2004
17. “Architecture Considerations for Anomaly Detection”, University of Wisconsin at Madison, August, 2004

18. "Local Worm Detection and Response: Algorithms and Analytical Models", Lawrence Livermore National Laboratory, June 2004
19. "Intrusion Detection", Carnegie Mellon University, October 2003
20. "Algorithms for Recognizing New Attack Step Relationships", University of Pennsylvania, September 2003
21. "Intrusion Detection", Emory University, October 2001
22. "Developing Data Mining Techniques for Intrusion Detection: A Progress Report", Center for Education and Research in Information Assurance and Security, Purdue University, October 2000
23. "A Data Mining Framework for Building Intrusion Detection Models", Institute for Security Technology Studies, Dartmouth College, Hanover, NH, July 2000
24. "A Data Mining Framework for Building Intrusion Detection Models", Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, February 2000
25. "A Data Mining Framework for Building Intrusion Detection Models", HRL Laboratories, Malibu, CA, November 1999