UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETSKOPE, INC.,

Petitioner,

v.

FORTINET, INC.,

Patent Owner.

PTAB Case No. IPR 2023-00457

Patent No. 9,231,968 B2

DECLARATION OF WENKE LEE IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 9,231,968 B2

TABLE OF CONTENTS

I. INTRODUCTION 1

A. Qualifications 2

1. Education 2

2. Work Experience 2

3. Publications 4

4. Curriculum Vitae 5

B. Materials Reviewed 5

C. Level of Ordinary Skill in the Art 7

D. Summary of Opinions 8

II. OVERVIEW OF THE TECHNOLOGY 9

A. Priority Date of the Claims 9

B. Overview of Relevant Technology 10

1. Malware Detection 10

2. Updating Malware Detection Signatures 13

3. Distributed Malware Signature Updates 14

C. The '968 Patent 17

1. Alleged Invention 17

2. Priority Date 20

D. The Challenged Claims 27

E. Claim Construction 27

III. UNPATENTABILITY OF THE '968 Patent'S CLAIMS 27

A. Standards for Invalidity 27

1. Obviousness 27

2. Written Description & Enablement 28

B. Ground I: Combining Jordan and FortiGate renders claims 1-9 obvious. 29

1. Motivation to Combine 37

2. Claims 1, 4, and 7 37

Netskope Exhibit 1002

3. Claims 2, 5, and 8: "…wherein the analysis process, to which the network traffic content is forwarded, executes on a computing device distinct from the network gateway device[/ data network device]." 76
4. Claims 3, 6, and 9: "…wherein the analysis process facilitates an analysis that includes receiving administrator input." 77
5. Jordan Is Prior Art....79
C. Ground 2: Combining Lee and Magdych renders claims 1-9 obvious. 84
1. Motivation to combine....86
2. Claims 1, 4, and 7....88
3. Claims 2, 5, and 8: 108
4. Claims 3, 6, and 9: 109
D. Ground 3: Combining Lee, Magdych, and Judge renders claims 1-9 obvious. 111
E. Ground 4: Combining Fang and FortiGate renders claims 1-9 obvious. 113
1. Motivation to combine....113
2. Claims 1, 4, and 7: 114
3. Claims 2, 5, and 8: 129
4. Claims 3, 6, and 9: 129

Netskope Exhibit 1002

LIST OF APPENDICES

Appendix A	*Curriculum Vitae* of Wenke Lee, Ph.D.

Netskope Exhibit 1002

I. INTRODUCTION

1. I, Wenke Lee, have been retained by Petitioner Netskope, Inc. ("Petitioner") to investigate and opine on certain issues relating to United States Patent No. 9,231,968 ("the '968 Patent") in their Petition for *Inter Partes* Review. The Petition requests that the Patent Trial and Appeal Board ("PTAB" or "Board") review and cancel claims 1-9 of the '968 Patent.

2. The opinions set forth in this declaration are based on my personal knowledge, my professional judgment, and my analysis of the materials and information referenced in this declaration and its exhibits.

3. I am being compensated for consulting services including time spent testifying at any hearing that may be held at my standard consulting rate of $750/hr. I am also reimbursed for reasonable and customary expenses associated with my work in this case. I receive no other forms of compensation related to this case. My compensation does not depend on the outcome of this *inter partes* review or the co-pending district court litigation, and I have no other financial interest in this *inter partes* review.

4. I understand that the '968 Patent has been assigned to Fortinet, Inc.

5. This declaration is based on the information currently available to me. To the extent that additional information becomes available, I reserve the right to continue my investigation and study, which may include a review of documents and

Netskope Exhibit 1002

information that may be produced, as well as testimony from depositions that have not yet been taken.

A. Qualifications

1. Education

6. In 1988, I received a Bachelor of Science in Computer Science from Sun Yat-Sen University. In 1990, I received my Master of Science in Computer Science from the City College of New York. In 1999, I received my Ph.D. in Computer Science from Columbia University.

2. Work Experience

7. Currently, I am a Professor of Computer Science at the Georgia Institute of Technology. I focus on systems and network security, machine learning, and applied cryptography. I have also been the John P. Imlay Jr. Chair in the College of Computing since 2016.

8. I was a software engineer at Intergraph Corporation from 1990 (after obtaining my Masters) to 1994.

9. Between 1994 and 1999, I was a Ph.D. candidate in Computer Science at Columbia. I focused my research on cybersecurity and machine learning. I also was an intern during the summer of 1996, where I was a researcher at AT&T Research. In 1997, I also was a summer intern at IBM T. J. Watson Research Center.

10. From 1999 to 2001, I was a tenure-track Assistant Professor in the Department of Computer Science at North Carolina State University. Between 2001

Netskope Exhibit 1002

to 2005, I was an Assistant Professor in the College of Computing at The Georgia Institute of Technology, and I was promoted to Associate Professor with tenure in 2005. I was later promoted to Full Professor with tenure in 2009.

11. I was the director of the Information Security Center at Georgia Tech from 2012 to 2015. Then, I was the Executive Director of the Institute for Information Security and Privacy at Georgia Tech from 2015 to 2021.

12. I co-founded Damballa, Inc. in 2006. Damballa commercialized botnet and malware detection technologies that were developed in my lab. In 2016, Damballa was sold to Core Security.

13. I have received the distinction of being named a "Fellow" of the Institute of Electrical and Electronics Engineers (IEEE), the global engineering society. There are three grades of membership, and Fellow is the highest of those three grades that are awarded based on merit. Each year, IEEE only allows one tenth of one percent of the total membership to become Fellows.

14. I have also received the distinction of being named a Fellow of the Association of Computing Machinery (ACM), the global professional society for computer scientists. It is even more difficult to become a Fellow of ACM than it is for IEEE. ACM's Fellows are limited to 1% of the overall membership.

Netskope Exhibit 1002

15. From 2010-2011, I served on the Editorial Board of one of IEEE's flagship journals—the IEEE Transactions on Dependable and Secure Computing (TDSC).

16. From 2005-2011, I served on the Editorial Board of one of ACM's journals—the ACM Transactions on Information and System Security (TISSEC).

17. I have served on more than 90 conference committees. I have also served on more than a dozen committees at Georgia Tech.

18. I have served as the primary advisor to eight Postdoctoral Fellows, twenty-five completed Ph.Ds. I am currently serving as the primary advisor for more than ten Ph.D. candidates.

19. My current research area focuses on malware analysis and detection. My CV describes my current interests and areas of research on the bottom of page 2 and top of page 3.

20. I have taught courses directed to Network Information Security essentially continuously since 1999.

3. Publications

21. I am a named inventor on ten issued patents. Most of those patents relate to the field of invention of the patent focused on in this declaration.

22. I have also published more than one hundred academic papers in top journals and conferences. I have also participated in over one hundred conference

Netskope Exhibit 1002

presentations. I have also participated in over twelve workshop presentations. Most of these relate to network security and more specifically, intrusion detection.

4. Curriculum Vitae

23. A copy of my curriculum vitae is attached as Appendix A to this declaration.

B. Materials Reviewed

24. My opinions expressed in this declaration are based on documents and materials identified in this declaration, including the '968 Patent and its prosecution history, the prior art references and background materials discussed in this declaration, and the other references specifically identified in this declaration. I have considered these materials in their entirety, even if only portions are discussed here. The following is an exemplary list of materials on which I based my opinions.

Exhibit	Description
Ex. 1001	U.S. Patent No. 9,231,968 B2 to Fang et al. ("the '968 Patent")
Ex. 1003	Prosecution History of U.S. Patent No. 9,231,968 ("the '968 Prosecution History")
Ex. 1004	Prosecution History of U.S. Patent App. No. 13/795,407 ("the '407 Prosecution History")
Ex. 1005	U.S. Patent No. 8,839,417 B1 to Jordan ("Jordan")

Netskope Exhibit 1002

Ex. 1006	U.S. Provisional App. No. 60/552,457 to Fang et al. ("the '457 provisional")
Ex. 1007	U.S. Provisional App. No. 60/520,577 to Jordan ("the '577 provisional")
Ex. 1008	FortiGate – 3600 User Manual ("FortiGate")
Ex. 1009	U.S. Patent App. Pub. No. 2004/0015719 A1 to Lee et al. ("Lee")
Ex. 1010	U.S. Patent No. 6,513,122 B1 to Magdych et al. ("Magdych")
Ex. 1011	U.S. Patent App. Pub. No. 2003/0172292 A1 to Judge ("Judge")
Ex. 1012	U.S. Patent App. Pub. No. 2012/0278896 A1 to Fang et al. ("Fang")
Ex. 1013	Heady et al., The Architecture of a Network Level Intrusion Detection System, UNIVERSITY OF NEW MEXICO (1990)
Ex. 1014	Paxson, *Bro a System for Detecting Network Intruders in Real-Time*, LAWRENCE BERKELEY NATIONAL LABORATORY (1998)
Ex. 1015	U.S. Pat. No. 5,440,723 to Arnold
Ex. 1016	Lee*, A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems*, COLUMBIA UNIVERSITY
Ex. 1017	U.S. Pat. App. Pub. 2003/0204632A1 to Willebeek-LeMair et al.
Ex. 1018	Kephart et al., *Fighting Computer Viruses*, SCIENTIFIC AMERICAN (1997)

Netskope Exhibit 1002

Ex. 1019	U.S. Pat. No. 7,080,408 to Pak et al.
Ex. 1020	Prosecution History of U.S. Patent No. 9,716,644 ("Wei's Prosecution History")
Ex. 1021	A copy of the First Half of FortiGate-3600 User Manual (2003), submitted during prosecution of the '968 Patent

25. I have also relied on my own experience and expertise in intrusion detection technologies, malware detection, computer network security, and distributed network intrusion detection architecture.

C. Level of Ordinary Skill in the Art

26. I am not an attorney and offer no legal opinions. I have been informed about certain aspects of the law for purposes of my analyses and opinions.

27. I understand that in analyzing questions of invalidity and infringement, the perspective of a person having ordinary skill in the art ("POSITA") is often implicated, and the Court may need assistance in determining that level of skill.

28. I understand that the claims and written description of a patent must be understood from the perspective of a POSITA. I have been informed that the following factors may affect the level of skill of a POSITA: (1) the educational level of the inventor; (2) the type of problems encountered in the art; (3) the prior-art solutions to those problems; (4) the rapidity with which innovations are made; (5) the sophistication of the technology; and (6) the educational level of active workers

Netskope Exhibit 1002

in the field. A person of ordinary skill in the art is also a person of ordinary creativity in the art.

29. Based on my experience and my reading of the '968 Patent, a POSITA at the time of the alleged priority date of the '968 Patent in November 2013 would have had a B.S. in computer science, computer engineering, or electrical engineering (or equivalent experience) and would have had at least two years of practical experience with network and data security. This definition is flexible and additional educational experience in computer science could make up for less work experience and vice versa. The level of ordinary skills in the art would be the same if the alleged priority date of the '968 Patent was around March or November 2004.

30. I am at least a person of ordinary skill in the art and was so on the date to which the '968 Patent claims priority. As shown by my qualifications and my curriculum vitae attached as Appendix A, I am aware of the knowledge and skill possessed by a person of ordinary skill in the art at the time of the priority date of the '968 Patent. In performing my analysis, I have applied the standard set forth above.

D. Summary of Opinions

31. I have reviewed and analyzed the '968 Patent.

32. Based on my review and analysis, it is my opinion that claims 1-9 of the '968 Patent are invalid based on the following grounds:

Netskope Exhibit 1002

Ground	Basis	References	Challenged Claims
1	§ 103	Jordan, FortiGate	1-9
2	§ 103	Lee, Magdych	1-9
3	§ 103	Lee, Magdych, Judge	1-9
4	§ 103	Fang, FortiGate	1-9

II. OVERVIEW OF THE TECHNOLOGY

A. Priority Date of the Claims

33. I have been informed that a U.S. patent application may claim the benefit of the filing date of an earlier patent application if the earlier patent application disclosed each limitation of the invention claimed in the later-filed U.S. patent application. I have also been informed that priority is determined on a claim-by-claim basis so that certain claims of a patent may be entitled to the priority date of an earlier-filed patent application even if other claims of the same patent are not entitled to that priority date.

34. I have also been informed that a patented claim is invalid if the claimed invention was patented, described in a printed publication, or in public use, on sale, or otherwise available to the public before the effective filing date of the claimed invention, or the claimed invention was described in an issued patent or a published

Netskope Exhibit 1002

patent application that was effectively filed before the effective filing date of the claimed invention.

35. I understand that due to written description issues, the challenged claims are entitled to a priority date of no earlier than November 5, 2013. But even if the Board finds adequate written description support, all references are prior art under pre-AIA 35 U.S.C. §§102(a), 102(b) and/or 102(e), except Fang.

B. Overview of Relevant Technology

1. Malware Detection

36. Preventing malicious traffic from entering a network has been a top concern for both the public and private sectors since the 1980s. That decade saw the first government-funded research into detecting and preventing cyberattacks as well as the first firewalls. Over time, various techniques were developed for detecting and remediating malicious traffic.

37. In general, network security monitoring involves analyzing both the inbound and outbound traffic of a network, e.g., an enterprise network, and deciding if the traffic is allowed. It is usually implemented as software running on a network device at the edge, e.g., the gateway, of the network. There are several kinds of network security monitoring technologies, including firewalls and network intrusion detection systems, which date back to the 1980's.

Netskope Exhibit 1002

38. A network security monitor enforces the security policies of an organization, which at the minimum specify access rules, e.g., how to authenticate a user, what access to which data and services can be granted to the user, etc., as well as protection rules, that is, what (potential) attacks, e.g., corporate data being sent to certain Internet sites or incoming email that contains malware, need to be identified and stopped and if necessary, recover from.

39. Intrusion detection systems (IDS), for example, were well-known by the 1990s. Those systems relied on rules to detect whether network traffic packets were suspicious, as shown in the figure below from a 1990 academic paper. (Ex. 1013.)

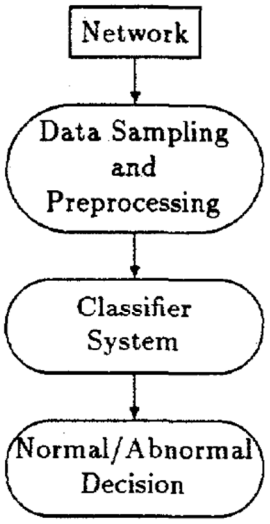


Figure 1: Basic architecture of intrusion detection system.

40. This figure illustrated the general process for an IDS system: a network device ran monitoring software and captured potentially-malicious traffic. After

Netskope Exhibit 1002

analyzing the traffic, the software determined whether to allow or block the traffic. (*Id.*, 4-5.)

41. Security policies are implemented by monitoring software as analysis rules. The network devices that run the monitoring software are responsible for capturing traffic data and providing the data to the monitoring software. Instead of capturing all traffic data, which can be in very large volume and thus overwhelm the monitoring software to a point that it cannot process data efficiently, the network devices can be configured to capture only the relevant traffic data, i.e., data relevant to security monitoring. The network devices also enforce the actions decided by the monitoring software, i.e., letting the traffic pass (i.e., letting it go to its intended destination) or blocking the traffic (i.e., not letting it go to its destination).

42. The 1998 "Bro" system was another example of a well-known system for detecting malware. (Ex. 1014.) Once the system received data, Bro's "Event Engine" constructed security-relevant event data (e.g., a "failed login attempt") according to predefined security policies. The "Policy Script Interpreter" then analyzed the event data to determine if a security policy violation occurred (e.g., "more than three failed login attempts"). If so, the system took the appropriate action, such as terminating the connection.

Netskope Exhibit 1002

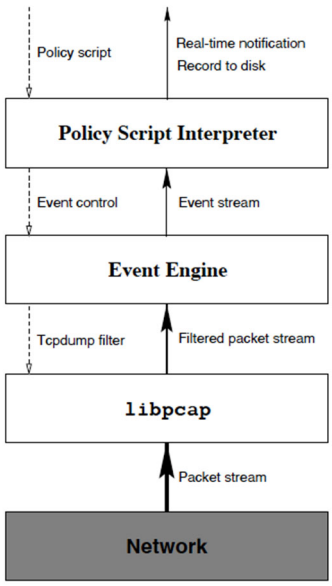


Figure 1: Structure of the Bro system

2. Updating Malware Detection Signatures

43. A malware detection signature generally relates to creating a pattern that indicates an attack or security violation. The malware detection signature is applied to incoming traffic and, if there is a match, the traffic is flagged and handled accordingly.

44. Using malware detection signatures dates back to at least 1993. Below, for example, is a figure from U.S. Patent No. 5,440,723 that depicted looking for a certain pattern in traffic to identify malicious data. (Ex. 1015.)

Netskope Exhibit 1002

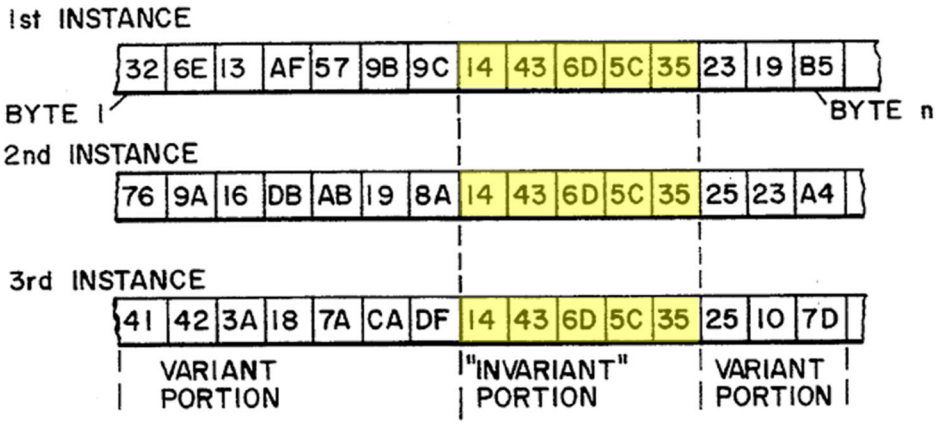


FIG. 6A (Annotated)

45. Given the importance of detection signatures, security systems needed both to generate new signatures in response to new threats and to update existing detection signatures in response to new information. This process could be performed manually or automated using machine learning algorithms, as illustrated by Dr. Lee's 1999 Ph.D. thesis ("A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems"). (Ex. 1016, 1.) ("Since many current IDSs are constructed by manual encoding of expert knowledge, changes to IDSs are expensive and slow. In this paper, we describe a data mining framework for adaptively building Intrusion Detection (ID) models.")

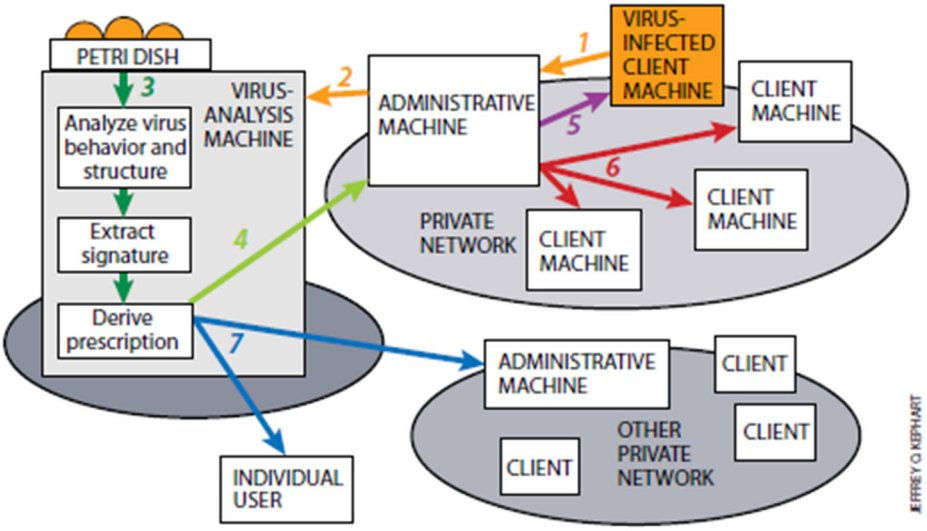
3. Distributed Malware Signature Updates

46. For decades, updated detection signatures have been distributed to multiple monitoring devices on a given network. Detection signature updates are usually carried out by a "back-end" analysis on a separate "update server" that runs

Netskope Exhibit 1002

the network security monitor. (*Id.*) The server can be operated by human analysts to manually produce the new and updated signatures or can run machine learning algorithms to automate these tasks. (*Id.*)

47. For example, around the late 1990's, companies such as IBM, Symantec Corporation, and McAfee Associates were developing technology that automatically detected new malware and responded to it. (Ex. 1017.) Below is a Figure and description of a security system that IBM was developing in the late 1990's where a central computer analyzed viruses. (*Id.*) Notably, the caption below the figure described a complete system for detecting and updating detection information.



DIGITAL IMMUNE SYSTEM for cyberspace is planned to work as shown here. An unknown virus spurs a client machine to forward a sample to an administrative machine (*1*), which in turn sends an encrypted sample to a central virus-analysis machine (*2*). This machine replicates the virus in a "petri dish" device and analyzes its behavior and structure (*3*). The resulting prescription is sent back to the administrative machine (*4*), which forwards it first to the infected client (*5*) and then to other machines on the local network (*6*). Subscribers around the world receive regular antivirus updates that protect them from the new virus (*7*).

Netskope Exhibit 1002

48. Updates to the active set of detection signatures were often distributed according to specific policies. For example, a traffic monitor for the engineering department might get a different signature than the traffic monitor for the finance department subnet because the two departments had different security policies and hence different signatures and rules. (*Id.*) Also, to manage the potentially large workload and customize updates according to the specific needs of different (sub)networks, there could be multiple update servers to accomplish "distributed updates." (*Id.*) For example, the engineering subnet and finance subnet could each have their own, dedicated servers, or two servers could split the total workload for the two departments. (*Id.*)

49. Detection signatures were created using information from multiple networks. For example, U.S. Pat. App. Pub. 2003/020632 disclosed a security system that "collected threats from about the world." (Ex. 1017., ¶57.) As illustrated in the Figure below, "[t]he collected information 130 [was] then analyzed and utilized by the network administrator 142 to design the detection signatures 132 (comprising, for example, security rules, policies and algorithms) that can be used by the system 10…." (*Id.*)

Netskope Exhibit 1002

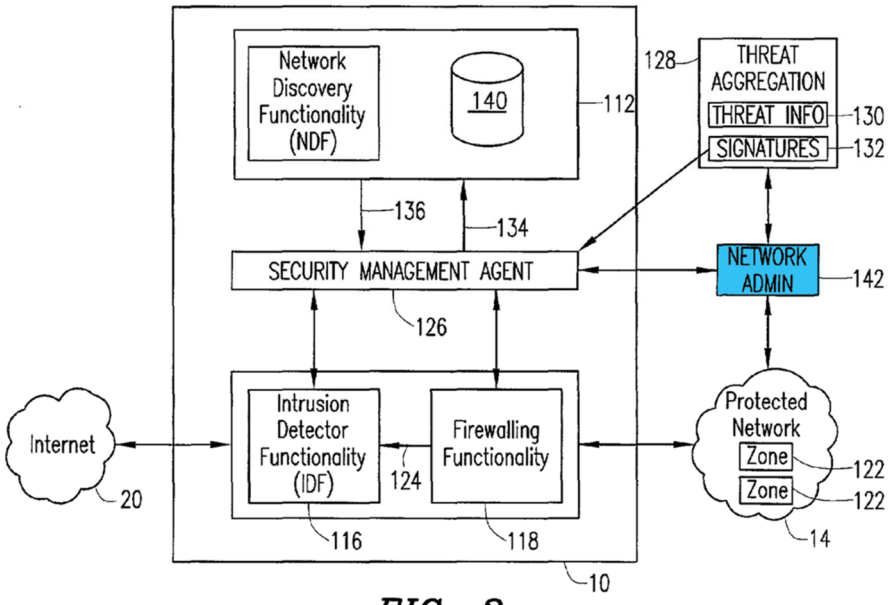


FIG. 2 (Annotated)

C. The '968 Patent

1. Alleged Invention

50. The '968 Patent concerns "[s]ystems, methods, and software for processing received network traffic content in view of content detection data and configuration data to either block, permit, or to further evaluate network traffic content when entering a network." ('968 Patent, Abstract.)

51. '968 Figure 1 (below) depicts an alleged content detection system. (*Id.*, 3:59-60.) It includes, *inter alia*, processing station (104), central station (102), update stations (106a-c), and content detection modules (108a-e), which could be gateways. (*Id.*, 4:11-18, 5:20-23, 5:34-35.)

Netskope Exhibit 1002

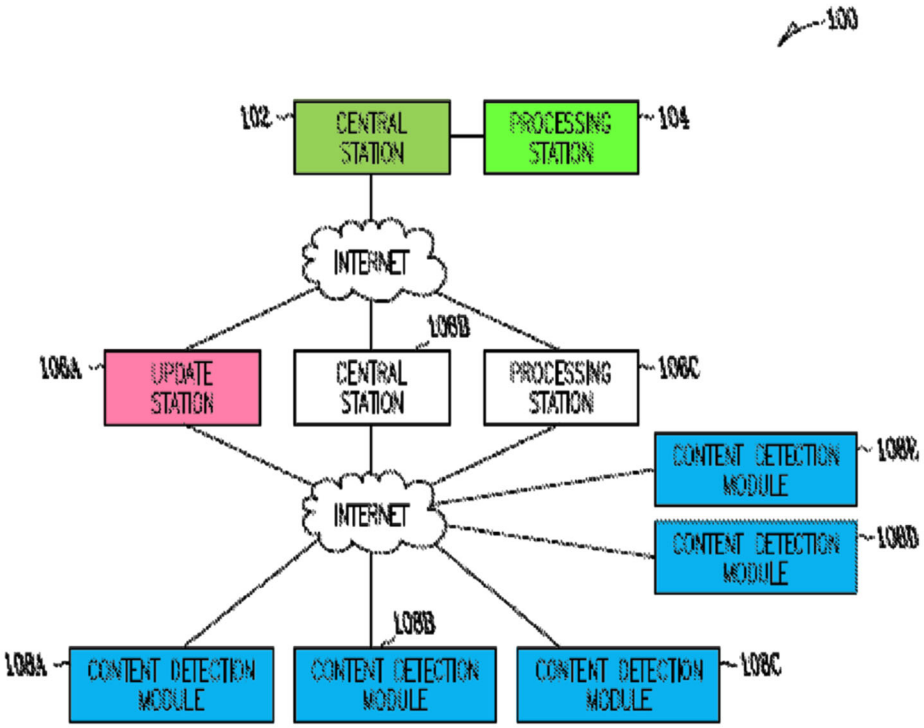


Fig. 1 (Annotated)

52. Petitioner redrew Figure 1 (annotated below) to make it more legible.[1]

[1] Elements 108b-c should read "update station."

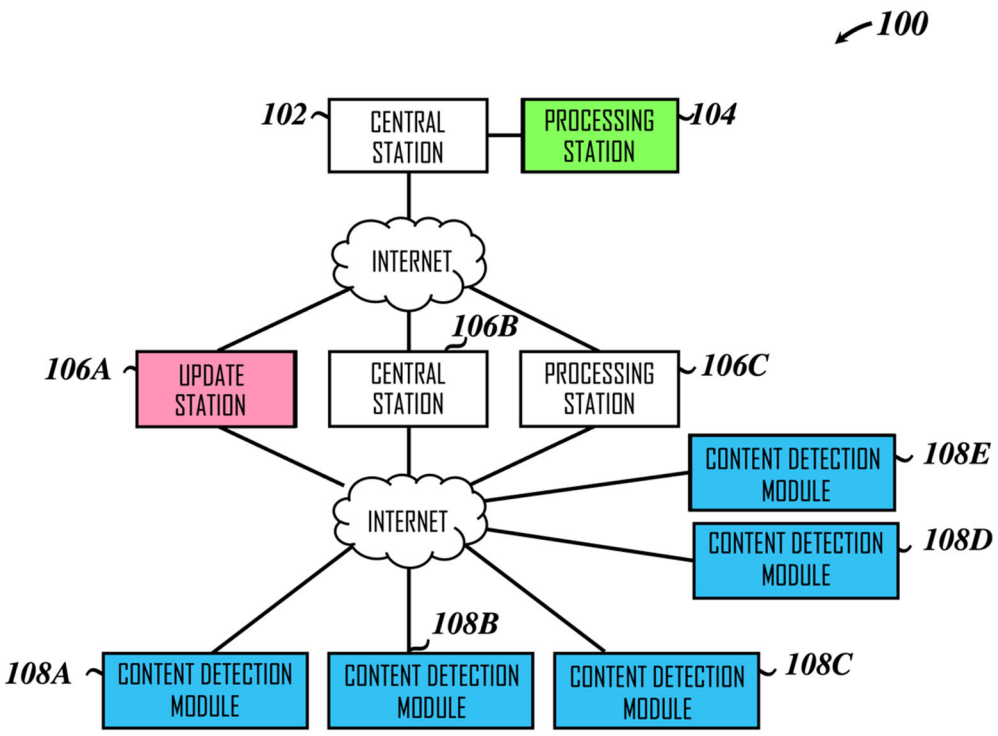


Fig. 1 (Redrawn and Annotated)

53. In operation, a CDM (108) receives content, such as a webpage or file. (*Id.*, 5:14-16, 5:64-6:2.) The CDM (108) determines if the content contains suspicious content. (*Id.*, 6:2-7.) If so, it sends the content to the processing station (104). (*Id.*) Processing station (104) then analyzes the content using "conventional or known techniques" to determine if the content poses a threat, such as containing a virus. (*Id.*, 6:12-15.) If the content is undesirable (e.g., is a threat), processing station (104) generates content detection data. (*Id.*, 6:19-33.) Content detection data broadly includes, *inter alia*, "virus signature, a virus definition, a spammer identification, a URL, a NIDS signature, a time at which content detection data is

Netskope Exhibit 1002

created, a level of threat, etc., and the like, e.g., any information that can be used in a content detection or screening process." (*Id.*, 6:33-38.)

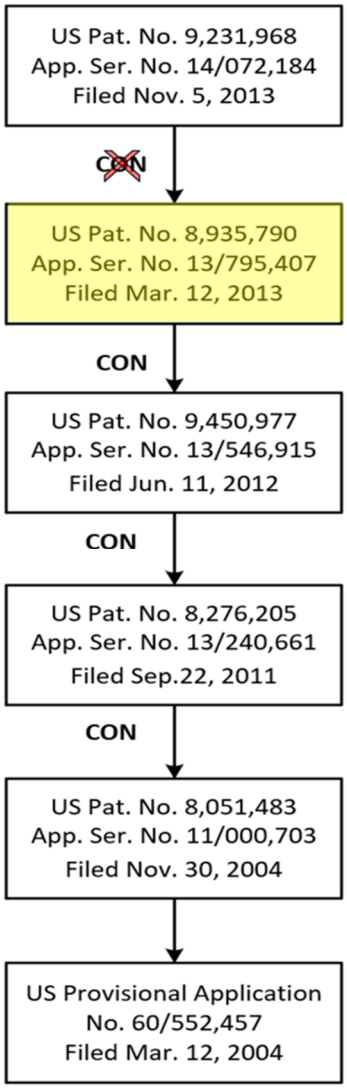
54. Once processing station (104) generates the content detection data, the processing station (104) sends the data to an update station (106). (*Id.*, 6:47-50.) Update station (106) then sends the content detection data to the CDM (108). (*See id.*, 6:55-58, 8:35-38.) The CDM (108) then uses the updated content detection data to detect the undesirable content. (*Id.*, 6:28-32, 8:59-62.) The CDM (108) is the focus of the '968 Patent's claims.

2. Priority Date

55. The '968 Patent issued from U.S. Patent Application No. 14/072,184, filed on November 5, 2013.

56. The '968 Patent claims priority to U.S. Patent Application No. 13/795,407, filed on March 12, 2013, as shown in the below priority chain.

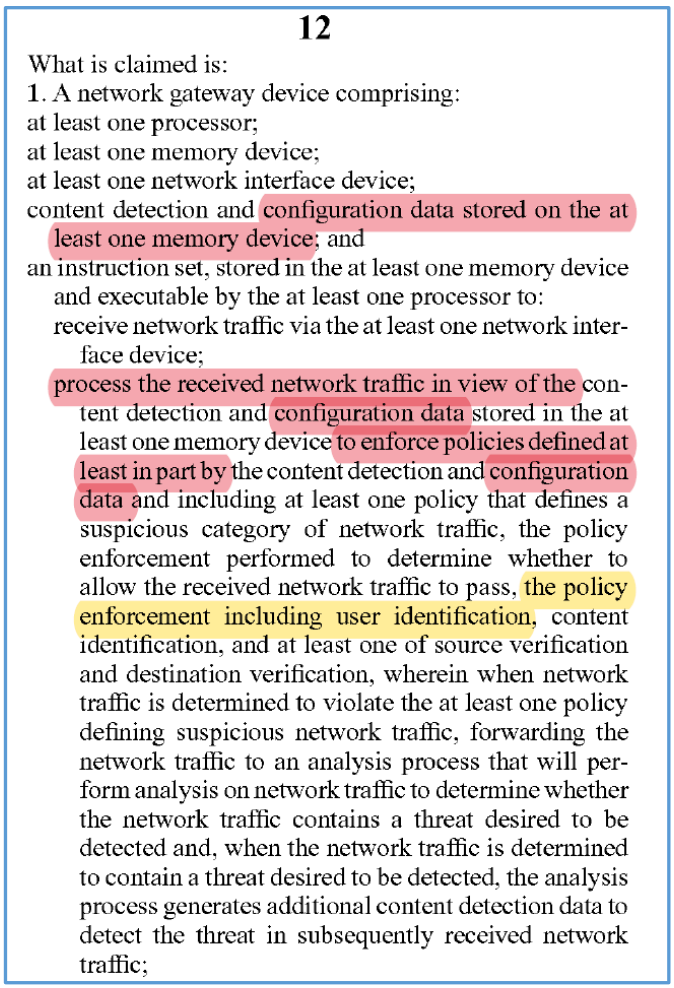
Netskope Exhibit 1002



57. But the challenged claims lack written description support in the '407 application because, *inter alia*, it does not disclose (1) "configuration data" and operations that implicate configuration data or (2) "user identification" and operations that implicate user identification. Therefore, the challenged claims are entitled to a priority date of no earlier than November 5, 2013, the date the '968 Patent was filed.

Netskope Exhibit 1002

58. The '968 Patent introduces at least two new terms and associated operations that lack written description support in the '407 application: (1) "configuration data" and operations that implicate configuration data (highlighted in red in exemplary claim 1 below[2]), and (2) "user identification" and operations that implicate user identification (highlighted in yellow).



12

What is claimed is:

1. A network gateway device comprising:

at least one processor;

at least one memory device;

at least one network interface device;

content detection and configuration data stored on the at least one memory device; and

an instruction set, stored in the at least one memory device and executable by the at least one processor to:

receive network traffic via the at least one network interface device;

process the received network traffic in view of the content detection and configuration data stored in the at least one memory device to enforce policies defined at least in part by the content detection and configuration data and including at least one policy that defines a suspicious category of network traffic, the policy enforcement performed to determine whether to allow the received network traffic to pass, the policy enforcement including user identification, content identification, and at least one of source verification and destination verification, wherein when network traffic is determined to violate the at least one policy defining suspicious network traffic, forwarding the network traffic to an analysis process that will perform analysis on network traffic to determine whether the network traffic contains a threat desired to be detected and, when the network traffic is determined to contain a threat desired to be detected, the analysis process generates additional content detection data to detect the threat in subsequently received network traffic;

59. **"Configuration Data" (Limitations 1[d]/1[e][ii]/1[e][iii]; 4[b][i]/4[b][ii]; 7[b][i]/7[b][ii])**: The independent claims require two distinct types

[2] Independent claims 4 and 7 contain similar limitations.

Netskope Exhibit 1002

of data that define policies and are used for processing received network traffic to enforce those policies—"content detection" data **and** "configuration data". While the claim term "content detection" data is described in several locations in the '968 Patent (*see, e.g.*, '968 Patent, 6:33-41), "configuration data" and its associated operations are not disclosed anywhere in the '968 Patent, the '407 application, or in any other application in the '968's priority chain. This fact alone indicates that "configuration data" lacks written description support.

60. Even searching the written description for something that might resemble "configuration data" fares no better. As noted, the claims require both "content detection data" and "configuration data" that define policies and are used for processing received network traffic by a network gateway device/content detection module. Yet the '407 application discloses neither of those uses with respect to configuration data.

61. Notably, the '407 application's only mention of "configuration" in the '407 application concerns the central station selecting the update stations in a "round robin configuration"—an entirely different device than the claimed CDM that performs an entirely different operation. (*See* '407 Application, p. 180, ¶ 37.) Indeed, every mention of the content detection module processing network traffic describes processing the network traffic using *only* "content detection data," And not "configuration data." (*See id.*, pp. 175-76, ¶29; p. 177, ¶32; pp. 181-82, ¶39; *see*

Netskope Exhibit 1002

also p. 275, ¶28; p. 178, ¶33; p. 180, ¶36; p. 12, ¶37; p. 181, ¶38; p. 182, ¶40; p. 182, ¶41.) The written description never describes "configuration data"—either expressly or impliedly—as recited in the claims.

62. **User Identification (Elements 1[e][v]/4[b][iv]/7[b][iv])**: While the challenged claims require that policy enforcement include, *inter alia*, "user identification," the '407 application provides no written description support for that limitation.

63. The '407 application mentions "user identification" in only two instances, neither of which related to policy enforcement. Both mentions concern a station (either central station (102) or update station (106)) receiving "subscriber data, such as a user identification of a content detection module 108… from processing station 104 or from update station(s) 106 for processing." (Ex. 1004 ('407 Application), p. 174, ¶27; p. 175, ¶28.)

Netskope Exhibit 1002

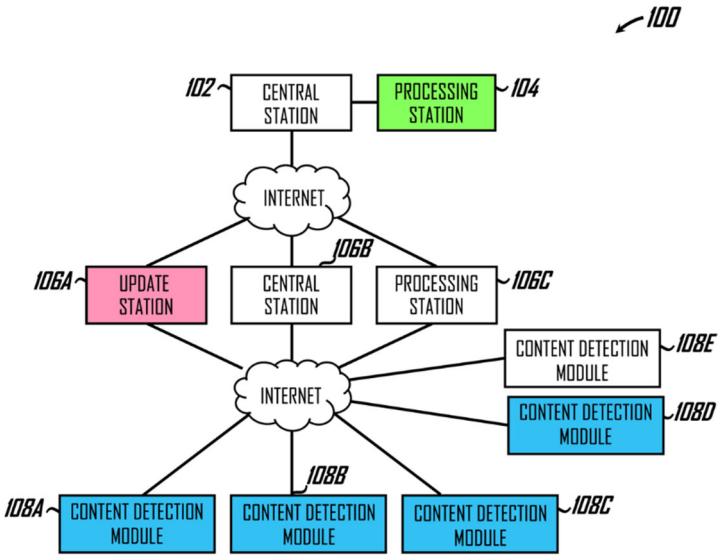


Fig.1 (Redrawn and Annotated)

64. Neither instance disclosed providing "user identification" to a content detection module/network gateway device, let alone that "user identification" is used in policy enforcement.

65. A different concept, "user authentication," is mentioned in connection with policy enforcement in the content detection module. (*See id*., 5:23-28.)

66. But, user identification and user authentication are technically distinct operations. User identification is often public, such as an email address or a username on a website. Conversely, authentication often uses private information, such as a password. (*Id.*) User identification is often a passive, simpler process where a user's identity (e.g., "Bob") is taken at face value, and a policy enforcement based on user identification can be, for example, checking against a list of allow/block identities (e.g., allow "Bob"). This operation is often repeated multiple

Netskope Exhibit 1002

times and offers no confidence in the user being who the user represents. Conversely, user authentication is a more complicated, less frequent process that determines if a user is who they claim to be, often involving cryptographic operations between the parties to establish a chain of trust. A POSITA would have not understood a security policy that was based on "user authentication" to be based on user identification. Notably, by using user identification instead of user authentication, Patent Owner has impermissibly broadened the scope of its claims to require a much simpler and commonplace operation of user identification to allow user access to data, which was not described or even contemplated in the original application.

67. On top of the written description issues, the claims of the '968 Patent are not enabled in the '407 application or in its detailed description because it failed to describe "configuration data" or "user identification" in "full, clear, concise, and exact terms as to enable a" POSITA to make and use the invention. 35 U.S.C. §112(a). Consequently, the '968 patent cannot claim priority to the '407 application because it did not enable a POSITA to make or use the claimed invention.

68. Given the above, the challenged claims introduce new matter and, consequently, cannot claim priority to any prior applications under 35 U.S.C. § 112. Therefore, the challenged claims should be entitled to a priority date no earlier than November 5, 2013.

Netskope Exhibit 1002

D. The Challenged Claims

69. I understand that the Petitioners are challenging claims 1-9 in the '968 Patent.

E. Claim Construction

70. I understand that Petitioner does not believe any claim constructions are needed here.

III. UNPATENTABILITY OF THE '968 PATENT'S CLAIMS

A. Standards for Invalidity

1. Obviousness

71. I am informed and understand that a patent cannot be properly granted for subject matter that would have been obvious to a person of ordinary skill in the art at the time of the alleged invention, and that a patent claim directed to such obvious subject matter is invalid under 35 U.S.C. § 103. It is also my understanding that in assessing the obviousness of claimed subject matter, one should evaluate obviousness in light of the prior art from the perspective of a person having ordinary skill in the art at the time the alleged invention was made (and not from the perspective of either a layman or a genius in that art). It is my further understanding that the question of obviousness is to be determined based on:

- The scope and content of the prior art;
- The difference or differences between the subject matter of the claim and the prior art (whereby in assessing the possibility of

Netskope Exhibit 1002

obviousness one should consider the manner in which a patentee and/or a Court has construed the scope of a claim);

- The level of ordinary skill in the art at the time of the alleged invention of the subject matter of the claim; and

- Any relevant objective factors (the "secondary indicia") indicating nonobviousness, including evidence of any of the following: commercial success of the products or methods covered by the patent claims; a long-felt need for the alleged invention; failed attempts by others to make the alleged invention; copying of the alleged invention by others in the field; unexpected results achieved by the alleged invention; praise of the alleged invention by the alleged infringer or others in the field; the taking of licenses under the patent by others and the nature of those licenses; expressions of surprise by experts and those skilled in the art at the subject matter of the claim; and whether the patentee proceeded contrary to accepted wisdom of the prior art.

- Any relevant objective factors (the "secondary indicia") indicating obviousness: independent invention of the claimed invention by others before or at about the same time as the named inventor thought of it; and other evidence tending to show obviousness.

2. Written Description & Enablement

72. I understand that a patent must satisfy the written description requirement and a separate and distinct enablement requirement. I further understand that the specification must "contain a written description of the invention, and of the

Netskope Exhibit 1002

manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same..." 35 U.S.C §112(a).

73. I understand that, to satisfy the written description requirement, the specification must describe the claimed invention in sufficient detail that one skilled in the art can reasonably conclude that the inventor had possession of the claimed invention at the time of filing. I understand that, to satisfy the enablement requirement, the specification must teach those skilled in the art how to make and use the full scope of the claimed invention without requiring "undue experimentation". I further understand that determining whether experimentation is 'undue,' so as to render a disclosure non-enabling, is made from the viewpoint of persons experienced in the field of the invention.

B. Ground I: Combining Jordan and FortiGate renders claims 1-9 obvious.

74. ***Jordan***: Prior to his invention, Jordan recognized the need for "a system or method capable of adequately protecting a network, such as a system or method that [was] ***capable of dynamically adapting to new threats***."[3] (*Id.*, 2:52-55.) With the U.S. Department of Defense's support, Jordan addressed that exact problem, inventing ways of "dynamically adapting to new threats in real-time (e.g., preventing

[3] Emphasis throughout the petition has been added by Petitioner unless indicated otherwise.

Netskope Exhibit 1002

so-called 'zero-day attacks'), or on an as-needed or on a 'just-in-time' (JTT) basis." (*Id.*, 2:65-3:1; *see also, id.*, 1:11-20, 1:24-25.)

75. Figure 1 (annotated below) depicted a network system according to Jordan. The network system included external network (150), traffic filter (110), switches (145(a)-145(b)), virtual space engine (130), analytical engine (140), and devices (120(a)-(d)). (*See id.*, 8:33-9:6.)

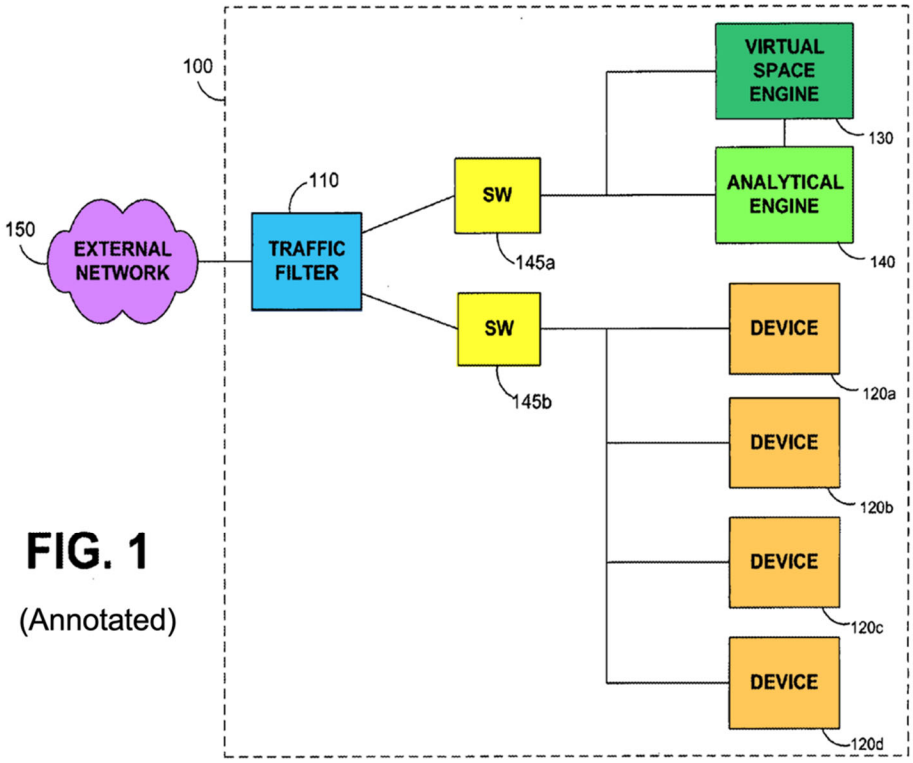


FIG. 1

(Annotated)

76. Figure 3 (below) depicted a process flow for handling traffic using this system. (*Id.*, 14:6-11.) The traffic filter (110) first received a communication (Step 302). (*Id.*) The traffic filter (110) then applied predefined rules (Step 304) to

Netskope Exhibit 1002

determine how to handle the communication. (*Id.*, 14:10-11, 15:19-22.) If the communication was normal, the traffic filter (110) passed the communication to its intended device (120(a)-(d)) (Step 308). (*Id.*, 14:19-24.) However, if an anomaly was detected, the traffic filter (110) passed the communication to the analytical engine (140) (Steps 306-310). (*Id.*, 14:27-36.) The analytical engine (140) then performed pattern recognition on the communication and, after doing so, generated new detection rules or updated existing ones (Steps 310 and 312). (*Id.*, 14:36-36, 14:49-53.) Those rules would be sent to the traffic filter (110), which used them to improve anomaly detection. (*Id.*, 14:54-56; *see also, e.g.*, *id.*, 8:33-9:64.)

Netskope Exhibit 1002

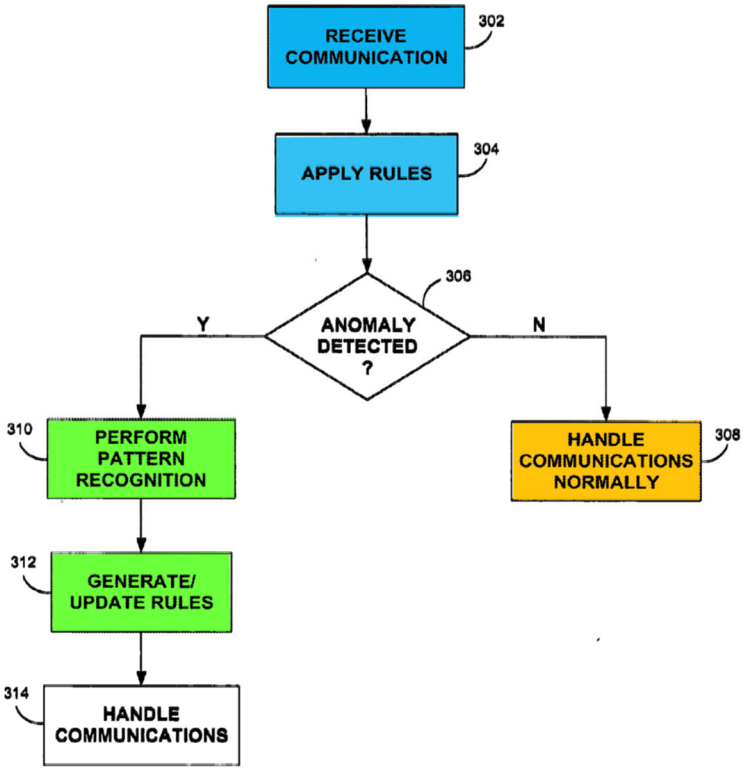


FIG. 3 (Annotated)

77. ***FortiGate***[4]: FortiGate described a "FortiGate-3600" firewall unit for enterprise networks. (*See*, FortiGate, 13.) FortiGate disclosed an "AntiVirus Firewall" that included "services such as virus protection and content filtering" and "firewall [and] intrusion detection…." (FortiGate, 13.) According to FortiGate, the FortiGate unit applied policies to filter traffic and protected network users from

[4] FortiGate was publicly available before the '968 Patent.

Netskope Exhibit 1002

external threats, like viruses. (*See* FortiGate, 13-16.) Below is an image of a FortiGate unit. (FortiGate, 13.)



78. FortiGate was provided by Patent Owner to the Patent Office during prosecution of another one of Patent Owner's patents (U.S. Patent No. 9,716,644 ("Wei").) During prosecution of Wei, Patent Owner submitted an IDS that included FortiGate. (*See* Wei's Prosecution History (Ex. 1020), Information Disclosure Statement, dated April 27, 2015 (FortiGate is listed on 6th page of document)). FortiGate is from the Patent Office's prosecution records of Wei.

79. In my professional opinion, FortiGate was publicly available in 2003 for a variety of reasons.

80. First, Patent Owner represented to the Patent Office that the publication date of the manual was 2003. Patent Owner submitted one-half of FortiGate in an IDS on April 27, 2015 during prosecution of the '968 Patent. (See '968 Prosecution History (Ex. 1003), Information Disclosure Statement, dated June 3, 2015 (listed on page 6th page of document).)

Netskope Exhibit 1002

81. In that IDS, Patent Owner represented that FortiGate's date was "2003." (*Id.*) Below is a portion of the IDS submitted by Patent Owner illustrating that FortiGate was published in 2003.

Modified form PTO/SB/08A(04-07) OMB 651-0031
US Patent & Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449A/PTO **INFORMATION DISCLOSURE STATEMENT BY APPLICANT** (Use as many sheets as necessary)	*Complete if Known*	
	Application Number	14/072,184
	Filing Date	November 5, 2013
	First Named Inventor	Yu Fang
	Group Art Unit	2432
	Examiner Name	Devin Almeida
Sheet 6 of 14	Attorney Docket No: 2831.006US6	

OTHER DOCUMENTS – NON PATENT LITERATURE DOCUMENTS		
Examiner Initial *	**Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.**	**T 1**
	"Exhibit G, Prior Art References Relevant To The '135 Patent, '205 Patent, And The '483 Patent", (3/2/15), 1-27	
	"F-Secure Anti-Virus 5 Win 95/98/ME/NT4.0/2000/XP DOS Administrator's Guide", F-Secure Corporation, (1996), 113 pgs	
	"FireEye's Disclosure of Invalidity Contentions", Fortinet, Inc. v. FireEye, Inc., US District Court of Northern District of California, San Franciso Division, Case No.5:13-cv-02496-HSG-PSG, (3/2/15), 38 pgs	
	"FortiGate - 3600 User Manual Part 1", Fortinet, (2003), 1-168	
	"FortiResponse Update Infrastructure", Fortinet, (2002), 10 pgs	

82. Additionally, Patent Owner made the same representation during prosecution of the Wei patent. During prosecution of Wei, Patent Owner submitted an IDS that included the entire FortiGate, and also represented that the publication date of FortiGate was "2003." (*See* Wei's Prosecution History (Ex. 1020), Information Disclosure Statement, dated April 27, 2015 (FortiGate is listed on 6th page of document)). Below is a portion of the IDS submitted by Patent Owner illustrating that FortiGate's date was "2003." Also, the first half of FortiGate (submitted during prosecution of Wei) and the first half of FortiGate (submitted during prosecution of the '968 patent) are mirror images, except for Bates Nos. on one copy. (*Compare* Ex. 1008 *with* Ex. 1021.)

Netskope Exhibit 1002

Modified form PTO/SB/08A(04-07) OMB 651-0031
US Patent & Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449A/PTO **INFORMATION DISCLOSURE STATEMENT BY APPLICANT** (Use as many sheets as necessary)	*Complete if Known*	
	Application Number	13/795,283
	Filing Date	March 12, 2013
	First Named Inventor	Shaohong Wei
	Group Art Unit	2462
	Examiner Name	Kevin Harper
Sheet 6 of 14	Attorney Docket No: 2831.012US4	

OTHER DOCUMENTS – NON PATENT LITERATURE DOCUMENTS		
Examiner Initial *	**Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.**	**T 1**
	"Exhibit F-2, U.S. Patent No. 7,979,543, Invalidity Claim Chart based on the Arbor Networks System", (3/2/15), 1-31	
	"Exhibit F-3, U.S. Patent No. 7,979,543, Invalidity Claim Chart based on the McAfee Prior Art System", (3/2/15), 1-40	
	"Exhibit F-4, U.S. Patent No. 7,979,543, Invalidity Claim Chart based on NetScreen Prior Art System", (3/2/15), 1-47	
	"Exhibit F-5, U.S. Patent No. 7,979,543, Invalidity Claim Chart based on the Symantec Prior Art System", (3/2/15), 1-31	
	"Exhibit F-6, U.S. Patent No. 7,979,543, Invalidity Claim Chart based on U.S. Patent No. 7,725,544 to Alspector", (3/2/15), 1-29	
	"Exhibit F-7, U.S. Patent No. 7,979,543, Invalidity Claim Chart based on U.S. Patent No. 7,320,020 to Chadwick", (3/2/15), 1-23	
	"Exhibit F-8, U.S. Patent No. 7,979,543, Invalidity Claim Chart based on U.S. Patent No. 8,046,832 to Goodman", (3/2/15), 1-29	
	"Exhibit F-9, U.S. Patent No. 7,979,543, Invalidity Claim Chart based on U.S. Patent No. 6,088,804 to Hill", (3/2/15), 1-27	
	"Exhibit G, Prior Art References Relevant To The '135 Patent, '205 Patent, And The '483 Patent", (3/2/15), 1-27	
	"F-Secure Anti-Virus 5 Win 95/98/ME/NT4.0/2000/XP DOS Administrator's Guide", F-Secure Corporation, (1996), 113 pgs	
	"FireEye's Disclosure of Invalidity Contentions", Fortinet, Inc. v. FireEye, Inc., US District Court of Northern District of California, San Franciso Division, Case No.5:13-cv-02496-HSG-PSG, (3/2/15), 38 pgs	
	"FortiGate - 3600 User Manual Part 1", Fortinet, (2003), 1-168	
	"FortiGate - 3600 User Manual Part 2", Fortinet, (2003), 169-329	
	"FortiResponse Update Infrastructure", Fortinet, (2002), 10 pgs	
	"Hardware Guide NetScreen-IDP V/N 2.1", NetScreen Technologies, (2003), 114 pgs	
	"High Availability QuickStart Guide IDP 100 & 500", NetScreen Technologies Inc., (2003), 32 pgs	

83. Second, FortiGate contains various statements that are typical of a publicly distributed document. For example, FortiGate is titled as a "User Guide" for Patent Owner's product, the "FortiGate-3600," also referred to as the "FortiGate unit." (FortiGate (Ex. 1008), 1, 13.) FortiGate's cover page lists a version and date: "Version 2.50 … September 5, 2003." (*Id.*, 1.) FortiGate's second page contains a copyright notice stating "© Copyright 2003 Fortinet Inc. All rights reserved." (*Id.*, 2.)

Netskope Exhibit 1002

84. Moreover, FortiGate is listed as a "Installation and Configuration Guide" for the FortiGate units. (*Id.*, 2.) FortiGate also contains a statement that no part of the text may be reproduced "without prior written permission of Fortinet Inc." (*Id.*, 2.) Readers were directed to contact "http://www.fortinet.com" for technical support. Readers were also directed to "Send information about errors or omissions in this document or any Fortinet technical documentation to techdoc@fortinet.com." (*Id.*) The document also contains a "Regulatory Compliance" section stating the product complies with "FCC Class A Part 15 CSA/CUS." (*Id.*, 2.)

85. In my professional opinion, all of these facts indicate that FortiGate was publicly distributed in 2003. First, Patent Owner itself represented that its own document was published in 2003 when it provided the document to the Patent Office during prosecution. Second, documents distributed to the public contain versions, dates, copyright notices, prohibitions on copying, contact information for technical support, contact information for issues the reader might find within the document, and regulatory compliance statements. In contrast, internal documents typically do not contain such statements or provisions. Moreover, FortiGate was represented as a user guide for Patent Owner's FortiGate-3600 network security unit. User guides are meant to be publicly distributed to purchasers at time of the purchase, and I see no NDA or other confidentiality agreement restricting the viewership of FortiGate.

Netskope Exhibit 1002

86. Accordingly, FortiGate was publicly available as early as 2003 in my professional opinion. Thus, FortiGate is prior art to the '968 Patent.

1. Motivation to Combine

87. A POSITA would have been motivated to combined Jordan and FortiGate with a reasonable expectation of success.

88. Both Jordan and FortiGate concerned network and data security. Jordan disclosed systems for detecting and handling anomalous/malicious communications. (Jordan, 5:28-30, 5:36-39.) Jordan further disclosed generating or updating rules to improve malware detection. (Jordan, 6:37-46; *see also id.*, FIG 1, 5:21-54, 8:33-9:6, 9:33-50.) For example, Jordan described a system that dynamically adapted to new threats to limit attacks on a network. (Jordan, 2:52-3;5, 9:33-50.) Likewise, FortiGate concerned analyzing and filtering network traffic. (FortiGate, 13, 14, 16, 20.) For example, FortiGate described an intrusion detection system that analyzed content (e.g., network traffic) in "real-time" and downloaded new/updated virus definitions to identify suspicious traffic. (FortiGate, 13, 16, 20.) Thus, a POSITA looking to improve Jordan would have look to complementary references like FortiGate.

89. Element-specific motivations to combine are discussed below.

2. Claims 1, 4, and 7

1[pre]: "A network gateway device comprising:"

90. Jordan disclosed a traffic filter (network gateway device).

Netskope Exhibit 1002

91. Figure 1 (below) depicted traffic filter (110), which could be "used as a ***perimeter device*** to ***control all communications crossing a network boundary***." (Jordan, 5:28-30 (emphasis added), Fig. 1.)

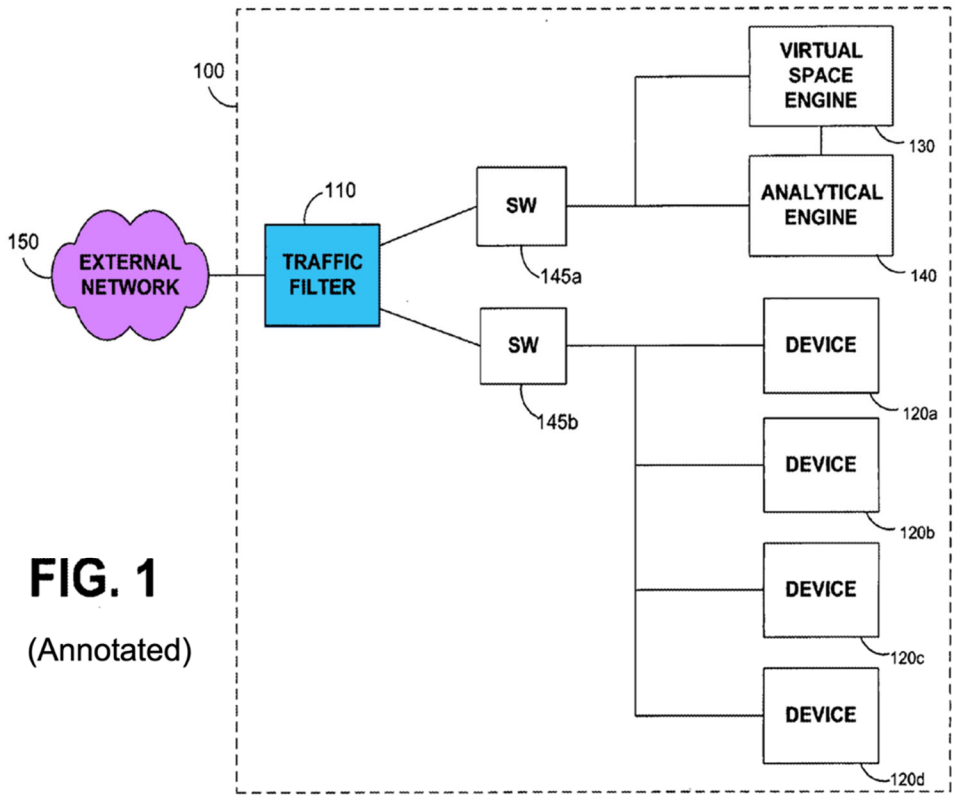


FIG. 1

(Annotated)

92. The '968 Patent described the network gateway device in similar fashion. Like the traffic filter, the gateway device sits between an external network (e.g., Internet) and a local network. (*See* '968 Patent, 5:20-23.) Indeed, FortiGate—written by Fortinet—described a "gateway" as simply something "that links different networks." (FortiGate, 297.) Accordingly, a POSITA would have understood that

Netskope Exhibit 1002

traffic filter was a network gateway device because both were perimeter devices, that linked networks.

4[pre]: "A method comprising:"

93. Jordan disclosed a "method for defending a computer network." (*Id*., 2:63-65.)

7[pre]: "A non-transitory computer-readable medium with instructions stored thereon which when executed by at least one processor of a data network device, causes the data network device to:"

1[a]: "at least one processor;"

1[b]: "at least one memory device;"

94. Jordan disclosed "a ***processor-readable medium*** ***comprising code representing instructions*** to cause a ***processor*** to" perform functions of the claimed invention. (Jordan, 3:6-8 (emphasis added); *see also id.*, 3:25-30, 25:29-32, FIG. 11.) A POSITA would have understood that a "processor-readable medium" was a "non-transitory computer-readable medium" because a processor is the component of a computer that reads memory. (*See, e.g.*, '968 Patent, 10:49-51 ("The term 'computer-readable medium' as used herein refers to any medium that participates in providing instructions to processor 204 for execution.").)

95. A POSITA would have also understood that processor-readable medium referred to a memory device because the processor-readable medium stored code or instructions.

Netskope Exhibit 1002

96. Jordan's traffic filter was a data network device. Traffic filter performed—using processor-readable medium, instructions, and a processor—data network functions, such as controlling communications that crossed a network boundary. (*See* element 1[pre]; Jordan, 5:28-30; *see also id.*, 3:6-40). Those functions were illustrated in Figure 14 (below).

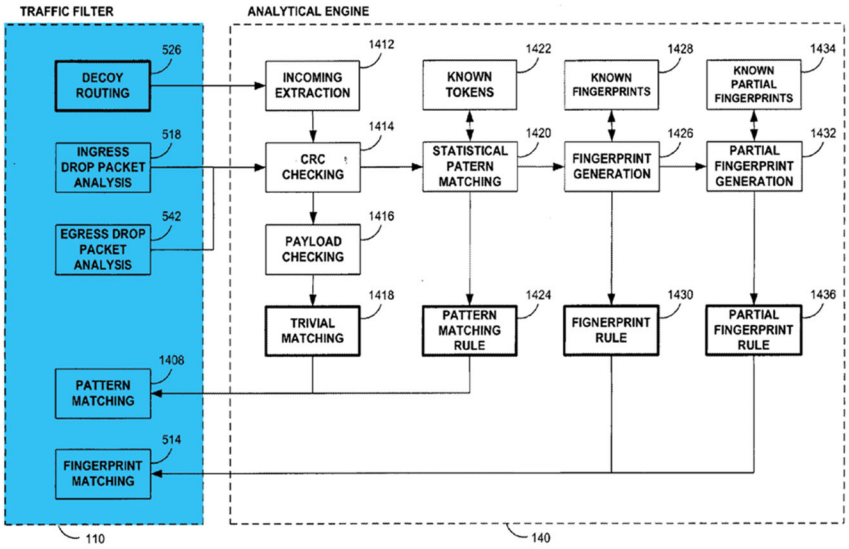


FIG. 14 (Annotated)

1[c]: "at least one network interface device;"

97. Jordan disclosed or strongly suggested that the traffic filter had a network interface device because it received and transmitted network communications.

Netskope Exhibit 1002

98. Figure 1 (below) showed that the traffic filter received traffic from an external network and routed it accordingly. (Jordan, 3:14-40, 14:19-36, 11:1-24, Figs. 1, 3, 5, 11.)

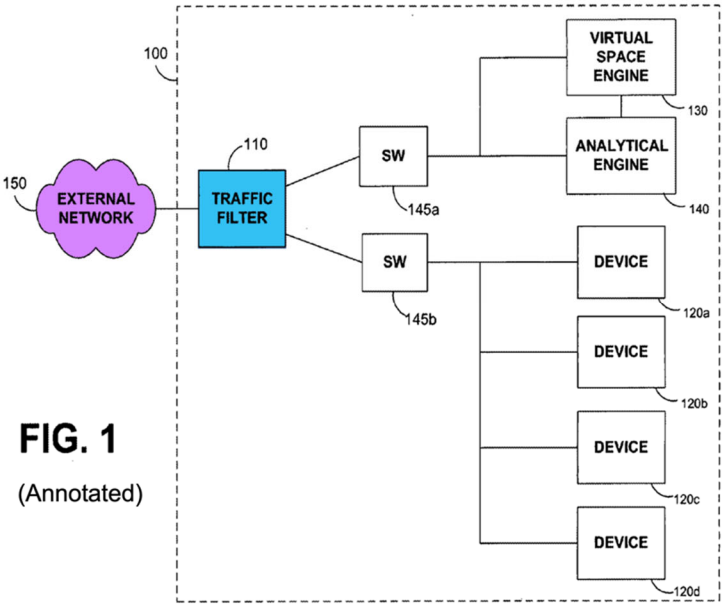


FIG. 1
(Annotated)

99. For example, Figure 5 (below) depicted "operations associated with handling incoming communications and operations associated with handling outgoing communications" in the traffic filter. (*Id.*, 17:46-49.) Jordan further explained that the traffic filter communicated with "other devices or network components within its network by ***any suitable means*** (e.g., via standard wired or wireless communication protocols)." (*Id.*, 17:53-57; *see also, e.g.*, *id.*, 11:10-24.)

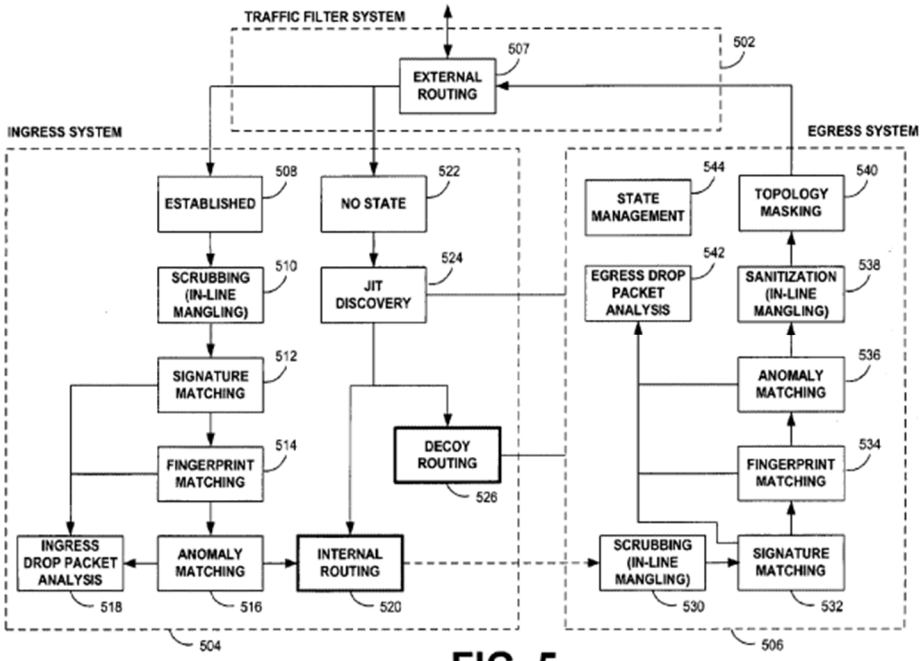


FIG. 5

100. Additionally, Figure 11 (below) showed the traffic filter had a communication module for communicating with the analytical engine. (*Id.*, 25:40-42.)

Netskope Exhibit 1002

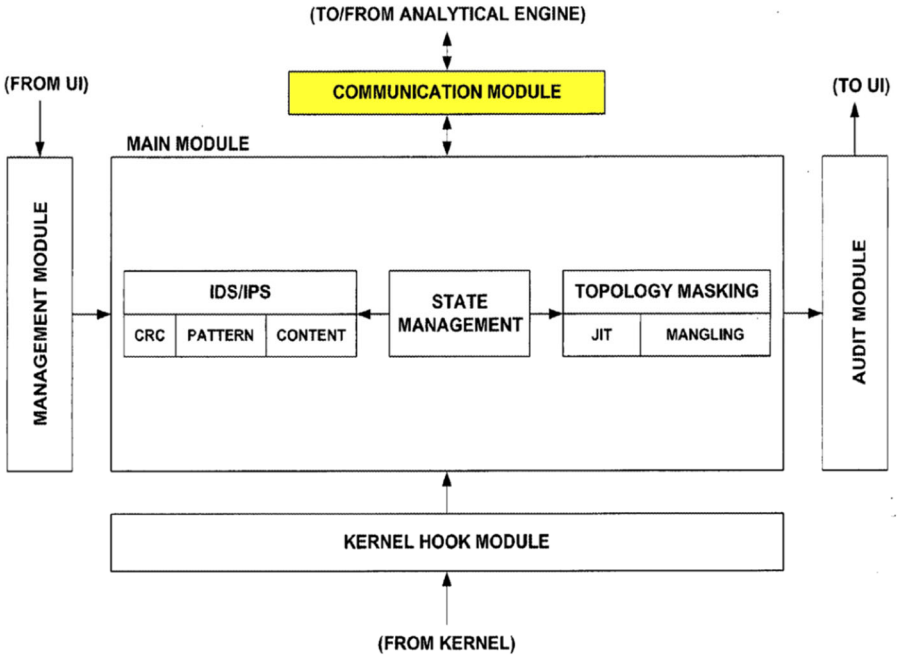


FIG. 11 (Annotated)

101. Accordingly, Jordan clearly disclosed or suggested a network interface device. Indeed, network interface devices were ubiquitous before the '968 Patent and would have been installed in any network-enabled device. For example, the '968 Patent even describes that the communication interface (218) can be "an integrated services digital network (SDN) card or a modem to provide a data communication connection a corresponding type of telephone line. As another example, communication interface 218 may be a local area network (LAN) card to provide a data communication between connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 218 sends and receives electrical, electromagnetic or optical signals that

Netskope Exhibit 1002

carry data streams representing various types of information." ('968 Patent, 11:20-34.) Such interface cards were routine.

1[d]: "content detection and configuration data stored on the at least one memory device; and"

102. Jordan disclosed that the traffic filter stored content detection data (e.g., virus lists or threat lists) on memory and suggested configuration data.

103. If Jordan did not disclose configuration data, a POSITA would have found it obvious to add it based on FortiGate's disclosures.

104. ***"content detection [data]"***: Jordan disclosed that the traffic filter filtered data using a variety of information (e.g., virus lists, threat lists).

105. For context, the '968 Patent broadly describes "content detection data" as follows:

> "By ***means of non-limiting examples,*** content detection data can be a ***virus signature***, ***a virus definition***, a spammer identification, a URL, a NIDS signature, a time at which content detection data is created, a level of threat, ***etc., and the like, e.g., any information that can be used in a content detection or screening process***."

('968, 6:33-38 (emphasis added).)

106. Jordan disclosed a wide variety of content detection data, including virus and threat lists. Figure 11 (below) showed that, traffic filter contained modules for intrusion detection systems/intrusion protection systems. (*Id.*, 25:61-63.) Those

Netskope Exhibit 1002

modules included, for example, "***pattern and content checking*** modules, which [could] ***search certain patterns and/or contents*** … of a communication data stream." (*Id.*, 25:67-26:3 (emphasis added).)

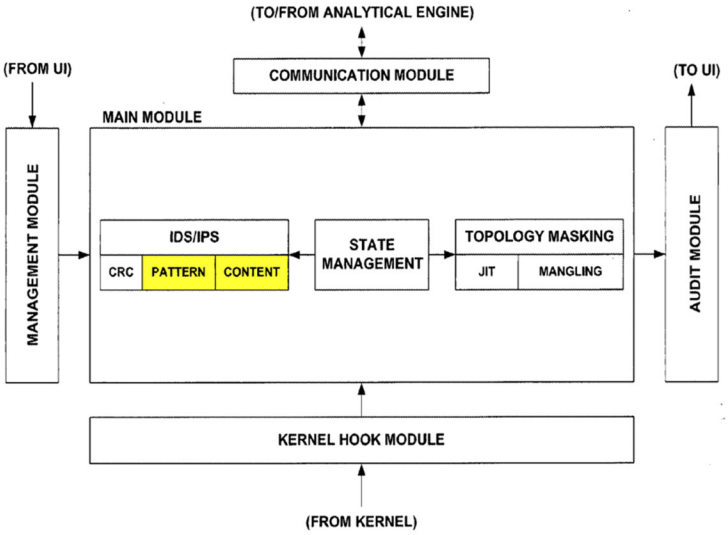


FIG. 11 (Annotated)

(*Id.*, Fig. 11.) Jordan explained that the traffic filter used various data when processing traffic for suspect content. For example, the data included "***lists of patterns (e.g., virus lists, threat lists, etc.)***." (*Id.*, 16:42-44 (emphasis added); *see also, e.g.*, 17:39-41 ("***[T]raffic filter can use other rules, such as predetermined rules (e.g., virus lists)***, or other preexisting rules.") (emphasis added).) The data could also include signatures, fingerprints, and payloads for matching. (*See id.*, 18:20-26 ("[I]ngress system 504 [of traffic filter] can perform additional functions on the incoming communication, such as ***signature matching***…and ***fingerprint***

Netskope Exhibit 1002

matching.") (emphasis added), 17:36-38 ("More specifically, a 'trivial rule' is a rule configured to handle an attack that always has the same payload."); 5:56-57.)

107. This data was stored in the traffic filter. (Jordan, 9:35-40, 14:9-12, 14:54-56, 15:29-31.) Figure 14 (below) showed that the rules were sent from the analytical engine (140) to the traffic filter (110), which applied the rules to recognize a pattern. (*See also, e.g.*, *id.*, 14:31-33 ("[T]he traffic filter 110 can ***apply those rules to recognize a pattern within the received communications***.").) Indeed, a POSITA would have understood that the content detection data (e.g., a virus signature) was stored on the traffic filter because it would need local access to that data to process traffic.

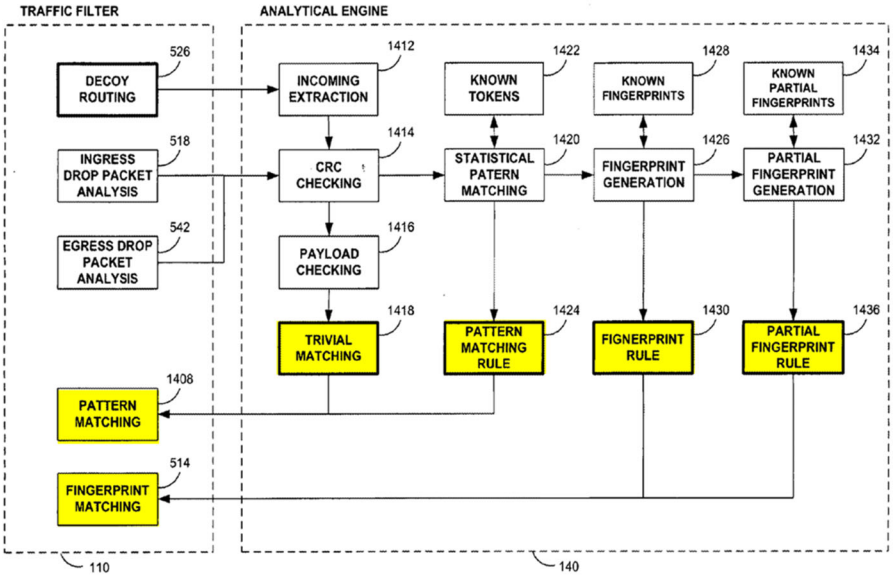


FIG. 14 (Annotated)

108. ***"configuration data"***: Jordan explained that traffic filter (110) had a management module (shown in Figure 5, below). The management module provided an interface between traffic filter (110) and a system administrator. (*Id.*, 25:50-51.)

109. Using the module, an administrator could "***manage rules*** of the traffic filter 110, ***set system parameters***, and/or read system logs." (*Id.*, 25:52-54 (emphasis added).)

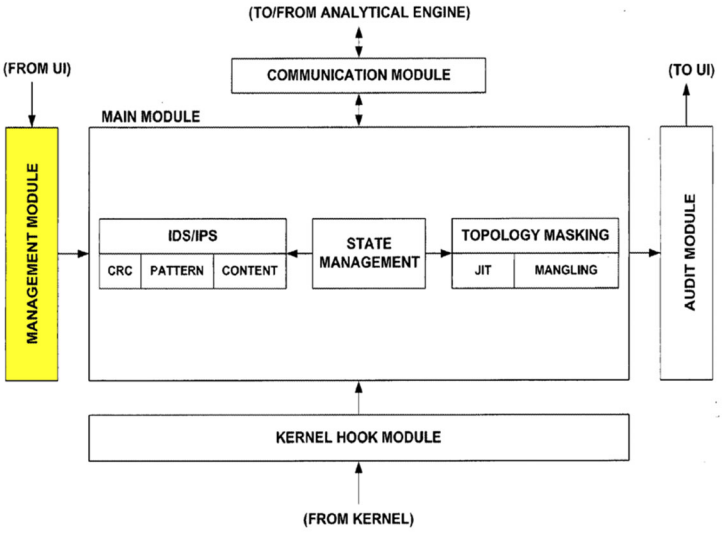


FIG. 11 (Annotated)

110. Thus, Jordan recognized that its system could be configured, it simply did not explain what those configurations could be. But FortiGate did.

111. FortiGate described a firewall configuration (shown below). An administrator could configure various things, like "Service," "Action," "NAT,"

Netskope Exhibit 1002

"Traffic Shaping," "Authentication," and "Antivirus & Web Filter." (FortiGate, 34-35; *see also id.*, 16.)

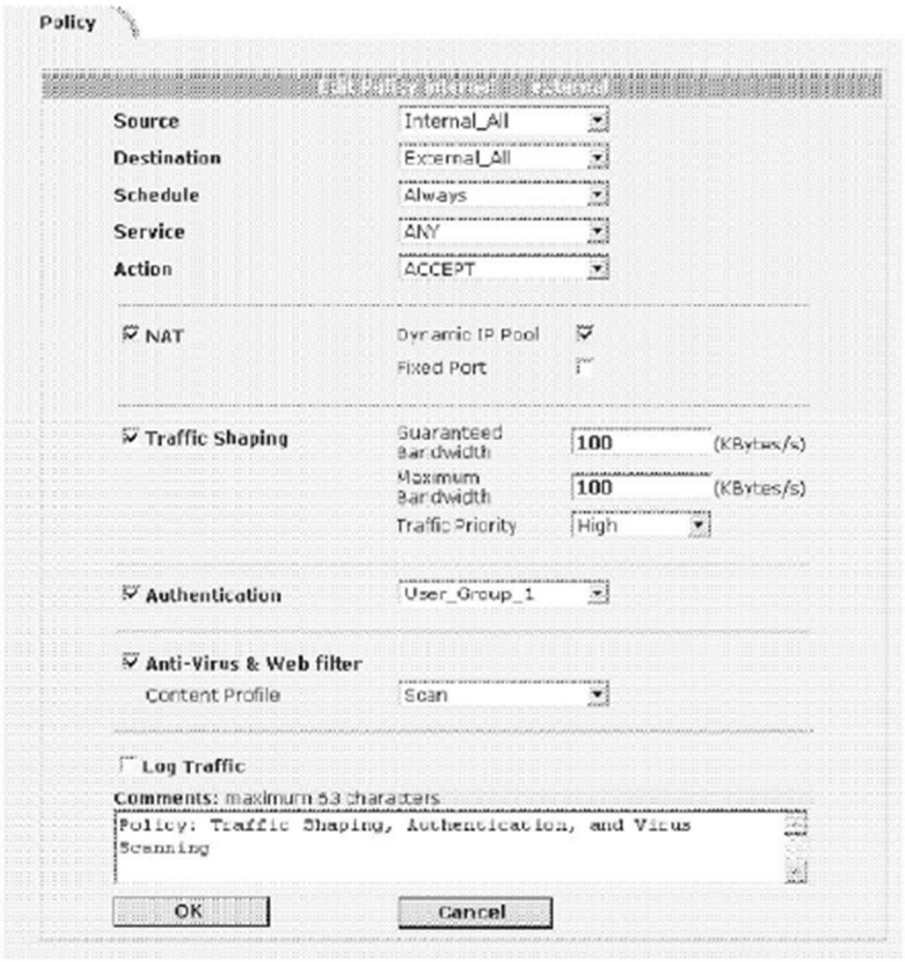
Table 4: Factory default firewall configuration

Internal Address	Internal_All	IP: 0.0.0.0 Mask: 0.0.0.0	Represents all of the IP addresses on the internal network.
External Address	External_All	IP: 0.0.0.0 Mask: 0.0.0.0	Represents all of the IP addresses on the external network.
Recurring Schedule	Always		The schedule is valid at all times. This means that the firewall policy is valid at all times.
Firewall Policy	**Internal->External**		Firewall policy for connections from the internal network to the external network.
	Source	Internal_All	The policy source address. Internal_All means that the policy accepts connections from any internal IP address.
	Destination	External_All	The policy destination address. External_All means that the policy accepts connections with a destination address to any IP address on the external network.
	Schedule	Always	The policy schedule. Always means that the policy is valid at any time.
	Service	ANY	The policy service. ANY means that this policy processes connections for all services.
	Action	ACCEPT	The policy action. ACCEPT means that the policy allows connections.
	☑ **NAT**		NAT is selected for the NAT/Route mode default policy so that the policy applies network address translation to the traffic processed by the policy. NAT is not available for Transparent mode policies.
	☐ **Traffic Shaping**		Traffic shaping is not selected. The policy does not apply traffic shaping to the traffic controlled by the policy. You can select this option to control the maximum or minimum amount of bandwidth available to traffic processed by the policy.
	☐ **Authentication**		Authentication is not selected. Users do not have to authenticate with the firewall before connecting to their destination address. You can configure user groups and select this option to require users to authenticate with the firewall before they can connect through the firewall.
	☐ **Antivirus & Web Filter**		Antivirus & Web Filter is not selected. This policy does not include a content profile that applies antivirus protection, web content filtering, or email filtering to content traffic processed by this policy. You can select this option and select a content profile to apply different levels of content protection to traffic processed by this policy.
	☐ **Log Traffic**		Log Traffic is not selected. This policy does not record messages to the traffic log for the traffic processed by this policy. You can configure FortiGate logging and select Log Traffic to record all connections through the firewall that are accepted by this policy.

Netskope Exhibit 1002

The figure below illustrates the policy configuration through the user interface. (*Id.*, 173.)

Figure 5: Adding a NAT/Route policy



112. FortiGate also disclosed other configurations. (See, e.g., id., 14 (for antivirus protection: "[c]onfigur[ing] antivirus protection to block files of specified file types from passing through the FortiGate unit"), 14 ("You can configure URL blocking to block all or just some of the pages on a web site.").)

113. Incorporating these configurations into Jordan's traffic filter would have improved its flexibility and available features for filtering and processing traffic. For example, configuration data that indicated a user needed to be authenticated would instruct the policy to require an additional authentication step before allowing a user to access data. These configurations would naturally be incorporated into Jordan's traffic filter, which processed traffic like the FortiGate firewall.

114. ***"stored on the at least one memory device"***: Jordan did not explicitly disclose that its content detection data and configuration data were stored on at least one memory device. However, Jordan suggested it and a POSITA would have found it obvious.

115. Jordan's traffic filter (110) applied rules for processing traffic. (Jordan, 14:9-11, 16:42-47, 14:30-34; *see id.*, 5:21-54.) A POSITA would have found it obvious to store the content detection data and configuration data on the filter's memory because the traffic filter would need to retrieve the data when processing traffic or after it was restarted.

1[e]: "an instruction set, stored in the at least one memory device and executable by the at least one processor to:"

116. Jordan disclosed or strongly suggested an instruction set stored in memory of the traffic filter and executable by a processor. (*See, e.g.*, *id.*, 3:25-30)

Netskope Exhibit 1002

(disclosing "a processor-readable medium comprising code representing instructions to cause a processor to [perform functions].".)

1[e][i]/4[a]/7[a]: "receive[-ing] network traffic via the at least one network interface device;"

117. Jordan disclosed that the traffic filter received network traffic (e.g., incoming communications).

118. Jordan explained that "the network system 100 include[d] a traffic filter 110, which [was] ***configured to receive communications*** from the external network 150 , and route those communications within the network system 100." (*Id.*, 8:39-42 (emphasis added); *see also, e.g.*, 5:23-25 (disclosing the traffic filter (110) "handle[d] incoming communications (i.e., communications received from sources external to the network of the traffic filter)….").)

119. Figure 3 (below) also showed that traffic filter (110) received communications at Step 302. (*Id.*, 14:9-11 ("Specifically, the ***communications [were] received in operation 302*** , and rules [were] applied by the traffic filter 110 in operation 304.") (emphasis added); *see also, e.g.*, 15:50-53 ("***traffic filter 110 receive[d] a communication, such as a network communication*** from a device external to the network system 100.").) A POSITA would have understood that the communications were network traffic because they were data transmitted over a network.

Netskope Exhibit 1002

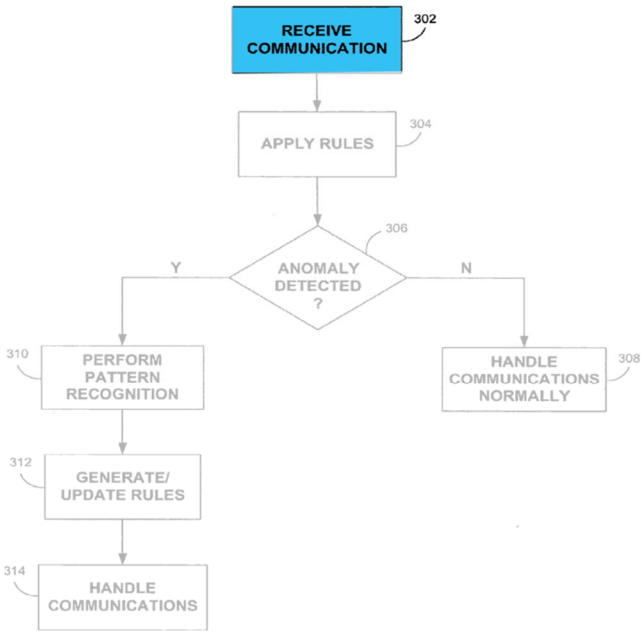


FIG. 3 (Annotated)

120. As discussed for element 1[c], Jordan disclosed or suggested that the traffic filter had a network interface device for receiving communications. Thus, in the above disclosures, a POSITA would have understood that those communications were received over the network interface. (*See, e.g.*, *id.*, 15:50-53 ("traffic filter 110 receive[d] a communication, such as a network communication from a device external to the network system 100 (e.g., via an external network 150.").)

1[e][ii]/4[b][i]/7[b][i]: "process[ing] the received network traffic[, by executing instructions on at least one processor of a data network device,] in view of the content detection and configuration data stored in the at least one memory device"

Netskope Exhibit 1002

121. Jordan disclosed that the traffic filter (110) processed received network traffic in view of content detection data (e.g., virus lists), and FortiGate disclosed processing network traffic in view of configuration data (e.g., Traffic Shaping).

122. Jordan disclosed that the traffic filter applied rules and performed pattern recognition on communications (Steps 304, 310 below). (*See, e.g.*, Jordan, 14:9-11.)

123. The rules included content detection data. (*See* Jordan, 5:55-6:2, 8:33-57.) For example, if traffic filter detected an anomaly in network communications, "***pattern recognition [could] be performed*** on the communication…." (*Id.*, 14:27-28 (emphasis added).) More specifically, "traffic filter 110 appl[ied] [rules predetermined by analytical engine 140] ***to recognize a pattern within the received communications***." (*Id.*, 14:31-33 (emphasis added).) As discussed for limitation 1[d], the content detection data included virus lists. (*See, e.g.*, *id.*, 16:42-44, 17:39-41.)

Netskope Exhibit 1002

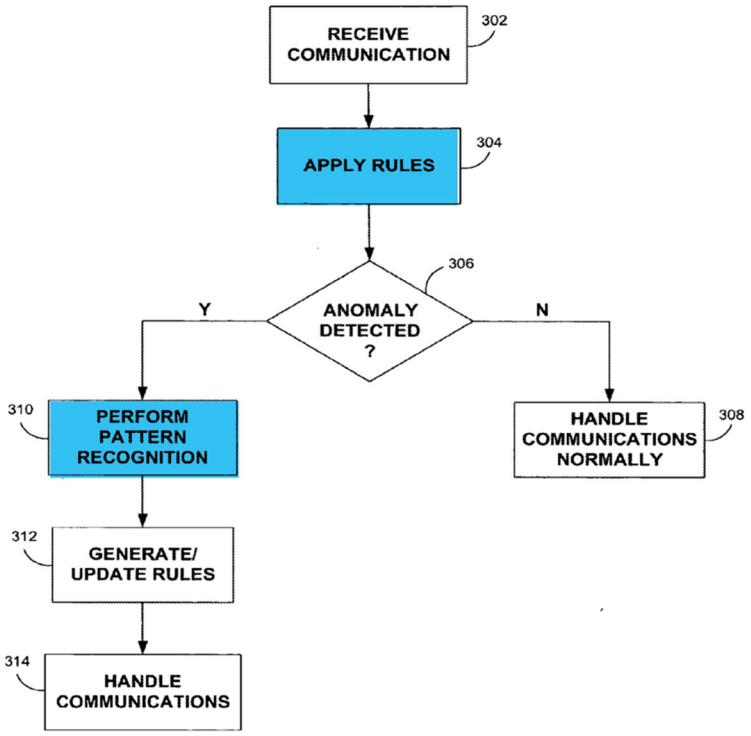


FIG. 3 (Annotated)

124. Similarly, Figure 5 (below) depicted a process flow for handling incoming communications (Ingress System (504)) of the traffic filter. (*Id.*, 17:50-52.) Steps 512-516 depicted various operations that used the content detection data (e.g., signatures and fingerprints). (*Id.*, 18:20-29.)

Netskope Exhibit 1002

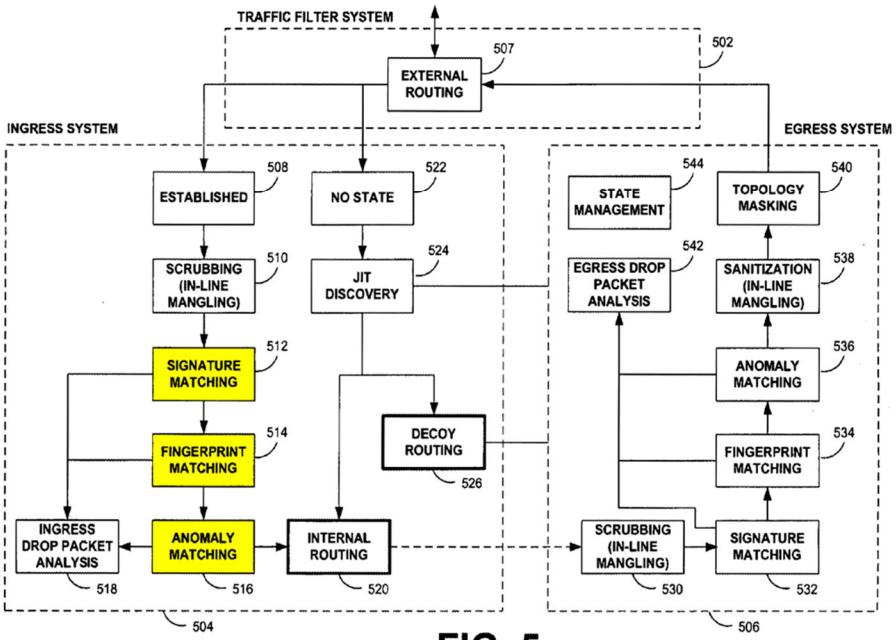


FIG. 5
(Annotated)

125. Figure 14 (below) also depicted the pattern matching (Step 1408) and fingerprint matching (Step 514) operations in the traffic filter that used the content detection data.

Netskope Exhibit 1002

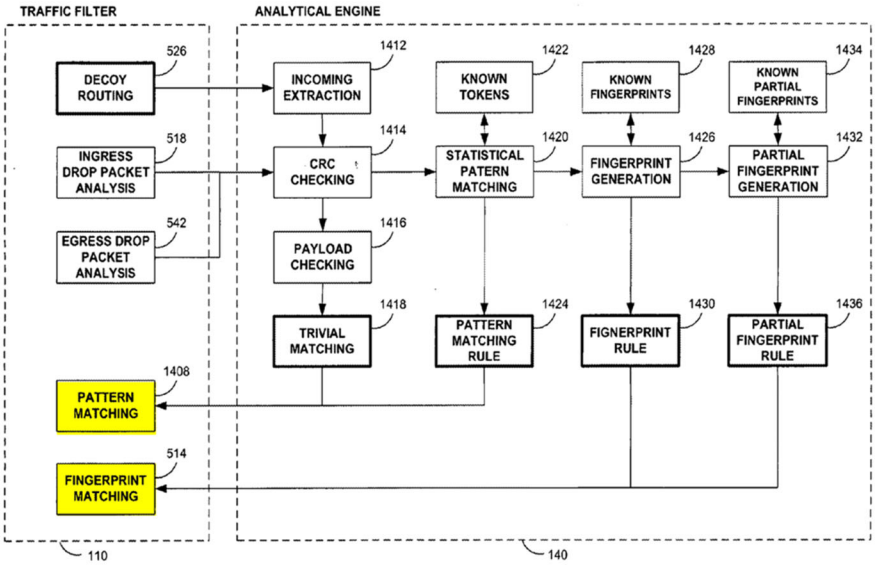


FIG. 14 (Annotated)

126. For configuration data, FortiGate disclosed processing received traffic according to firewall policies defined by such data. An administrator could configure the "Service," "Action," "NAT," "Traffic Shaping," "Authentication," and "Antivirus & Web Filter." (FortiGate, 34-35; *see also id.* 16.) The exemplary configuration below showed how the FortiGate unit received and processed traffic according to the configuration data.

Netskope Exhibit 1002

Table 4: Factory default firewall configuration

Internal Address	Internal_All	IP: 0.0.0.0 Mask: 0.0.0.0	Represents all of the IP addresses on the internal network.
External Address	External_All	IP: 0.0.0.0 Mask: 0.0.0.0	Represents all of the IP addresses on the external network.
Recurring Schedule	Always		The schedule is valid at all times. This means that the firewall policy is valid at all times.
Firewall Policy	**Internal->External**		Firewall policy for connections from the internal network to the external network.
	Source	Internal_All	The policy source address. Internal_All means that the policy accepts connections from any internal IP address.
	Destination	External_All	The policy destination address. External_All means that the policy accepts connections with a destination address to any IP address on the external network.
	Schedule	Always	The policy schedule. Always means that the policy is valid at any time.
	Service	ANY	The policy service. ANY means that this policy processes connections for all services.
	Action	ACCEPT	The policy action. ACCEPT means that the policy allows connections.
	☑ **NAT**		NAT is selected for the NAT/Route mode default policy so that the policy applies network address translation to the traffic processed by the policy. NAT is not available for Transparent mode policies.
	☐ **Traffic Shaping**		Traffic shaping is not selected. The policy does not apply traffic shaping to the traffic controlled by the policy. You can select this option to control the maximum or minimum amount of bandwidth available to traffic processed by the policy.
	☐ **Authentication**		Authentication is not selected. Users do not have to authenticate with the firewall before connecting to their destination address. You can configure user groups and select this option to require users to authenticate with the firewall before they can connect through the firewall.
	☐ **Antivirus & Web Filter**		Antivirus & Web Filter is not selected. This policy does not include a content profile that applies antivirus protection, web content filtering, or email filtering to content traffic processed by this policy. You can select this option and select a content profile to apply different levels of content protection to traffic processed by this policy.
	☐ **Log Traffic**		Log Traffic is not selected. This policy does not record messages to the traffic log for the traffic processed by this policy. You can configure FortiGate logging and select Log Traffic to record all connections through the firewall that are accepted by this policy.

(Annotated)

127. "Traffic Shaping," for example, "control[led] the maximum or minimum amount of bandwidth available to traffic processed by the policy."

128. As described above for element 1[d], a POSITA would find it obvious the content detection data and configuration data were stored on a memory device in the traffic filter.

Netskope Exhibit 1002

1[e][iii]/4[b][ii]/7[b][ii]: "to enforce policies defined at least in part by the content detection and configuration data and including at least one policy that defines a suspicious category of network traffic,"

129. Jordan disclosed that the traffic filter (110) (network gateway) enforced policies defined at least in part by the content detection (e.g., virus lists), and FortiGate disclosed enforcing policies defined by configuration data (e.g., Traffic Shaping, Authentication). Jordan also disclosed that the traffic filter (110) included at least one policy that defined a suspicious category of network traffic (e.g., communications with uncommon attributes).

130. ***"to enforce policies defined at least in part by the content detection and configuration data"*:** As discussed for element 1[e][ii], Jordan disclosed that the traffic filter (110) applied rules (enforce policies) for controlling network traffic. (*See* Element 1[e][ii]; *see also, e.g.*, Jordan, 17:39-41 ("[T]he ***traffic filter use[d] other rules***, such as predetermined rules (e.g., virus lists), or other preexisting rules.") (emphasis added); *see also id.*, 8:52-54, 9:35-38, 14:9-11, 14:30-33, 31:10-11.) Figure 3 (below) depicted this.

Netskope Exhibit 1002

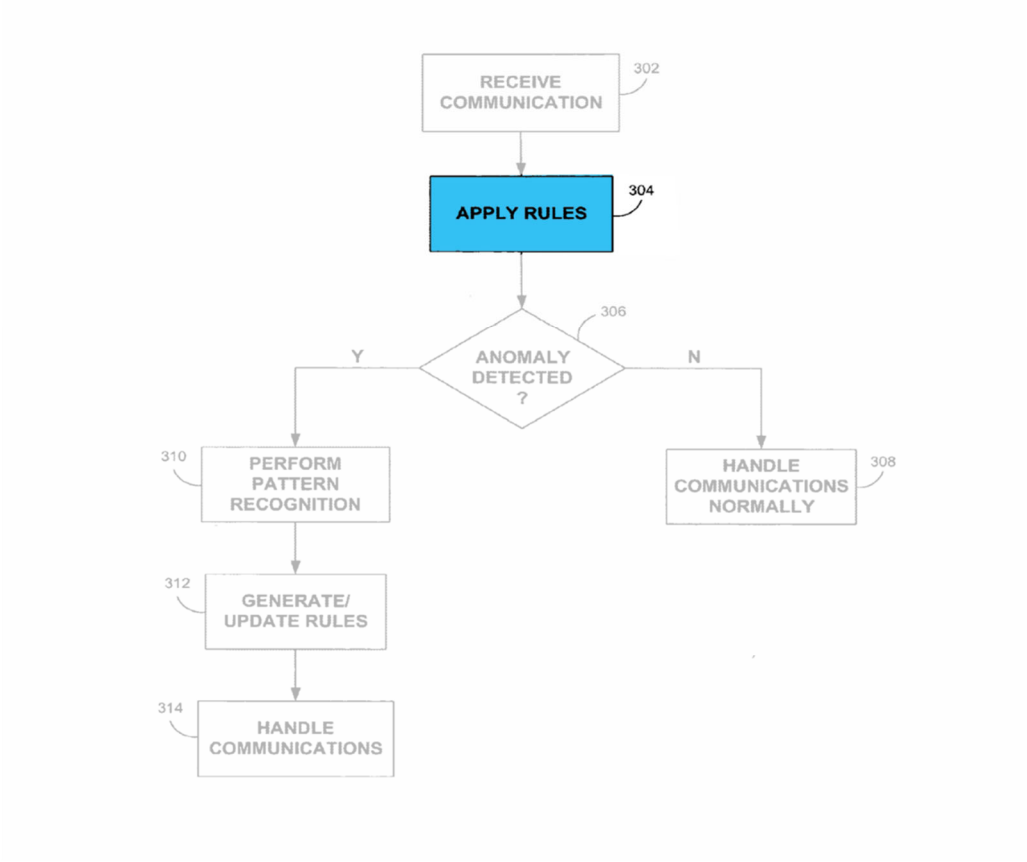


FIG. 3 (Annotated)

131. A described above in element 1[e][ii], the content detection data (e.g., virus lists, pattern matching) and configuration data (e.g., Traffic Shaping, Authentication) were defined in rules/policies. For example, the traffic filter (110) applied rules defined by content detection data because "traffic filter 110 [could] apply [rules predetermined by analytical engine 140] ***to recognize a pattern within the received communications***." (Jordan, 14:31-33 (emphasis added).) Content detection data also included virus lists. (*See, e.g.*, *id.*, 16:42-44 ("[S]ystems [could] perform malicious detection by relying on continuously updated lists of patterns (e.g., virus lists, threat lists, etc.)"), 17:39-41.)

Netskope Exhibit 1002

132. ***including at least one policy that defines a suspicious category of network traffic***: Jordan disclosed a least one rule (policy) that defined an anomalous communication. For context, Jordan explained that "anomalous communications" were "communications having attributes that [were] not common from the perspective of the traffic filter or the network of the traffic filter." (Jordan, 5:41-44.) Jordan also disclosed that "[m]alicious communications [could] also often be considered anomalous communications…." (*Id.*, 5:55-56.)

133. These categories of network traffic were defined in rules. (Jordan, 3:28-30 ("determine if the received network communication ***include[d] an anomaly according to a predetermined rule set***.…") (emphasis added); *see also, e.g.*, *id.*, 6:40-43 ("when an anomalous communication [was] received, the analytical engine [could] be configured to ***generate rules that prevent[ed] additional, similar anomalous communications*** from being received.") (emphasis added), 10:21-23 ("If the traffic filter 110 deem[ed] a communication anomalous and the ***structure of that anomalous communication forms the basis of a filtering rule***….") (emphasis added).) Anomalous communications could be ones that requested a service that did not exist or were unrecognizable by the traffic filter 110. (*See* Jordan, 15:10-28.) A rule could also define as "anomalous" a user logging in both remotely and from their desktop computer. (*Id.*, 17:5-8.)

Netskope Exhibit 1002

1[e][iv]/4[b][iii]/7[b][iii]: "the policy enforcement performed to determine whether to allow the received network traffic to pass [on a data network]5,"

134. Jordan disclosed that the traffic filter applied rules to determine whether to allow network traffic to pass.

135. Jordan disclosed that "traffic filter 110 [] ***implement[ed] even recent rules*** generated based on recently received communications…and consequently the ***traffic filter 110 [] prevent[ed] such malicious payloads from being delivered*** to one or more devices 120 within the network." (Jordan, 9:40-45 (emphasis added); *see also, e.g.*, *id.*, 21:33-37 ("malicious communications can be used to generate and/or ***update a filtering rule that can be used by the traffic filter 110*** to improve the filtering capability of the network system 100 generally, and ***to prevent similar attacks in the future***") (emphasis added), 6:40-44.) For example, if signature matching (512), fingerprint matching (514), or anomaly matching (516) failed, an analysis was performed to "determine if at least a portion of the incoming communication…***should be dropped,***" as shown by Step 518. (*Id.*, 18:35-38 (emphasis added).)

[5] "On a data network" applies to only claims 4 and 7.

Netskope Exhibit 1002

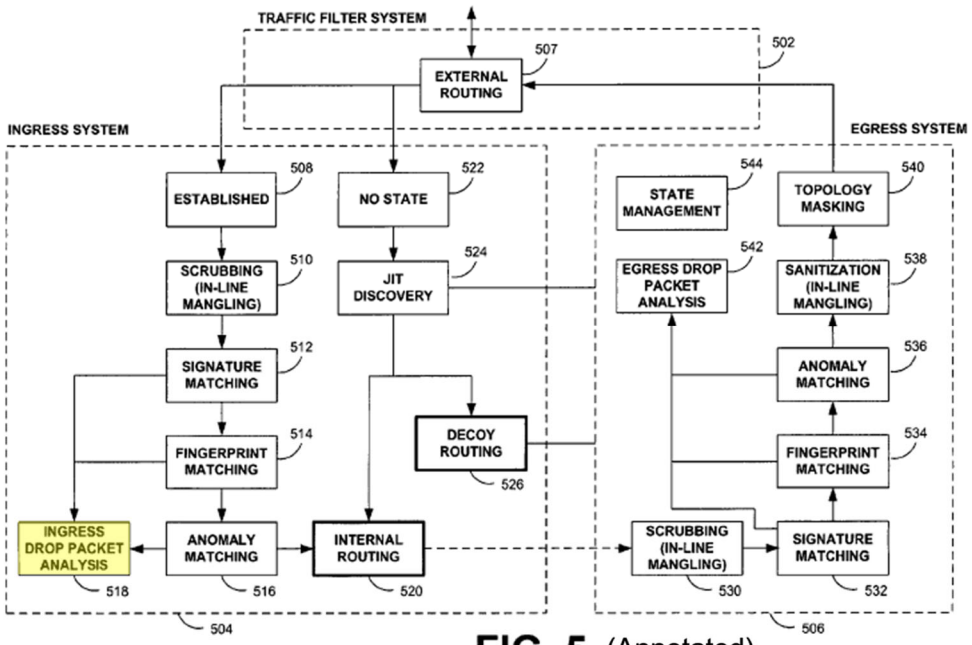


FIG. 5 (Annotated)

136. Additionally, Jordan disclosed that the traffic filter (110) determined whether to pass the received network traffic onto network system (100) (data network). (Jordan, 9:43-45 ("traffic filter 110 can prevent such malicious payloads from being delivered to one or more devices 120 within the network.")). Figure 1, for example, showed traffic filter 110 as the gateway between the external network and network system (100). (*See also, e.g.*, *id.*, 5:28-32.)

Netskope Exhibit 1002

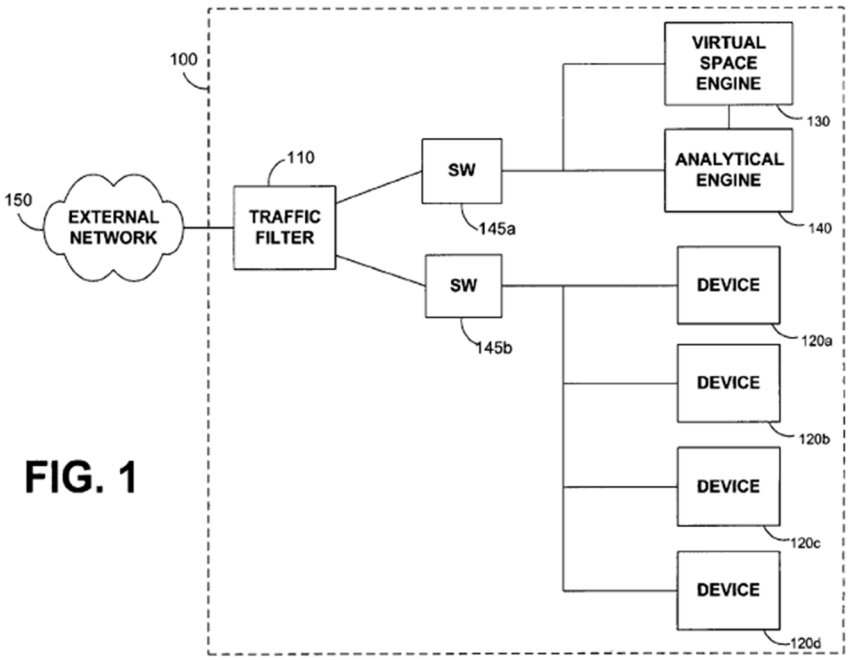


FIG. 1

1[e][v]/4[b][iv]/7[b][iv]: "the policy enforcement including user identification, content identification, and at least one of source verification and destination verification,"

137. Jordan disclosed or at least suggested user identification, content identification, source verification, and destination verification. To the extent Jordan did not disclose one or more of these, FortiGate did.

138. ***user identification***: Jordan disclosed policy enforcement that included user identification. For example, Jordan described a rule that prohibited a user from logging in both remotely and at a desktop simultaneously. (Jordan, 17:5-17:9.) This rule would, of course, be based on the user's identification.

139. If Jordan did not disclose user identification, FortiGate did.

140. As one example, FortiGate described filtering email based on the sender appearing on an Email block list. (FortiGate, 15.) A POSITA would understand that email lists could include user identification (e.g., usernames).

141. As another example, FortiGate disclosed a Cerberian web filter option, which made traffic decisions based on user identification. (FortiGate, 274.) This option "[could] only be applied to user groups." (*Id.*)

142. ***content identification***: Jordan disclosed that rule enforcement by the traffic filter (110) included an intrusion detection system module that checked the contents of data streams (e.g., traffic). (Jordan, 25:61-26:3 ("A main module is provided, and includes multiple sub-modules, such as a state management, intrusion detection systems (IDS)…[and t]he IDS/IPS module can also include pattern and ***content checking modules, which can search certain patterns and/or contents, respectively, of a communication data stream***.") (emphasis added).) Similarly, the '968 Patent described that content scanning included "detecting undesirable content, such as … [a] porno website, etc." ('968 Patent, 5:23-31.) A POSITA would have found it obvious that the content searching disclosed in Jordan would include searching for malicious or inappropriate content because content filters commonly checked for such things.

143. Further, FortiGate disclosed scanning email contents to determine if they contained a banned word or phrase. (FortiGate, 15 ("[i]f an ***email is found to***

Netskope Exhibit 1002

contain a word or phrase in the banned word list, the FortiGate adds an Email tag to subject line of the email. Receivers can then use their mail client software to filter messages based on the Email tag.") (emphasis added), 281.) FortiGate disclosed a similar feature for pornography. (*Id.*, 277.)

144. ***source and destination verification***: Jordan disclosed at least source verification because it described rules "***regarding … receipt of communications from certain network sources*** …." (Jordan, 6:51-55 (emphasis added).) Likewise, a POSITA would have it obvious that Jordan's rules could include destination verification since that source and destination verification were commonly used together.

145. FortiGate also disclosed source and destination verification. Specifically, firewall policies could specify a source and destination address, as shown below. (FortiGate, 23.) These addresses identified source IP addresses on the internal network (highlighted in yellow) and destination IP addresses on an external network (highlighted in purple); for example, if a source address could be verified to be on the list of internal addresses, connections from that source were allowed. The below table also shows a similar verification for destination addresses.

Netskope Exhibit 1002

Table 4: Factory default firewall configuration (Annotated)

Internal Address	Internal_All	IP: 0.0.0.0 Mask: 0.0.0.0	Represents all of the IP addresses on the internal network.
External Address	External_All	IP: 0.0.0.0 Mask: 0.0.0.0	Represents all of the IP addresses on the external network.
Recurring Schedule	Always		The schedule is valid at all times. This means that the firewall policy is valid at all times.
Firewall Policy	**Internal->External**		Firewall policy for connections from the internal network to the external network.
	Source	Internal_All	The policy source address. Internal_All means that the policy accepts connections from any internal IP address.
	Destination	External_All	The policy destination address. External_All means that the policy accepts connections with a destination address to any IP address on the external network.
	Schedule	Always	The policy schedule. Always means that the policy is valid at any time.
	Service	ANY	The policy service. ANY means that this policy processes connections for all services.
	Action	ACCEPT	The policy action. ACCEPT means that the policy allows connections.
	☑ **NAT**		NAT is selected for the NAT/Route mode default policy so that the policy applies network address translation to the traffic processed by the policy. NAT is not available for Transparent mode policies.
	☐ **Traffic Shaping**		Traffic shaping is not selected. The policy does not apply traffic shaping to the traffic controlled by the policy. You can select this option to control the maximum or minimum amount of bandwidth available to traffic processed by the policy.

1[e][vi]/4[c][i]/7[c][i]: "[wherein] when network traffic is determined to violate the at least one policy defining suspicious network traffic, forwarding the network traffic to an analysis process that will perform analysis on network traffic to determine whether the network traffic contains a threat desired to be detected and,"

146. Jordan disclosed that the traffic filter determined if traffic violated a policy defining anomalous traffic and, if so, it forwarded the traffic to an analytical

Netskope Exhibit 1002

engine (140) (analysis process) that analyzed the traffic to determine if it needed to generate or update rules.

147. As discussed for element [1[e][iii], Jordan disclosed a policy that defined anomalous communications, such as a communication that requested a service not offered by any of the devices on the network. (Jordan, 6:16-18; *see also id.*, 9:7-11.) If a communication violated the policy, the traffic filter routed the communication to analytical engine (140). (Jordan, 15:19-24 ("if a received communication ***violate[d] a predefined rule*** (e.g., a rule defined previously by the analytical engine 140, etc.), the ***traffic filter 110 can consider the communication to be anomalous***. Such anomalous communications can be ***handled by*** the virtual space engine 130 and/or the ***analytical engine 140***."), 5:58-63 ("[I]f the traffic filter determine[d] that a payload of a network communication contain[ed] malicious data (e.g., a virus, worm, etc.), then the ***anomalous network communication can be routed to the analytical engine by the traffic filter*** for determination of rules for handling such communications…."), 9:51-53, 11:6-9, 14:16-48.) This is illustrated in Figure 1 (below).

Netskope Exhibit 1002

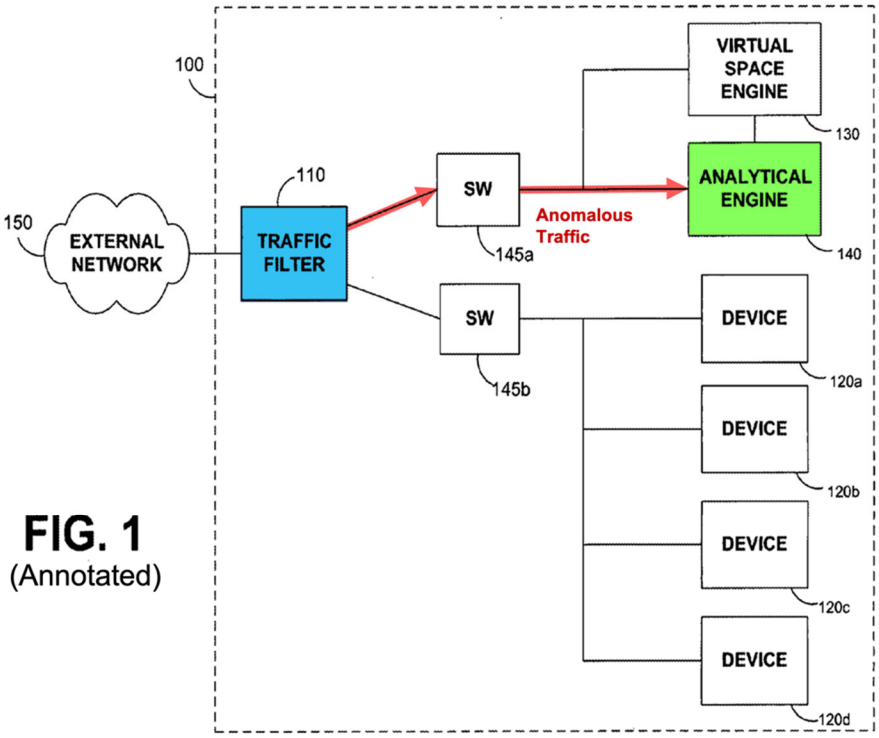


FIG. 1
(Annotated)

148. Jordan also showed this process in connection with Figure 3 (annotated below). Specifically, if anomalous communications was detected (Step 306), pattern recognition was performed by the traffic filter or analytical engine depending on if rules existed. (Jordan, 14:27-30.)

149. If no rules existed for the communication, a copy of the communication could be sent to the analytical engine 140. (*Id.*, 14:33-36.) The analytical engine then analyzed the communication for malware and performed pattern recognition on it. (*Id.*, 14:37-44; *see also id.*, 5:58-6:2, 6:16-28, 6:37-46, 6:47-55, 9:35-42, 9:51-64, 16:19-23.)

Netskope Exhibit 1002

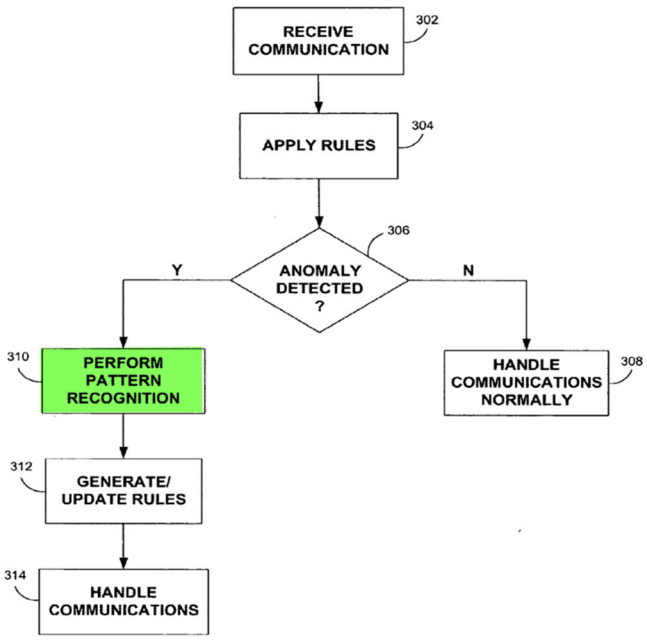


FIG. 3 (Annotated)

1[e][vii]/4[c][ii]/7[c][ii]: "when the network traffic is determined to contain a threat desired to be detected, the analysis process generates additional content detection data to detect the threat in subsequently received network traffic;"

150. Jordan disclosed that the analytical engine (analysis process) generated additional content detection data to detect the anomalous data in subsequent traffic.

151. As described in element 1[e][vi], Jordan disclosed that the analytical engine (140) generated or updated rules if the anomalous communication (e.g., traffic) was malicious. (Jordan, 6:37-44 (stating the analytical engine "analyze[d] a received communication, and determine[d] if rules should be generated or updated

[for use by the traffic filter (110)] to prevent any harm from the received communication[,] … [such as] generat[ing] rules that prevent additional, similar anomalous communications from being received by a network associated with the anomalous communication."); *see also id.*, 5:58-63 ("[I]f the traffic filter determines that a payload of a network communication contains malicious data (e.g., a virus, worm, etc.), then the anomalous network communication can be routed to the analytical engine … for determination of rules for handling such communications.").)

152. After analyzing the received anomalous communication, the analytical engine (140) "generate[d] rules that prevent additional, similar anomalous communications from being received by a network associated with the anomalous communication." (Jordan, 6:42-44.) This act of updating and generating new rules is seen in Figure 3 (below).

Netskope Exhibit 1002

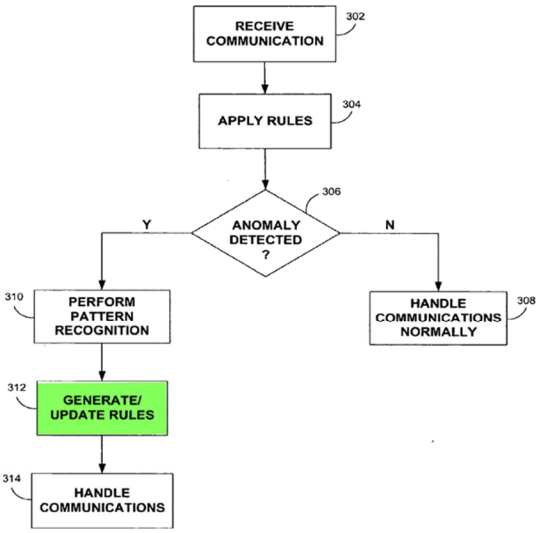


FIG. 3 (Annotated)

153. The generated/updated rules included virus lists (e.g., content detection data). Jordan disclosed that the analytical engine's analysis included pattern recognition. (Jordan, 14:36-37.) After that analysis, "the analytical engine 140 [] then generate[d] and/or update[d] rules…based on the pattern recognition performed." (Jordan, 14:51-53.)

1[f]/4[d]/7[d]: "receive, via the [at least one] network interface device, the additional content detection data indirectly from the analysis process via an update station;"

154. Jordan disclosed that the traffic filter received the new/updated rules via a switch. If Jordan's switch does not constitute an "update station," a POSITA would have found it obvious to incorporate FortiGate's update station into Jordan.

155. Figure 14 (below) illustrated the analytical engine sending new/updated rules to the traffic filter.

Netskope Exhibit 1002

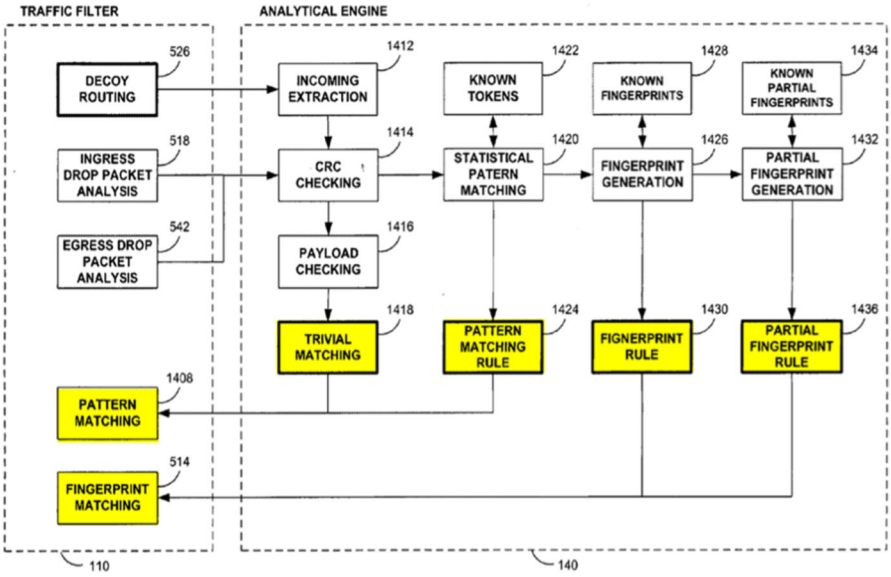


FIG. 14 (Annotated)

The rules contained additional content detection data (e.g., virus lists). (Jordan, 9:38-40, 14:54-56.)

156. The additional content detection data went indirectly from the analytical engine to the traffic filter. As shown in Figure 1 (below) showed that a switch (145a) intermediated traffic between analytical engine (140) and the traffic filter (110). Accordingly, the traffic filter received the data indirectly because it went through the switch.

Netskope Exhibit 1002

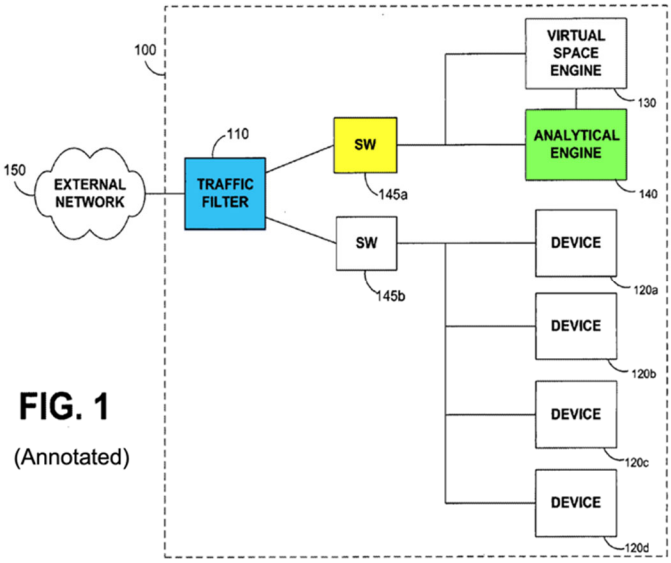


FIG. 1

(Annotated)

157. If Jordan's traffic filter is not an "update station," a POSITA would have found it obvious to incorporate that feature from FortiGate. FortiGate disclosed a FortiResponse Distribution Center (FDC) that received threats from Fortinet customer's and updated antivirus definitions. (FortiGate, 335.) The FDC then pushed updates to update stations called FortiResponse Distribution Servers (FDSs). (FortiGate, 336.) The FDSs (update stations) pushed updates to FortiGate firewalls. (*Id.*) Likewise, the '968 Patent describes update stations, such as a computer or server, that receives content detection data and send it to content detection modules. ('968, 4:53-5:13.) A POSITA would have been motivated to incorporate FortiGate's update stations into Jordan to improve Jordan's scalability. As was common in networks, a network like Jordan's would have multiple traffic filters/gateways to guard various ingress/egress points. In that case, whenever the analytical engine

Netskope Exhibit 1002

generated additional content detection data, it could send the information to the update station, which could coordinate pushing the update to the various traffic filters/gateways.

1[g]/4[e]/7[e]: "store[-ing] the additional content detection data on the at least one memory device"

158. Jordan did not explicitly disclose that the additional content detection data was stored on the traffic filter's memory device. However, a POSITA would find it obvious that it was.

159. The traffic filter would need to retrieve and use it, and retain it if the unit is turned off.

1[h]/4[f]/7[f]: "process[ing] subsequently received network traffic in view of the additional content detection data;"

160. Jordan disclosed that the traffic filter applied new/updated rules to future traffic using, for example, updated signatures.

161. Jordan disclosed that the traffic filter (110) used the new/updated rules for improved anomaly detection against future traffic. (Jordan, 14:49-56; *see also id.*, 16:24-29 ("[T]he analytical engine 140 can generate or update rules in optional operation 418. These rules can be communicated to the traffic filter 110 for use by the traffic filter in filtering future traffic received from devices external to the network system 100 (e.g., via the external network 150).".)

Netskope Exhibit 1002

162. Jordan disclosed that the updated/new rules comprised additional content detection data. For example, Jordan disclosed that new rules were generated based on pattern recognition. (Jordan, 14:49-53.) As described above in element 1[d], ***virus lists*** (e.g., content detect data) were generated by the analytical engine (140) by performing pattern recognition. (Jordan, 16:37-44.)

1[i]/4[g]/7[g]: "block[ing] network traffic determined to violate at least one policy; and"

163. Jordan disclosed that the traffic filter blocked network traffic that violated at least one policy (e.g., a set of rules). (Jordan, 9:40-45 ("[T]he traffic filter 110 can implement even recent rules generated based on recently received communications (e.g., communications containing malicious payloads), and consequently the traffic filter 110 can prevent such malicious payloads from being delivered to one or more devices 120 within the network….")).

1[j]/4[h]/7[h]: "allow[ing] network traffic to pass that does not violate a policy."

164. Jordan disclosed that the traffic filter (110) allowed network traffic to pass that did not violate a policy (e.g., a set of rules). (Jordan, 5:33-36 ("[T]raffic filter can route ***normal communications (i.e., non-anomalous communications)*** to a device within the network to which they are addressed, or by which they should normally be handled.") (emphasis added); *see also id.*, 8:58-9:6.)

Netskope Exhibit 1002

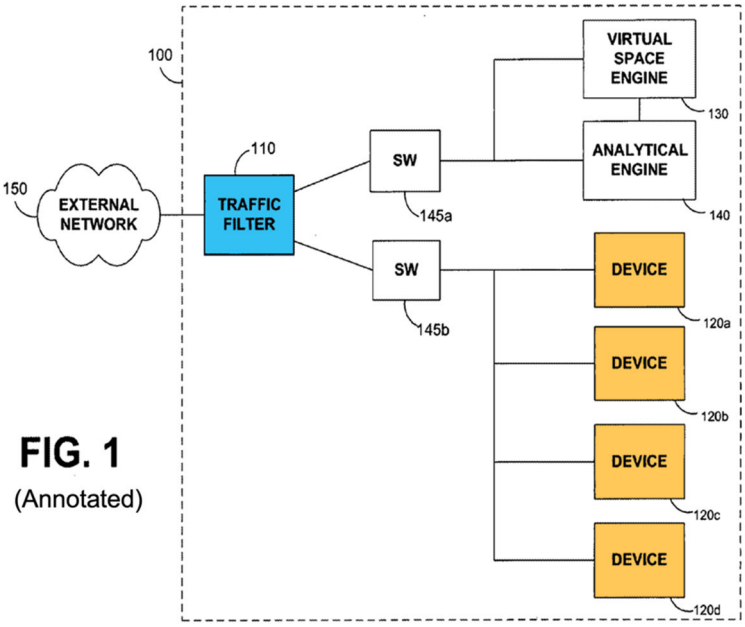


FIG. 1
(Annotated)

165. It would be obvious that the new rules would allow normal traffic to pass through the traffic filter (110).

3. Claims 2, 5, and 8: "…wherein the analysis process, to which the network traffic content is forwarded, executes on a computing device distinct from the network gateway device[/ data network device]."

166. Jordan disclosed that the traffic filter was distinct from the analytical engine, which processed the received network communications. Specifically, Jordan disclosed that the traffic filter (110) and analytical engine (the device executing the analysis process) were separated by a switch. (Jordan, 10:31-40). Further, Figure 1 (annotated below) depicted traffic filter (110) and analytical engine (140) in separate boxes, meaning they are distinct devices.

Netskope Exhibit 1002

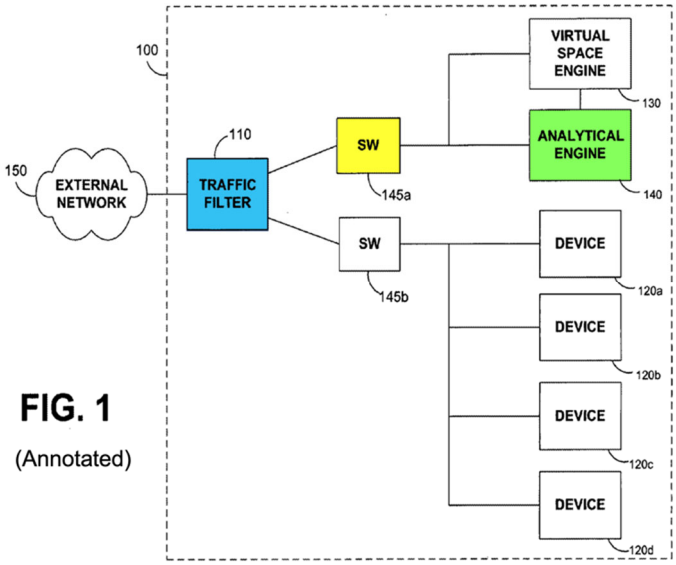


FIG. 1

(Annotated)

4. Claims 3, 6, and 9: "…wherein the analysis process facilitates an analysis that includes receiving administrator input."

167. Jordan disclosed that the analytical engine could be accessed via a user interface (highlighted in red). The user interface allowed a user to access functions of the analytical engine, such as the rule generation capability (highlighted in green). (Jordan, 24:64-25:5 ("The UI 1002 can access functions of both the traffic filter 110, and the analytical engine 140. For example, the UI can access the rule generation capability of the analytical engine 140…."); *see also* Figure 3 (below).)

Netskope Exhibit 1002

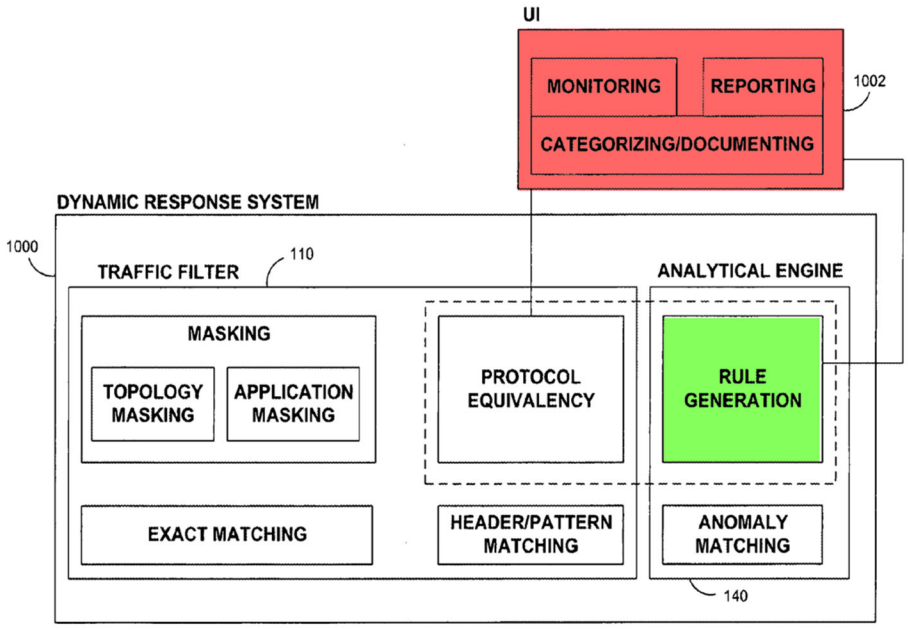


FIG. 10 (Annotated)

168. Because a user could access functions of the analytical engine, such as the rule generation capability, a POSITA would find it obvious that the analysis process permitted receiving administrator input.

169. As further evidence of obvious, Jordan disclosed a "management module" that allowed a system administrator using a user interface to set rules for the traffic filter. (Jordan, 25:50-54 ("A management module [was] provided as an interface between traffic filter 110 and a system administrator (e.g., accessing the system via a user interface, etc.), and can allow a system administrator to manage rules of the traffic filter 110, set system parameters, and/or read system logs."); *see also* Jordan, 25:28-49.)

Netskope Exhibit 1002

170. Having an administrator analyze suspect traffic was nothing new. For example, U.S. Pat. No. 7,080,408 ("Pak") (Ex. 1019), was directed to "Network Communications Having Suspicious Contents." (Pak, Title.) Pak discussed that when analyzing suspicious traffic, an administrator could be involved. Pak, 6:25-28 ("Another pattern includes operation 320, in which the suspect content is sent to an expert for evaluation before being released from quarantine. The expert can be akin to the scientist described above."; *see also id.*, 6:28-35.) Accordingly, receiving administrator input to analyze the suspicious traffic was common before the time of the alleged invention.

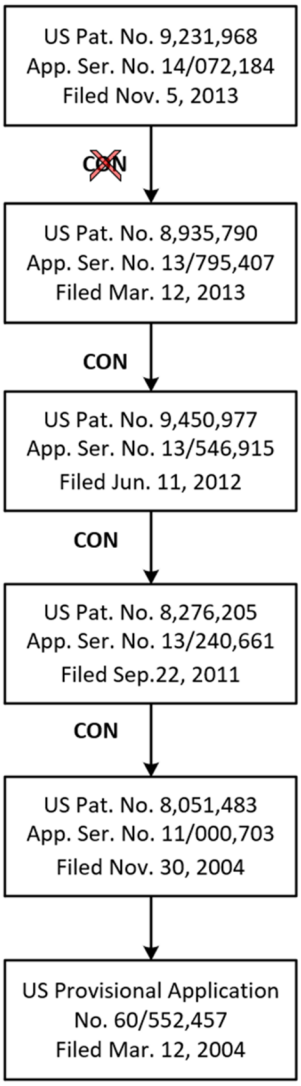
5. Jordan Is Prior Art

171. Jordan is prior art for three independent reasons.

172. First, as discussed above (*see supra*, Section II.C.2), the challenged claims are entitled to a priority date of no earlier than November 5, 2013 because they lack written description support in any parent application. Jordan's priority date is no later than November 17, 2004, the date on which its application was filed.

173. Second, setting aside those written description issues, other limitations in the challenged claims lack written description in the three-page provisional application to which they claim priority (U.S. Provisional Application 60/552,457) as explained below. Thus, the priority date of the challenged claims is no earlier than November 20, 2004.

Netskope Exhibit 1002



174. **"configuration data" and "user identification"**: Like the '407 application, the '457 provisional lacks written description support for "configuration data," "user identification" and their associated operations. For example, the '457 provisional does not mention "user identification" (or even "user," "identification," or "authentication") and discloses only processing received traffic in view of security update data. (*See* '457 provisional at § 3.1 ("There are two kinds of update data: security update data and subscriber data. Security update data refers to the data

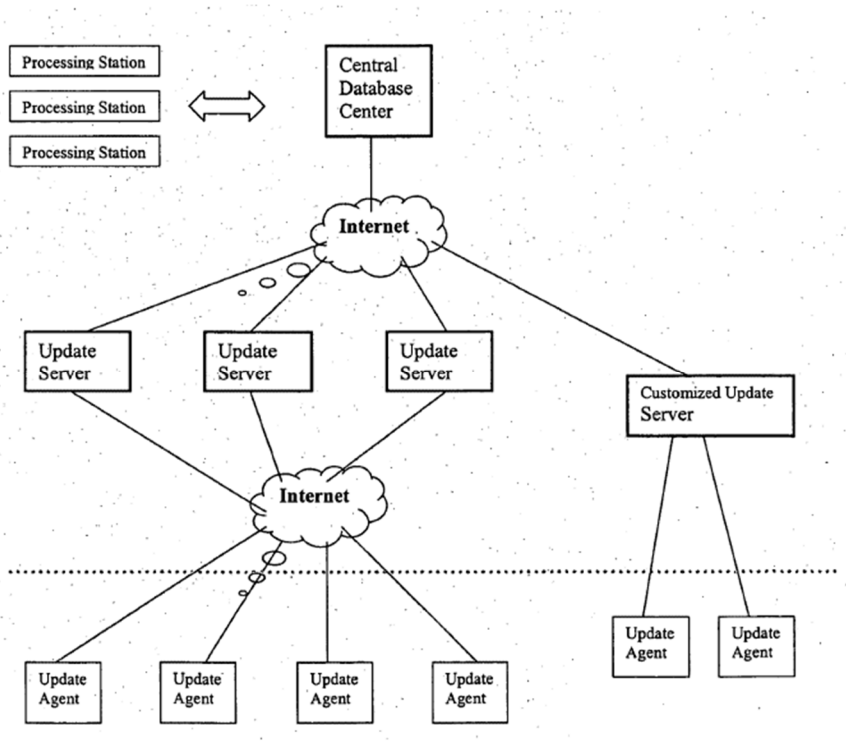
Netskope Exhibit 1002

such as virus definitions, NIDS signature, etc. They will be used by an Update Agent to check its pass-through traffic to ensure its protected network or host to be compromised by unsafe content.").)

175. **Limitations 1[e][vi]/4[c][i]/7[c][i]**: The '457 provisional application fails to disclose that a "network gateway device" (claim 1) or "data network device" (claims 4 and 9) "forward[] the network traffic to an analysis process."

176. The '457 provisional discloses processing stations. As shown in the below figure from the '457 provisional, the processing stations communicated with the Central Database Center. There is no communication between the processing stations and the update agents, which "inspect[ed] the traffic passing through [them] to ensure they are safe based on the latest update…." (*Id.*, § 2.) In fact, the '457 application is entirely silent as to how processing stations receive information.

Netskope Exhibit 1002



177. The lack of written description support is not surprising given that the '968 Patent offers only a single sentence that arguably support this operation. ('968 Patent, 6:3-7 ("For example, a content detection module 108 may detect a content that is suspicious (or that requires further processing), in which case, content detection module 108 then sends the electronic content to processing station 104 for processing.")). That disclosure not found anywhere in the '457 provisional. (Ex. 1006, 2.)

178. **Limitation 1[e][v]/4[b][iv]/7[b][iv]**: The challenged claims require "policy enforcement including user identification, content identification, and at least one of source verification and destination verification…" The '457 provisional disclose only that an update agent used security update data (e.g., a virus definition

Netskope Exhibit 1002

or NIDS signature) to process traffic. (*See* '457 provisional, §§2 (update agent "inspect[d] the traffic passing through it to ensure they are safe based on the latest update it gets from an Update Server"), 3.1 (disclosing "two kinds of update data: security update data and subscriber data…").) It discloses nothing about policy enforcement including user identification, content identification, source verification, or destination verification.

179. Finally, Jordan properly claims priority to provisional application 60/520,577, filed on November 17, 2003, because at least claim 1 has written description support in the '577 provisional, as shown in the table below. Thus, Jordan's priority date (November 17, 2003) predates even the earliest priority date of the '968 Patent (March 12, 2004).

Jordan Claim 1	**'577 Provisional Support**
A method implemented by a first device, the method comprising:	Fig. 3; ¶17
receiving a first network communication from a second device, the first network communication requesting a service;	Fig. 3
responding to the second device by mangling a communication received from within a network, in response to a determination that the service is available on the network;	Fig. 3 (top half); ¶¶45, 47
modifying a payload of a packet to remove traits of the network and to provide apparent traits;	¶¶47-48
responding to the second device that the service is available and mimicking	Fig. 3 (bottom half); ¶¶47, 62

Netskope Exhibit 1002

protocols for the service, in response to a determination that the service is not available on the network;	
receiving a second network communication from the second device; and	Fig. 3 (bottom half (session ACK/PUSH))
identifying whether existing rules are present for performing a pattern recognition for the second network communication, wherein if the existing rules are not present at the first device, at least a portion of the second network communication is copied to generate new rules for pattern recognition activities.	¶¶16, 20, 32, 60, 61, 64

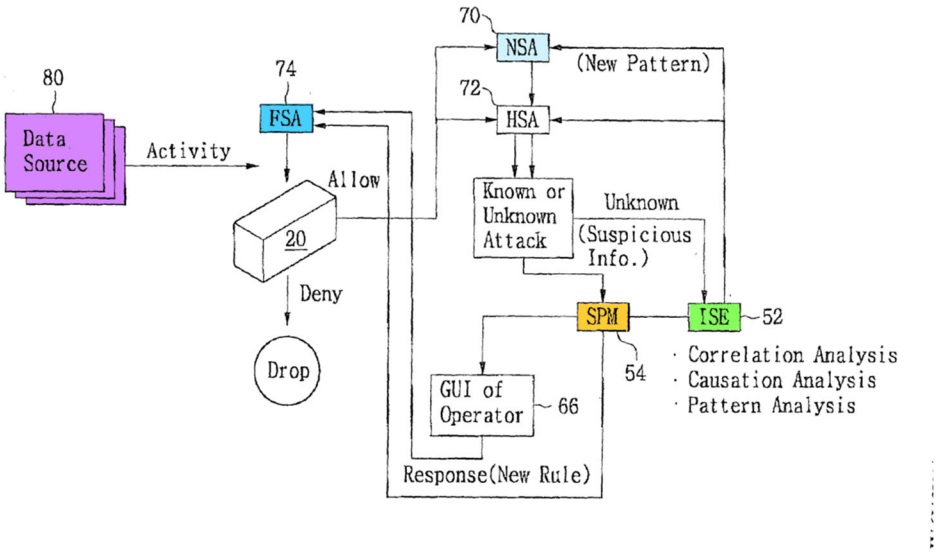
C. Ground 2: Combining Lee and Magdych renders claims 1-9 obvious.

180. ***Lee:*** Lee disclosed "an intelligent and integrated security system which [could] ***precisely detect intrusions and take real-time measures in response to the detected intrusions***." (Lee, ¶10 (emphasis added).)

181. Lee's Figure 2 (annotated below) included a firewall security agent (FSA (74)) that caused a firewall to block traffic from an attack data source (80)), a network security agent (NSA (70)) that analyzed suspicious traffic, and an intelligent security engine (ISE (52)) that analyzed traffic sent from the other agents (e.g., FSA and NSA) to decide if an attack was occurring and generated a pattern to prevent future attacks. (Lee, ¶¶51-53; *see also id.*, ¶¶13, 14 & 34 (referencing similar functions illustrated in Fig. 1.))

Netskope Exhibit 1002

FIG 2 (Annotated)



182. How the security agents functioned and interacted was shown in Figure 2 (annotated above). The FSA (74) blocked traffic from an attack data source (80). (*Id.*, ¶51.) The NSA (70) detected known attacks, suspicious information, and traffic, and generated a report to the ISE (52). (*Id.*) The ISE (52) analyzed the attack and if a new attack was detected, the new signature was transferred to the NSA (70) and a new or modified rule was provided to the FSA (74) through the SPM (54). (*Id.*, ¶52.)

183. ***Magdych:*** Magdych disclosed a secure gateway for "detecting attacks on a network." (Magdych, Title, Abstract.)

184. Magdych disclosed a firewall device that had a firewall portion (402) and an intrusion detection system portion (404). (*Id.*, 4:17-26.)

Netskope Exhibit 1002

185. The firewall device (210) intermediated traffic between a remote traffic source (202) and target (206) to detect and address threats. (*Id.*, 4:17-26.) This is illustrated in Figure 4 of Magdych (below).

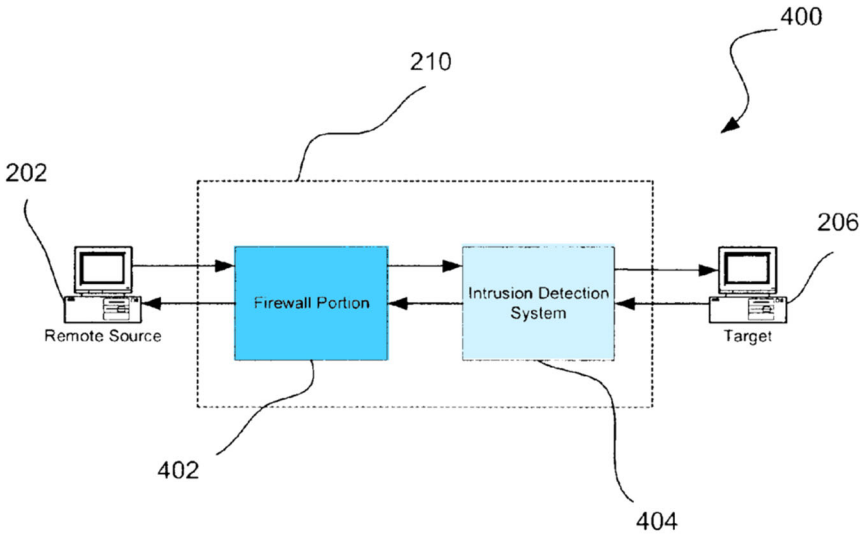


Figure 4 (Annotated)

1. Motivation to combine

186. A POSITA would have been motivated to combine Lee and Magdych with a reasonable expectation of success.

187. First, both Lee and Magdych concerned network and data security; particularly, protecting networks from threats by analyzing network traffic for potential malware. (Lee, ¶10 ("Another object of this invention is to provide an intelligent and integrated security system which can precisely detect intrusions and

Netskope Exhibit 1002

take real-time measures in response to the detected intrusions.")), Abstract ("The ISE performs a correlation analysis and a causation analysis on suspicious traffic…."); Magdych, Abstract ("A system, method and computer program product are provided for detecting attacks on a network. … The data representing text that are marked as hostile are then acted upon in order to prevent an attack.".) Thus, a POSITA looking to improve Lee would have looked to complementary references like Magdych.

188. Second, combining Magdych and Lee provided Lee with improved security, flexibility, efficiency, and control. As discussed above, Lee disclosed a system with separate agents performing separate functions. For example, firewall agent (FSA 74) "applie[d] a security policy to the firewall", which could include making "the firewall 20 block traffic from the attack data source 80" (Lee, ¶¶46, 51), while network security agent (NSA 70) "detect[ed] known attacks, suspicious information and traffic…." and received a "generated signature … so that more rapid confrontation in response to future attacks of the same pattern is made possible" (*Id.*, ¶¶51-52).

189. Magdych recognized that existing firewalls could not filter traffic based on content, so they were less effective at screening malicious content. (Magdych, 2:29-36.) Thus, Magdych explained that there was a need to integrate intrusion detection capabilities, like the ones performed by the NSA, into firewalls. (*Id.*)

Netskope Exhibit 1002

190. With Magdych's disclosures in mind, a POSITA would have been motivated to combine Lee's NSA and FSA functions together into a single firewall unit. First, it would save rack space and combining software into a single module would have been feasible given increased processor speeds and memory. Second, it would have improved Lee's efficiency by having a single device perform at least two levels of analysis (firewall-level and content-level analyses). It would have also improved Lee's security. Currently, Lee's architecture has the NSA behind the FSA—i.e., deeper within the network. But it was standard practice to deploy traffic filters at a network's perimeter to minimize any potential damage in the event of an attack. This is reflected by Lee's use of a DMZ zone, although a POSITA would have found it obvious that the safety of even the DMZ could be improved by adding content detection before traffic even entered the DMZ. Finally, Lee explained that security agents (such as the NSA or FSA) could be readily added or removed. (Lee, ¶ 40.)

191. Thus, a POSITA would be motivated to follow the teachings of Magdych and incorporate the NSA and FSA software modules into a single device that could better monitor network traffic and provide better security.

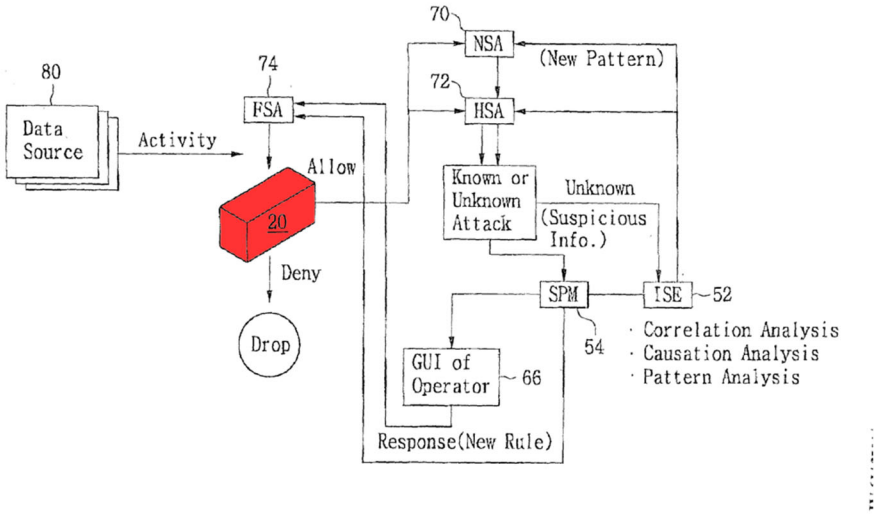
2. Claims 1, 4, and 7

1[pre]:

Netskope Exhibit 1002

192. Lee disclosed a network gateway that included firewall (20). (Lee, ¶34 ("The firewall 20 [was] a set of associated programs located in a ***network gateway*** server and protect[ed] resources of the internal network from outside users.") (emphasis added).) Lee depicted firewall (20) in Figure 2 (below).

FIG 2 (Annotated)

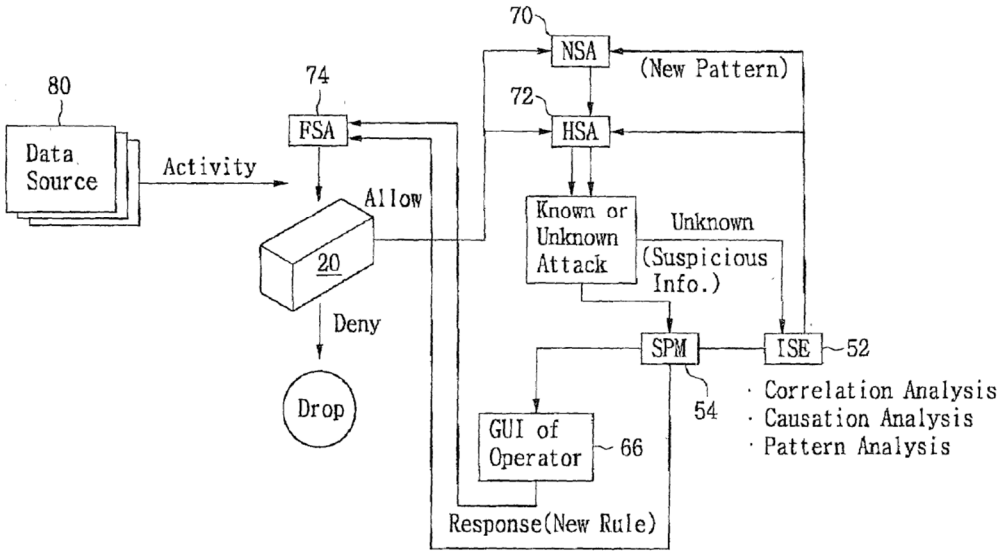


4[pre]:

193. Lee disclosed "intelligent and integrated security system which [could] precisely detect intrusions and take real-time measures in response to the detected intrusions." (Lee, ¶10.) Figure 2 (below) depicted a process flow (method) for detecting and responding to intrusions in real time. (*Id.*, ¶¶50-52.)

Netskope Exhibit 1002

FIG 2



7[pre]:

194. Lee suggested this element and a POSITA would have found it obvious based on the disclosures in Lee.

195. Lee disclosed that "firewall 20 [was] a set of associated programs ***located in a network gateway server***." (*Id.*, ¶34.) FSA (74) was provided within the firewall and was a software program, as were the NSA and HSA. (Lee, ¶¶38-39; *see also id.*, ¶31.) Those agents "search[ed] for characteristic patterns of data over the network" and "produce[d] and transmit[ted] a security alert message to one or both of communicating devices and the security manager." (*Id.*, ¶39.) For firewall-FSA, for example, Lee disclosed that it performed "the more ***CPU-intensive function*** of packet-by-packet inspection." (*Id.*, ¶35.) "CPU," of course, referred to a "central processing unit"—also known as a processor.

Netskope Exhibit 1002

196. A POSITA would have found it obvious that Lee disclosed the memory (non-transitory computer-readable medium) with instructions that would be accessible by the CPU. In order for the CPU to perform its functions, it would need memory with instructions.

1[a]:

197. As discussed for element 7[pre], Lee's network gateway had a processor.

1[b]:

198. As discussed for element 7[pre], a POSITA would have found it obvious that Lee's network gateway had memory.

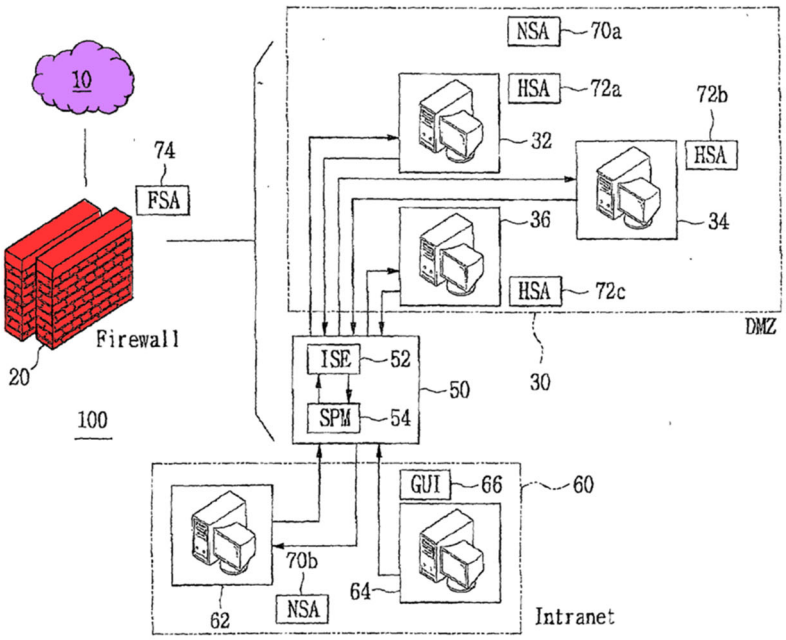
1[c]:

199. Lee disclosed that the firewall 20 had a network interface device.

200. The firewall 20 intermediated traffic between a secure network connected and the insecure outside world. (Lee, ¶¶35-36; *see also, e.g.*, *id.*, ¶7 (disclosing the "trend of integrating wired communication links and wireless telecommunication networks.").) Figure 1 (below) depicted firewall 20 receiving traffic from an outside network (e.g., the Internet). (Lee, ¶ 35 ("secure network [was] connected to an insecure outside world via the firewall 20.").) Lee explained that firewall 20 had "***three network interfaces***: one [went] to the inside of the intranet; one [went] to the unsecured external network 10; and the third [went] to the DMZ 30." (Lee, ¶36 (emphasis added).)

Netskope Exhibit 1002

FIG 1 (Annotated)



1[d]:

201. Lee disclosed that content detection data were used in the gateway system and Magdych disclosed configuration data used in its gateway. While neither Lee nor Magdych expressly disclosed that information being stored on the memory, a POSITA would have found that obvious.

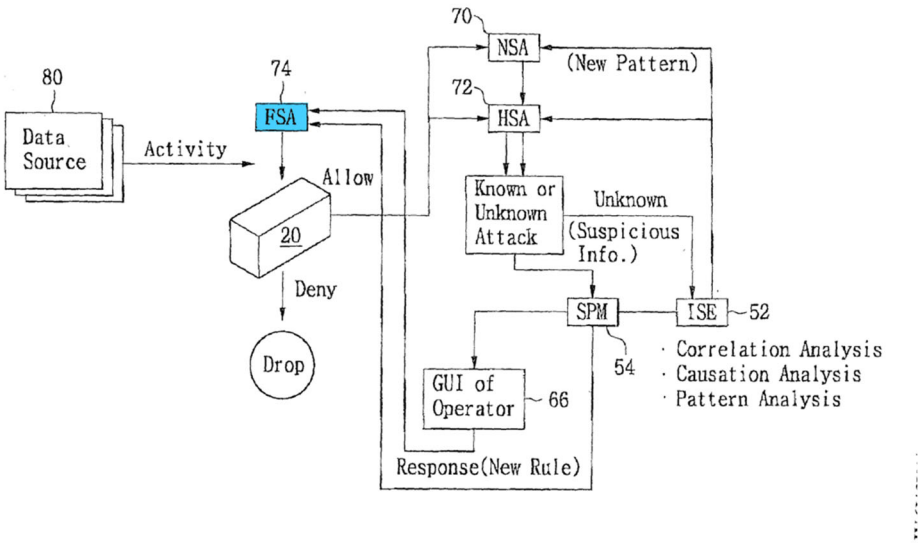
202. ***content detection data***: Lee disclosed that the FSA used content detection data.

203. As discussed in the Jordan-FortiGate ground, the '968 Patent broadly describes "content detection data." (*See* '968, 6:33-38 (emphasis added).)

Netskope Exhibit 1002

204. Lee disclosed that the FSA (74) (highlighted in Figure 2 below) used various information to screen traffic. (Lee, ¶46 ("In order to block traffic from the attackers, ***FSA 74 applies a security policy to the firewall 20 based on information transferred from SPM 54***.") (emphasis added).) For example, the firewall 20 filtered traffic based on whether the traffic matched a signature of an attack. (Lee, ¶ 52 ("[W]hen the ***new pattern of attack is recognized***, a new or modified rule is given to FSA 74 through SPM 54 so that traffic from the attacker 80 can be blocked.") (emphasis added); *see also, e.g.*, *id.*, ¶89.) As another example, the firewall 20 filtered traffic based on ***domain names and IP addresses***. (*Id.*, ¶34 ("The firewall 20 confirm[ed] if requests of an outsider [were] from permitted domain names or IP addresses.").)

FIG 2 (Annotated)



Netskope Exhibit 1002

205. ***configuration data*:** As described above, the '968 Patent does not define "configuration data." (*See* Section III.B.2, limitation 1[d].) Still, Magdych disclosed that predefined firewall rules could process traffic based on selectable parameters, such as a port. For example, the configuration data indicating a particular port number, indicated the nature of a service associated with the data (e.g., HTTP, Telnet, etc.) and caused the policy to restrict access based on the port number that was selected." (Magdych, 2:11-29, 2:55-53, 4:33-34, claim 1.) These parameters would be defined in policies.

206. A POSITA would have found it obvious to incorporate Magdych's selectable parameters into Lee because Lee already disclosed that the firewall had a graphical user interface for "enhanced control of network access and for advanced security features related to intrusion…." (Lee, ¶ 34.) When the NSA is integrated into firewall (20), as discussed above, a POSITA would have found it obvious that these functions would be performed in the combined device.

207. While neither Lee nor Magdych expressly disclosed that the content detection data was stored on memory in the network gateway, a POSITA would have found that obvious based on Lee's disclosures. As discussed above, firewall 20 received various information for analyzing traffic (e.g., rules, signatures). It was common practice at the time to store information for processing traffic on the device

Netskope Exhibit 1002

that was using the information to process the traffic because the device would need to retrieve the information and retain it in case the unit is shut off.

1[e]:

208. As discussed for element 7[pre], a POSITA would have found it obvious that Lee's firewall had an instruction set, stored in the at least one memory device and executable by the at least one processor to detect malicious traffic and respond in real-time.

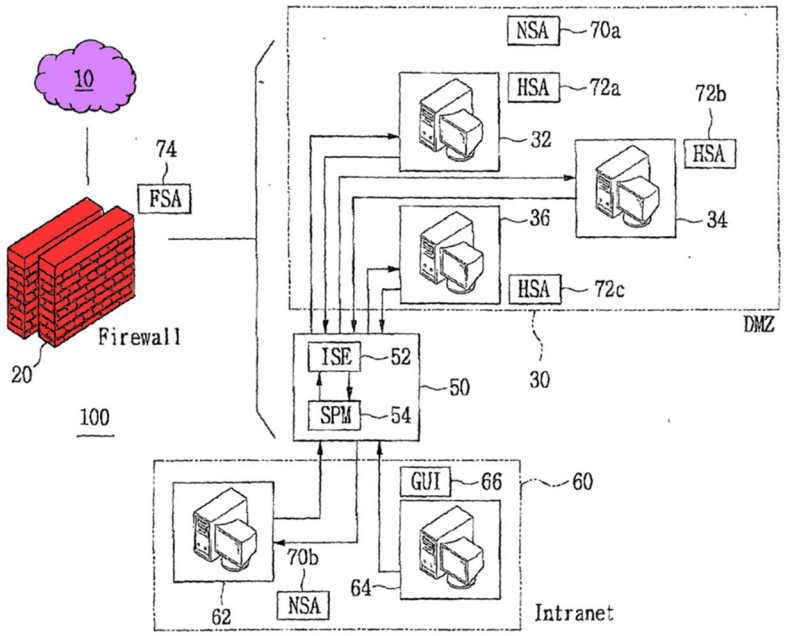
1[e][i]/4[a]/7[a]:

209. Lee disclosed that the firewall received data (network traffic) via the network interface device.

210. As described above for limitation 1[c], Lee disclosed that firewall 20 had a network interface device. Lee explained that firewall 20 had a network interface that went to an unsecured external network. (Lee, ¶36 (emphasis added).) As illustrated below in Figure 1 (below), Lee disclosed that firewall 20 received traffic from an outside network. (Lee, ¶35, ¶51.)

Netskope Exhibit 1002

FIG 1 (Annotated)



1[e][ii]/4[b][i]/7[b][i]:

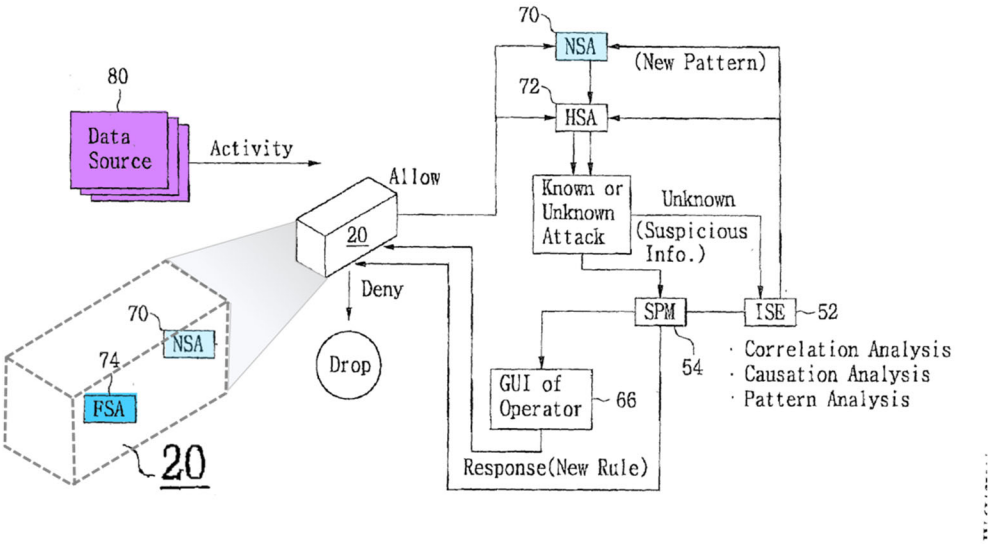
211. In the Lee-Magdych combination, firewall 20 processed network traffic according to the content detection and configuration data.

212. As shown in modified Figure 2 (below), firewall 20 received traffic from a data source. The FSA within the firewall then processed traffic according to the content detection data. (Lee, ¶52 ("[W]hen the new pattern of attack [was] recognized, a new or modified rule [was] given to FSA 74 through SPM 54 so that traffic from the attacker 80 can be blocked.") (emphasis added); *see also, e.g.*, *id.*, ¶¶ 89, 34 ("The firewall 20 confirm[ed] if requests of an outsider [were] from

Netskope Exhibit 1002

permitted domain names or IP addresses."), ¶36 ("All incoming HTTP connections headed for the internal network [were] blocked by the firewall 20….").)

FIG 2 (Redrawn and Annotated)



213. Magdych disclosed that received traffic was processed according to the selected parameter (e.g., port). (Magdych, cl. 1 ("discarding the data based on a predetermined set of rules utilizing a firewall associated with the gateway which is coupled to the remote source").)

214. As discussed for element 1[d], Lee combined with Magdych disclosed that the content detection and configuration data was stored in memory.

1[e][iii]/4[b][ii]/7[b][ii]:

215. Lee disclosed or suggested that the firewall enforced policies defined by the content detection data (e.g., virus pattern or signature), and Magdych showed

Netskope Exhibit 1002

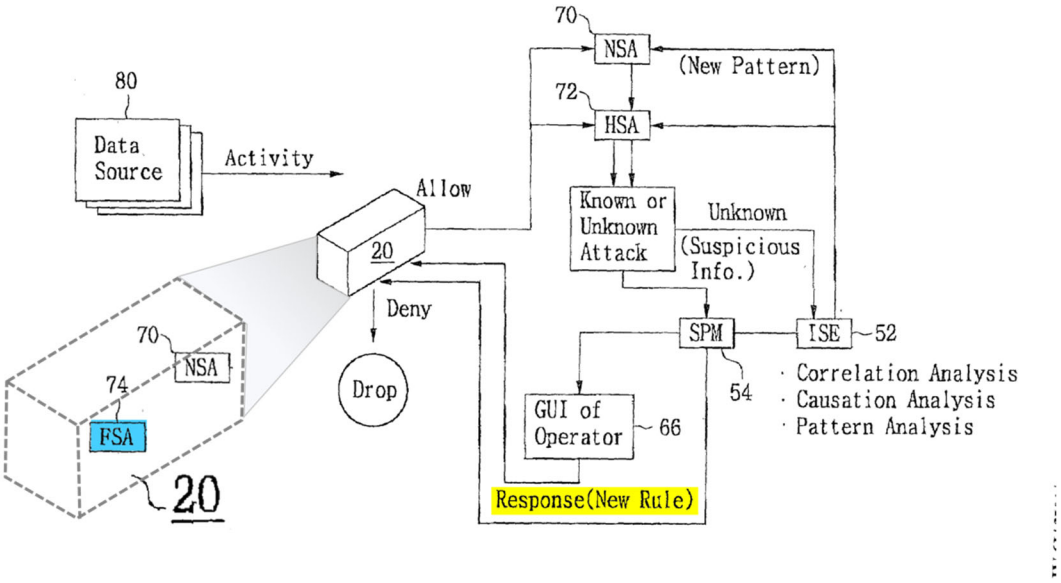
that a POSITA would have found it obvious that the firewall enforced policies defined by configuration data (e.g., port).

216. The FSA within the firewall applied policies to traffic to determine if traffic should be allowed or blocked. (Lee, ¶14; *see also, e.g.*, ¶34 ("The firewall 20…typically include[d] a graphic user interface (GUI) for…***security policy enforcement.***") (emphasis added), 46 ("FSA 74 [was] an active agent that ***adopt[ed] modified security policy*** according to the decision and analysis of ISE 52 and SPM 54, and ***[made] the firewall react accordingly***. ***In order to block traffic from the attackers, FSA 74 applie[d] a security policy to the firewall 20*** based on information transferred from SPM 54." (emphasis added)), 52, 92.)

217. The policies were defined by content detection and configuration data. For content detection data, for example, Lee disclosed that the policies/rules included a pattern or signature for detecting malicious data. (*See, e.g.*, *id.*, ¶52 ("At the same time, when the new pattern of attack [was] recognized, a new or modified rule [was] given to FSA 74 through SPM 54 so that traffic from the attacker 80 [could] be blocked.").) The firewall receiving the updated rule with the content detection data is shown in Figure 2 (annotated below).)

Netskope Exhibit 1002

FIG 2 (Redrawn and Annotated)



(*Id.*, Fig. 2.) A POSITA would have found it obvious that configuration data defined a policy based on Magdych and the knowledge of a POSITA. For example, Magdych disclosed that the configuration data discussed above (*see supra,* III.B.2, element 1[d]) defined rules and the rules instructed the firewall how to process traffic (e.g., which port(s) could traffic pass through). (Magdych, 4:36-38.)

1[e][iv]/4[b][iii]/7[b][iii]:

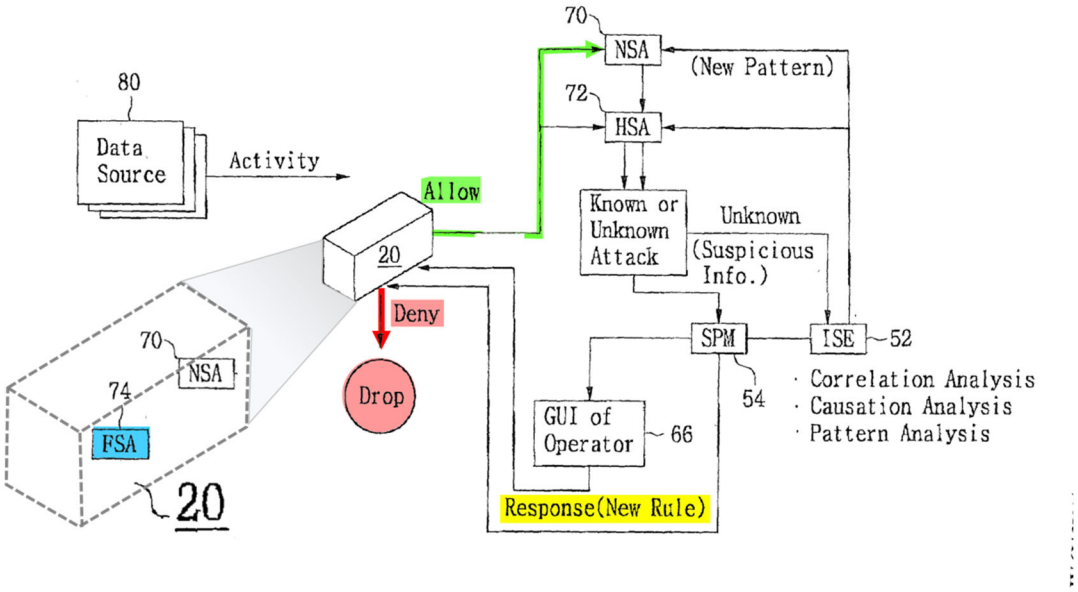
218. Lee disclosed that the firewall enforced policies to determine whether received network should pass onto the internal network.

219. For starters, Lee disclosed that the firewall (20) included a GUI for "security policy enforcement." (Lee, ¶32.) As part of that, firewall (20) determined whether to allow network traffic to pass. (*See, e.g.*, *id.*, Lee, ¶¶47, 52.)

Netskope Exhibit 1002

This was shown in Figure 2 (annotated below) where firewall (20) could either allow or deny traffic.

FIG 2 (Redrawn and Annotated)



1[e][v]/4[b][iv]/7[b][iv]:

220. Lee disclosed or suggested that policy enforcement included user identification, described that the policy enforcement included content identification, and Magdych disclosed source and destination verification.

221. ***User identification***: Lee suggested that the FSA used malware signatures that could include user identification. Specifically, Lee described an exemplary scenario where an attacker created a user ID. (Lee, ¶81.) The system could detect and react to that situation by determining if the attacker was a root user.

Netskope Exhibit 1002

(*Id.*, ¶84 ("If normal, ISE 52 confirms if the actor of the operation is a root user (step 154). When the actor is not a root user, a confrontation action is performed (step 156).").) A POSITA would have found it obvious that this analysis could occur in the firewall and user identification could be used by the firewall to enforce policies that specified a particular user ID.

222. ***Content identification***: Lee disclosed that policy enforcement included analyzing traffic using a malware signature. (Lee, ¶¶46 ("FSA 74 applies a ***security policy to the firewall 20*** based on information transferred from SPM 54.") (emphasis added), 49 ("SPM 54 instruct[ed] the application of dynamic policy to associated agents, and applie[d], to the agents, ***dynamic security policies*** according to a change of services provided by hosts and ***the detection signatures 52 generated by ISE***.") (emphasis added), 92.)

223. If Lee did not disclose content identification, Magdych did. Magdych disclosed that "assembled packets [were] analyzed in operation 506 for the purpose of ***identifying content*** that may be harmful to known vulnerabilities of the target devices 206." (Magdych, 5:1-3, cl. 1 (disclosing "comparing the data representing text to a predetermined list of data representing text associated with attacks utilizing the intrusion detection System.").) Obviously, this could include content like pornography.

Netskope Exhibit 1002

224. ***source and destination verification***: Magdych disclosed that the firewall "utilize[d] a predetermined set of rules to discard received data as a function of a plurality of parameters such as ***a source, a destination***, and/or a port associated with data." (Magdych, 4:36-39 (emphasis added).)

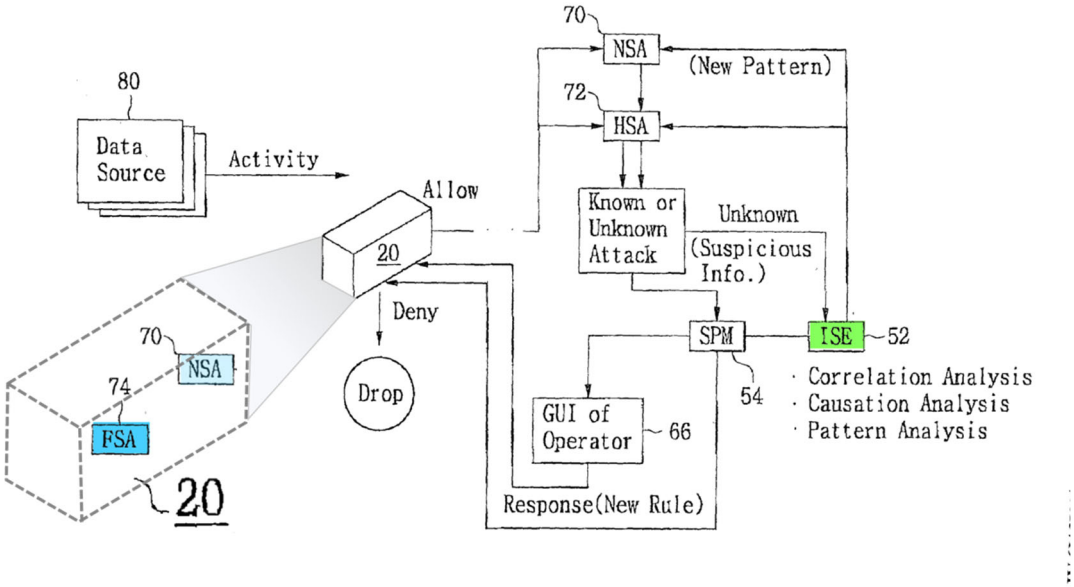
1[e][vi]/4[c][i]/7[c][i]:

225. In the Lee-Magdych combination, the firewall with NSA forwarded suspicious traffic to ISE (52) (analysis process), and ISE (52) performed anomaly detection on the traffic.

226. Lee disclosed that the NSA (70) "detect[ed] known attacks, suspicious information and traffic, and generate[d] a report to ISE 52." (Lee, ¶51.) Then, the suspicious traffic was analyzed by the ISE 52, which determined if there was an attack. (*Id.*, ¶ 2; *see also, e.g.*, *id.*, ¶41 ("NSA 70 confront[ed] threats against network security issue and provide[d] ***analysis of suspicious traffic*** and alert messages to known attacks…***NSA 70 deliver[ed] all traffic in a reduced form to ISE 52***, and ISE 52 then perform[ed] anomaly detection based on the delivered traffic.") (emphasis added), 68.)

Netskope Exhibit 1002

FIG 2 (Redrawn and Annotated)



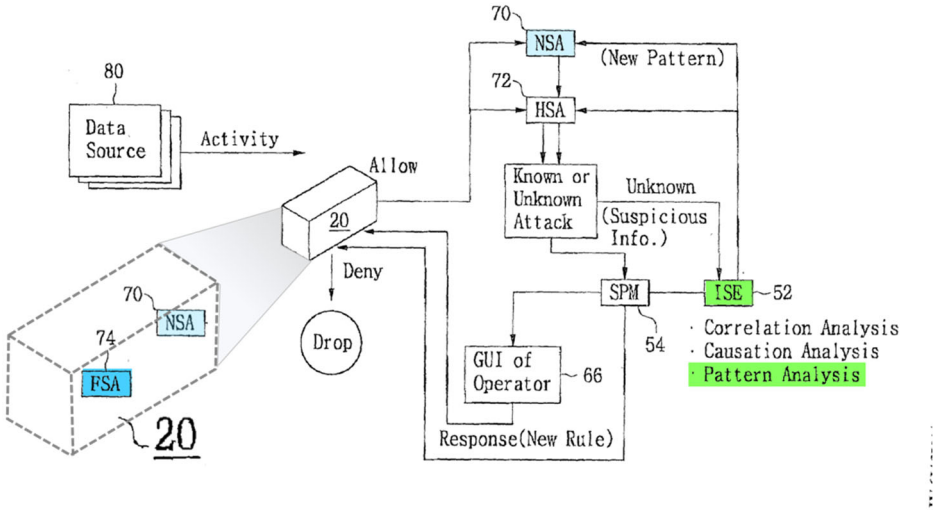
1[e][vii]/4[c][ii]/7[c][ii]:

227. Lee described that when network traffic was threatening, the ISE generated additional content detection data to detect the threat in subsequent traffic.

228. Lee disclosed that ISE (52) "perform[ed] an anomaly detection and, if detected as an attack and the attack [was] an unknown pattern, a new signature [was] generated through a learning process." (Lee, ¶52; *see also, e.g.*, *id.*, ¶48 (disclosing that ISE "analyze[d] alert messages from agents installed within each of individual security systems [and] determine[d] if there if [sic] an attack and ***generate[d] a signature through learning***.") (emphasis added).) This was shown in Figure 2 (below), which depicted a Pattern Analysis in ISE (52).

Netskope Exhibit 1002

FIG 2 (Redrawn and Annotated)

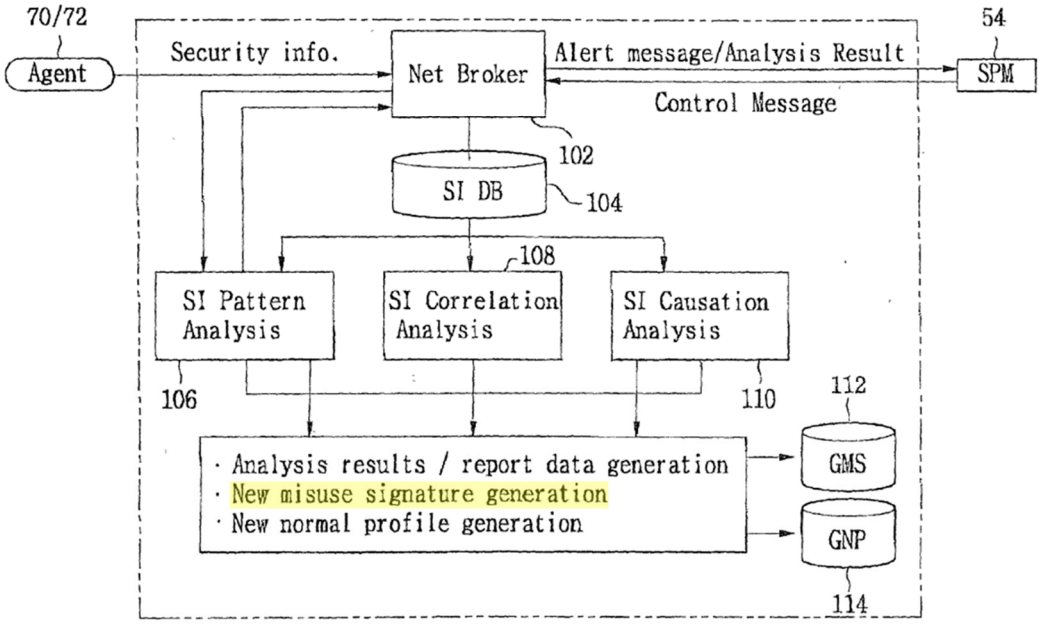


229. The updated signature was then transmitted to the various agents. (*Id.*, ¶52 ("The generated signature [was] transferred to NSA 70 and HSA 72, so that more rapid confrontation in response to future attacks of the same pattern [was] made possible. At the same time, when the new pattern of attack [was] recognized, a new or modified rule [was] given to FSA 74 through SPM 54 so that traffic from the attacker 80 can be blocked.").)

230. Additionally, Figure 4 (below) depicted "[n]ew misuse signature generation." (Lee, ¶¶ 55-56.)

Netskope Exhibit 1002

FIG 4 (Annotated)

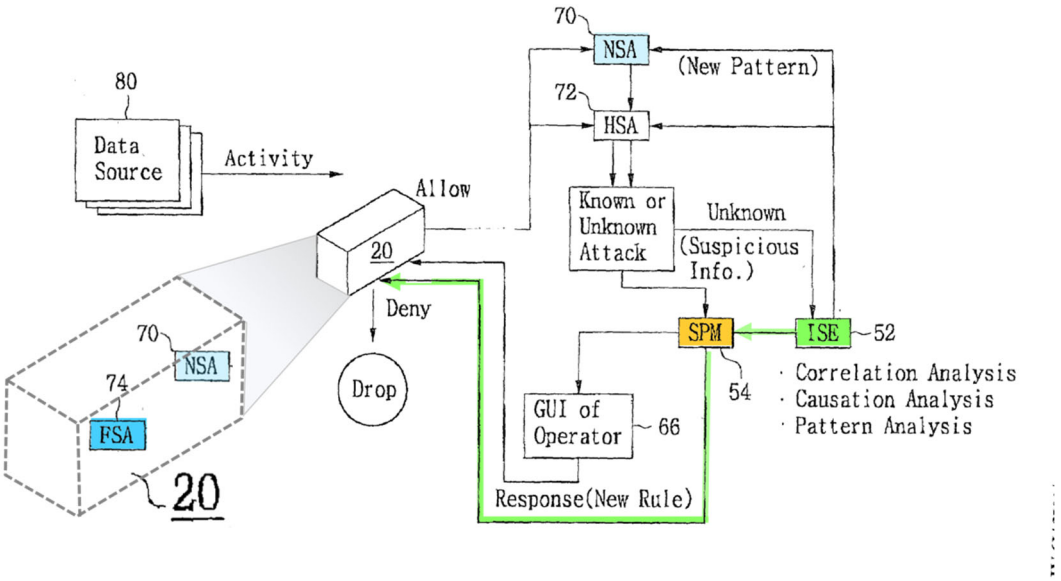


1[f]/4[d]/7[d]:

231. Lee's SPM acted as an update station by providing new rules generated by the ISE to the agents, including the FSA in firewall (20). Specifically, Lee disclosed that a generated signature from the ISE was transferred to the NSA (70) and FSA (74) in firewall (20). (Lee, ¶52; *see also, e.g.*, *id.*, ¶¶46, 49, 56.) This was shown in Figure 2 (annotated below), where the "new pattern" was transmitted from ISE to SPM, then SPM to FSA.

Netskope Exhibit 1002

FIG 2 (Redrawn and Annotated)



232. While Lee did not explicitly describe that the new rules was received via a network interface device, a POSITA would have found it obvious because the SPM transferred digital information to firewall (20) and firewall (20) had a network interface for the internal part of the network. (Lee, ¶¶36, 46.)

1[g]/4[e]/7[e]:

233. Lee did not explicitly describe that the content detection data was stored on the memory device, but a POSITA would have found it obvious that it was. First, Lee described that generated signatures were given to NSA (70) and a new or modified rule is given to the FSA (74). (Lee, ¶52.) Lee also described that in another embodiment, the security policy could be applied to other intranets and their associated firewalls. (Lee, ¶89.) In view of this information, a POSITA would

Netskope Exhibit 1002

understand that the FSA and NSA would store the updated rules or signatures, such as content detection data, in order to use it and apply it to future traffic. Otherwise, the information would be lost.

1[h]/4[f]/7[f]:

234. Lee disclosed that the modified firewall used generated signatures and new/modified rules (additional content detection data) to process traffic after it received those signatures and rules. Specifically, Lee disclosed that FSA included "***adopting a security policy*** transferred from the SPM and ***causing the firewall to block a traffic*** from an attacker." (*Id.*, ¶14 (emphasis added); *see also id.*, ¶¶51 (stating SPM "applies a new rule to FSA 74 ***to make the firewall 20 block traffic*** from the attack data source 80") (emphasis added), 52, 89.)

1[i]/4[g]/7[g]:

235. Lee disclosed blocking networking traffic determined to violate at least one policy. Specifically, Lee disclosed that "a new or modified rule is given to FSA (74)…so that traffic from attacker 80 ***[could] be blocked***." (Lee, ¶52 (emphasis added); *see also, id.*, ¶¶46, 51, 89, Claim 10.)

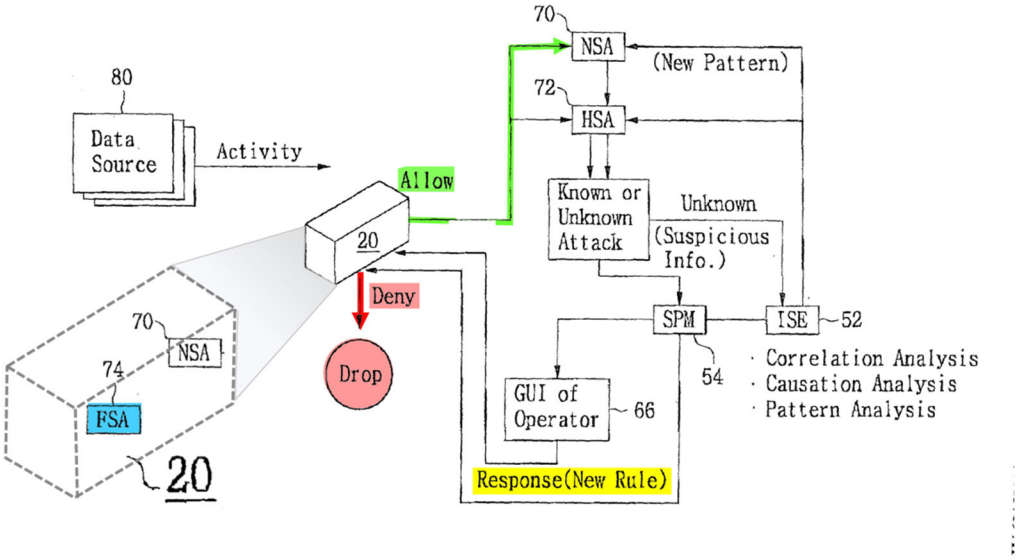
1[j]/4[h]/7[h]:

236. While Lee did not explicitly disclose that traffic that did not violate a policy was allowed to pass, but a POSITA would understand that Lee obviously disclosed this limitation. Specifically, Lee describes that the FSA can block traffic from an attacker (Lee, ¶14; *see also id.*, ¶¶46, 51, 52, 89.) Lee also illustrated in

Netskope Exhibit 1002

Figure 2 (modified and annotated below) that the firewall allowed traffic to pass. A POSITA would understand that since the FSA would block traffic that matched the policy with the new signature, it would allow signatures to pass that did not match the policy.

FIG 2 (Redrawn and Annotated)



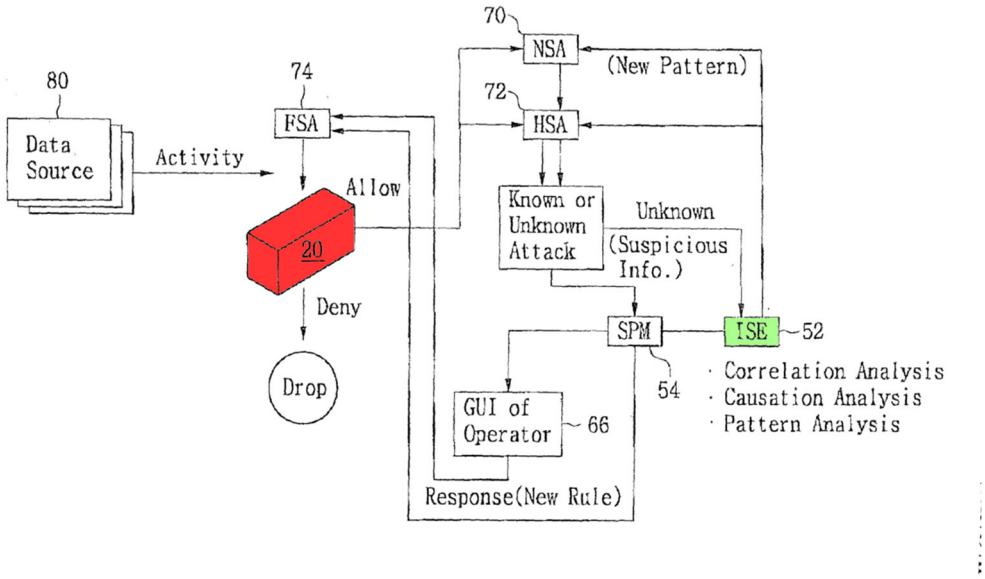
3. Claims 2, 5, and 8:

237. Lee disclosed that the analysis process was executed on a computing device distinct from the data network device. Specifically, Lee described that the ISE may be located in a security management module (50), which was distinct from the firewall. (Lee, ¶38 describing that an "intelligent security management module 50 comprises ISE…" but "[f]or the firewall 20, an FSA … is provided."); *see also id.*, ¶47, 48 ("ISE 52 is one of the analysis engines which analyzes alert messages from agents [e.g., FSA, NSA] installed within each of individual security systems.").)

Netskope Exhibit 1002

The distinction between the ISE and other components of the system was shown in Figure 2 of Lee (annotated below).

FIG 2 (Annotated)



4. Claims 3, 6, and 9:

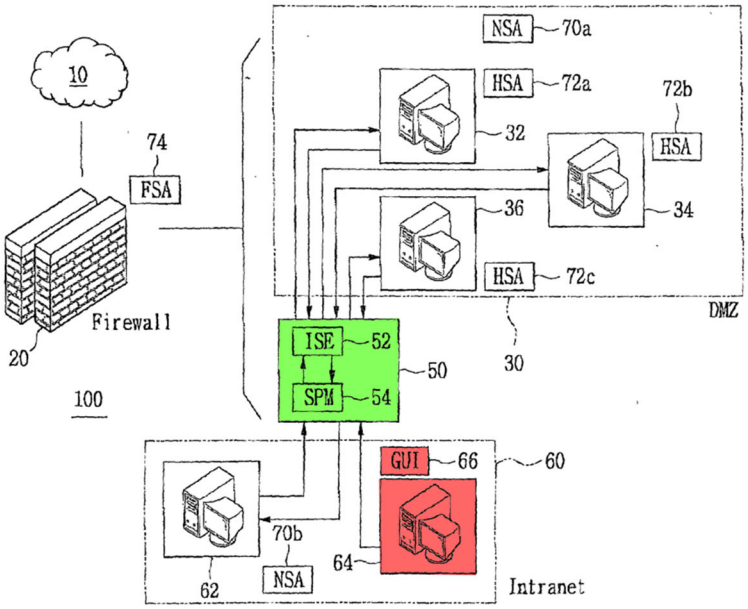
238. Lee disclosed that policy enforcement provided a GUI, which would facilitate receiving administrator input. Lee disclosed that the firewall "typically includes a ***graphic user interface (GUI) for enhanced control of*** network access and for advanced security features related to intrusion and statistics on network uses and ***security policy enforcement***." (Lee, ¶34 (emphasis added).) While this discussion did not focus on the ISE, Lee later described that the ISE could also be controlled using a GUI. Specifically, Lee described that a manager system could

Netskope Exhibit 1002

"control an intelligent security management module 50, [which comprised ISE 52], ***through GUI***." (Lee, ¶ 38.)

239. A POSITA would find it obvious that a GUI, which could control the ISE (e.g., the analysis process), meant that an administrator could provide input into the ISE's analysis. This is further demonstrated in Figure 1 (annotated) of Lee.

FIG 1 (Annotated)



240. Having an administrator analyze suspect traffic was nothing new. For example, U.S. Pat. No. 7,080,408 ("Pak") (Ex. 1019), was directed to "Network Communications Having Suspicious Contents." (Pak, Title.) Pak discussed that when analyzing suspicious traffic, an administrator could be involved. Pak, 6:25-28 ("Another pattern includes operation 320, in which the suspect content is sent to an expert for evaluation before being released from quarantine. The expert can be akin

Netskope Exhibit 1002

to the scientist described above."; *see also id.*, 6:28-35.) Accordingly, receiving administrator input to analyze the suspicious traffic was common before the time of the alleged invention.

D. Ground 3: Combining Lee, Magdych, and Judge renders claims 1-9 obvious.

241. To the extent that the Board does not agree with Petitioner's previous interpretations of "configuration data" for elements 1[d]; 1[e][ii]/4[b][i]/7[b][i]; 1[e][iii]/4[b][ii]/7[b][ii], a POSITA would have found it obvious to combine the teachings of Lee and Magdych with Judge's express disclosure of "configuration data" and examples thereof in the context of determining rules/policies for responding threats that included viruses, spam, worms, and other attacks. (Judge, Abstract, ¶¶2, 44.) Thus, Judge serves the dual purpose of curing deficiencies, if any, of the Lee-Magdych combination and confirming the knowledge of a POSITA regarding configuration data used in intrusion detection systems.

242. Judge described analogous technology to the '968 Patent, Lee and Magdych, which is network security. (Judge, Abstract.) Specifically, Judge described an "architecture" with "a centralized threat management center [(TMC)] that can collect threat information and create rules and/or policies for messaging security systems [(MSS)]." (Judge, ¶40.) "An MSS is an example of an application layer security system such as hardware device 210" (*id.*, ¶78) and that hardware

Netskope Exhibit 1002

device "could be incorporated into a standard firewall system 140 or router…" (*id.*, ¶81.)

243. Judge further described when analyzing threat information (e.g., threatening communications/traffic), "features in conjunction with ***configuration data such as goals*** are ***used to produce rules***." (*Id.*, Abstract.) Judge also recognized that "***configuration information can include threat types of interest*** and weights associated therewith." (*Id.*, ¶44.) This configuration information could also be modified. (*Id.*, ¶45 ("[A]n interface may be provided to allow establishing configuration information regarding one or more threat types….").)

244. A POSITA would have found it obvious to incorporate these examples of configuration data into the Lee-Magdych combination because doing so would increase the parameters, features, accuracy, and control of the Lee's modified filtering system. By creating rules/policies that included goals or threat types of interest, Lee's modified filtering system could respond more accurately and more appropriately based on the desired intent of a specific security system or based on known vulnerabilities of a protected network. For example, a threat-type of interest might be incoming traffic that does not match a virus signature but exhibits malicious behavior patterns that can be associated with a threat. Incorporating that threat type of interest into the security system would allow the malicious traffic to be detected or prohibited when it otherwise would not be. Incorporating goals and

Netskope Exhibit 1002

threat types of interest (e.g., configuration data) into the applied policies would help better protect networks.

E. Ground 4: Combining Fang and FortiGate renders claims 1-9 obvious.

245. ***Fang[6]:*** Fang has the same description as the '968 Patent, and precisely like the '968 Patent, disclosed "systems and methods for updating content detection devices and systems." (Fang, Title.) Thus, the description in Fang also describes or renders obvious the alleged invention of the '968 Patent (*see, supra*, Section II.C.2.)[7]

246. ***FortiGate:*** FortiGate is described above (*see supra*, Section III.B.)

1. Motivation to combine

247. A POSITA would have been motivated to combine Fang with FortiGate with a reasonable expectation of success for several reasons.

248. Fang and FortiGate concerned network security and, in particular, detecting malicious content. Fang generally described means for processing network traffic content in view of certain information to determine how to handle the traffic. (Fang, Abstract.) Likewise, FortiGate "support[ed] network-based deployment of application-level services-including antivirus protection and full-scan content filtering." (FortiGate, 13.) It provided "improve[d] network security,

[6] Given that these claims lack § 112 support, Fang is prior art. (*See, supra*, Section II.C.2.)
[7] Fang is the printed publication of application '915 (the grandparent application of the '968 Patent) and has the same figures and detailed description as the '968 Patent.

Netskope Exhibit 1002

reduce[d] network misuse and abuse, and help[ed] [] use communications resources more efficiently…." (*Id.*)

249. Moreover, both Fang and FortiGate described higher-level modules that created updated signatures and distributed those updated signatures to the firewall units (e.g., content detection modules) via an intermediary device, such as an "update station." (*See* Section III.E.2, Limitation 1[f] infra.) Thus, a POSITA would have been motivated to improve Fang by incorporating complementary techniques from other references, like FortiGate, that addressed the same problem and consisted of a similar architecture.

250. Element-specific motivations to combine are discussed below.

2. Claims 1, 4, and 7:

251. These claims recite similar limitations and are, thus, analyzed together.

1[pre]:

252. Fang disclosed a "content detection module (108)" that received electronic content, such as network traffic. (Fang, ¶31.) Fang depicted content detection module (108) in Figure 1 (annotated below). Fang explained that the content detection module could be a gateway. (*Id.*)

Netskope Exhibit 1002

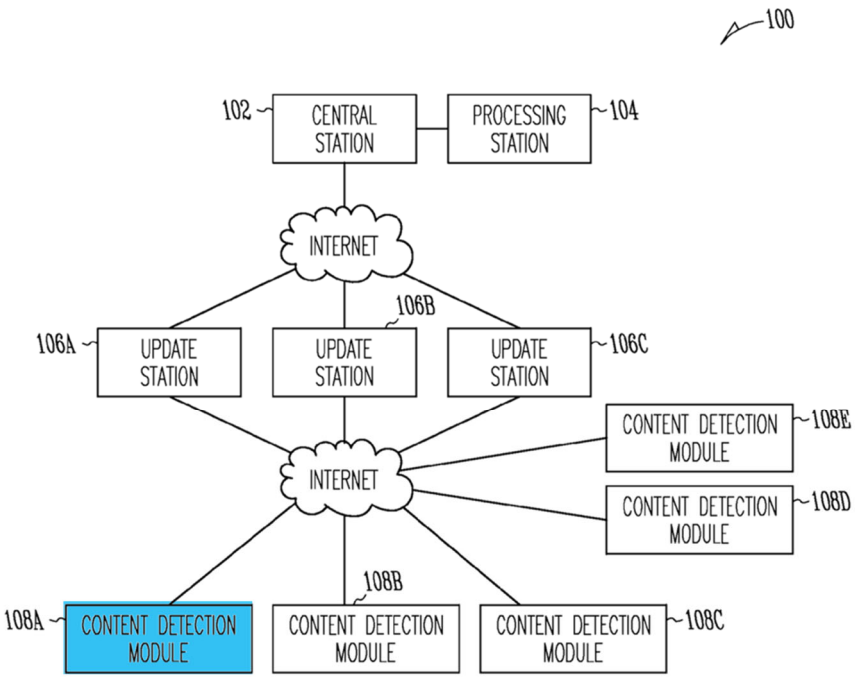


Fig. 1
(Annotated)

4[pre]:

253. Fang disclosed "methods for updating content detection services and systems." (Fang, Title; *see also, e.g.*, *id.*, ¶ 9, Cl. 7.)

7[pre]:

254. Fang disclosed this element, as shown in Figure 2 (below). (*See also, e.g.*, *id.*, ¶¶45, 47, 49, 50, 52, cl. 13.)

Netskope Exhibit 1002

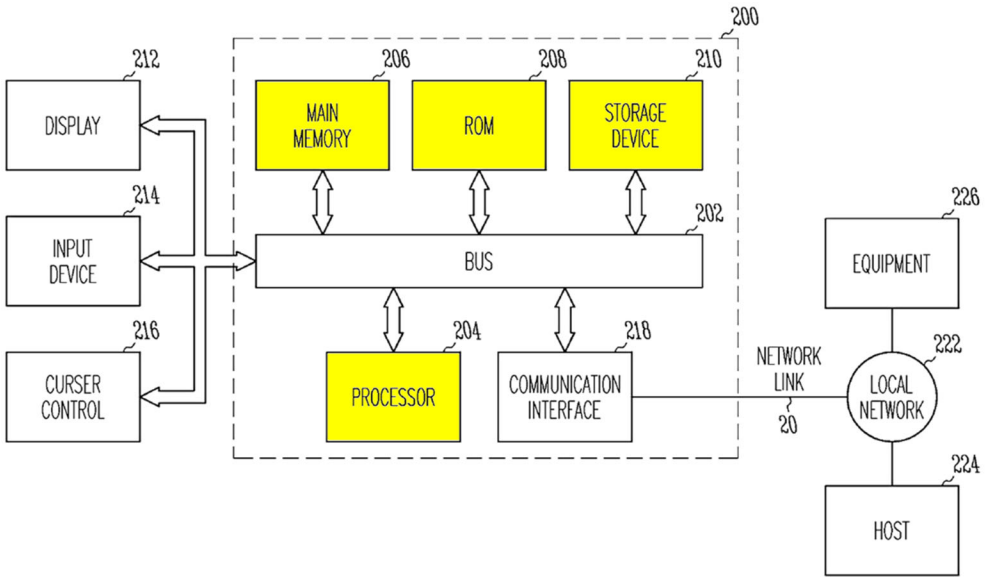


Fig.2
(Annotated)

1[a]:

255. As discussed for element 7[pre], Fang disclosed a processor. (*See, e.g.*, Fang, ¶¶47, 49, 52, Fig. 2.)

1[b]:

256. As discussed for element 7[pre], Fang disclosed a memory device in the form of storage device (210), main memory (206), and read-only memory (208). (*See, e.g.*, Fang, ¶¶47, 49-50, 52, Fig. 2.)

1[c]:

257. Fang disclosed that content detection module (108) (network gateway) had a network interface device.

Netskope Exhibit 1002

258. Fang disclosed that content detection module (108) had a communication interface, annotated in Figure 2 (below). (Fang, ¶53.) Communication interface (218) provided, *inter alia*, "a two-way data communication coupling to a network link 220 that is connected to a local network 222." (*Id.*)

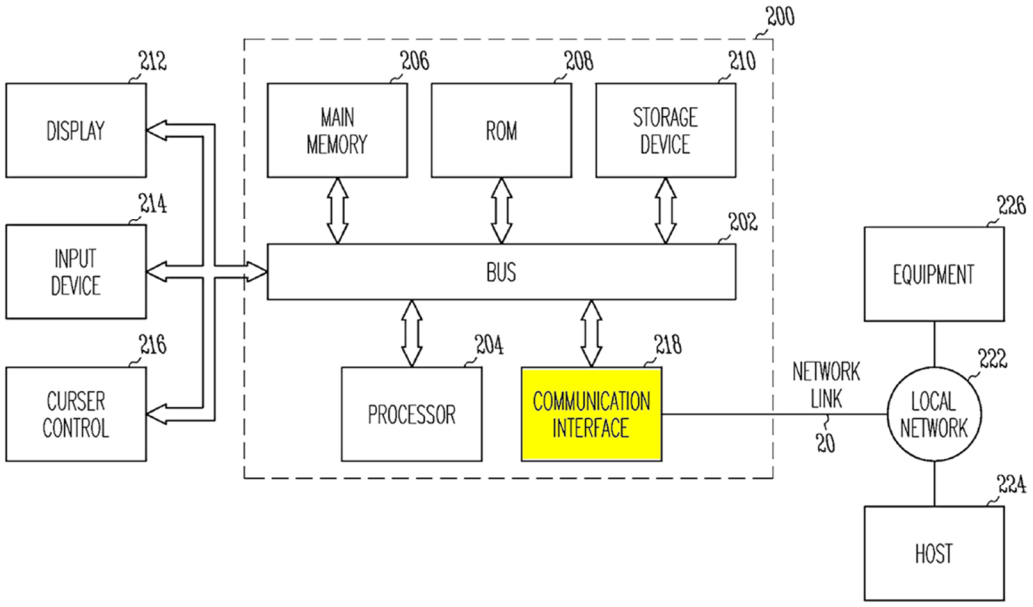


Fig.2
(Annotated)

259. Moreover, Fang disclosed a "network interface device" for receiving content. (*Id.*, Cls. 7, 13.) A POSITA would have understood that the network interface device and communication interface referred to the same thing because POSITAs generally used them interchangeably. Figure 2 of Fang illustrates the communication interface (218).

Netskope Exhibit 1002

1[d]:

260. Fang disclosed content detection data (e.g., a virus signature) stored in the memory device. While Fang did not disclose configuration data, a POSITA would have found it obvious to add that information to Fang based on FortiGate's disclosures.

261. ***content detection data:*** Fang disclosed that "***content detection data*** [could] be a virus signature, a virus definition, a spammer identification, a URL, a NIDS signature, a time at which content detection data is created, a level of threat, etc., and the like, e.g., any information that can be used in a content detection or screening process." (*Id.*, ¶34 (emphasis added).)

262. ***configuration data*:** As discussed in Section II.C.2, the challenged patent failed to disclose "configuration data." By extension, Fang too failed to disclose "configuration data." Still, a POSITA would have found it obvious to add configuration data to Fang based on FortiGate's disclosures. (*See supra*, Section III.B.2, Limitation 1[d].)

263. By incorporating these configurations and the others disclosed in FortiGate, Fang's content detection module would have improved flexibility and features for filtering and controlling traffic. These configurations could naturally be incorporated into Fang's content detection module, which likewise processed incoming traffic like the FortiGate firewall.

Netskope Exhibit 1002

264. ***stored on the at least one memory device:*** While Fang and FortiGate did not expressly disclose storing content detection and configuration on memory, a POSITA would have found that was obvious based on their disclosures. For starters, Fang disclosed that content detection module (108) received content detection data from an update station and determined whether content contained undesirable content based on the content detection data. (*Id.*, ¶31.) For example, content detection module (108) could use a virus signature received from an update station to detect a virus. (*Id.*) Likewise, FortiGate disclosed that configuration data was defined in a policy that FortiGate used when processing traffic. (*E.g.*, FortiGate, 35 (stating that content profiles, which could be added to NAT settings, allowed a user to "build up protection configurations that can easily be applied to different types of Firewall policies").) Accordingly, a POSITA would have found it obvious that the content detection and configuration data were stored in memory because it would need to be in order for the network gateway/firewall to access and use the information when processing traffic.

1[e]:

265. As discussed for element 7[pre], Fang disclosed that content detection module (108) had an instruction set that was stored on a memory device and executable by a processor to perform certain operations. (*See, e.g.*, ¶¶47, 49, 52, Fig. 2.)

Netskope Exhibit 1002

1[e][i]/4[a]/7[a]:

266. Fang disclosed that content detection module (108) received electronic content (e.g., content data, such as traffic) over a communication/network interface.

267. Fang disclosed that content detection module (108) "receive[d] electronic content (content data), and determine[d] whether the electronic content contain[ed] undesirable content." (Fang, ¶ 31; *see also, e.g.*, *id.*, ¶ 33 ("For example, a content detection module 108 may detect a content that [was] suspicious (or that require[d] further processing).").)

268. While Fang did not expressly disclose that content detection module (108) received the traffic over a network interface device, a POSITA would have found it obvious that it did based on Fang's disclosures. Specifically, as discussed above, Fang disclosed a communication interface that provided two-way data communications. Fang also disclosed that the communication interface "[c]omputer system 200 [could] send messages and ***receive data***, including program code, ***through*** the network(s), network link 220, and ***communication interface 218***." (*Id.*, ¶ 54 (emphasis added); *see also, e.g.*, *id.*, cl. 7 ("receiving content via a network interface device;").) Moreover, Figure 1 (below) depicted content detection modules connecting to the Internet. Consequently, a POSITA would have found it obvious that the content detection module received traffic over a network interface device.

1[e][ii]/4[b][i]/7[b][i]:

Netskope Exhibit 1002

269. Fang disclosed that content detection module (108) processed received traffic in view of content detection data (e.g., virus signatures), and FortiGate's firewall processed received traffic in view of configuration data (e.g., "Traffic Shaping," "Authentication").

270. Fang disclosed that content detection module (108) processed the received traffic in view of the content detection data. For example, Fang disclosed that each content detection module "determine[d] whether the electronic content contain[ed] undesirable content ***based on the content detection data*** …." (Fang, ¶ 31; *see, e.g.*, *id.*, ¶12 ("the content detection data usable by a content detection module to detect content").) For example, content detection module "[could] be configured to detect virus based on a virus signature received from update station 106."

271. Additionally, FortiGate disclosed that received traffic was processed according to firewall policies that were defined by configuration data. As described above in element 1[d], an administrator could configure the, for example, "Traffic Shaping" and "Authentication." (FortiGate, 34-35; *see also id.* 16.) Table 4 of FortiGate (below) described how the FortiGate unit received and processed traffic according to the configuration data.

Netskope Exhibit 1002

Table 4: Factory default firewall configuration

Internal Address	Internal_All	IP: 0.0.0.0 Mask: 0.0.0.0	Represents all of the IP addresses on the internal network.
External Address	External_All	IP: 0.0.0.0 Mask: 0.0.0.0	Represents all of the IP addresses on the external network.
Recurring Schedule	Always		The schedule is valid at all times. This means that the firewall policy is valid at all times.
Firewall Policy	**Internal->External**		Firewall policy for connections from the internal network to the external network.
	Source	Internal_All	The policy source address. Internal_All means that the policy accepts connections from any internal IP address.
	Destination	External_All	The policy destination address. External_All means that the policy accepts connections with a destination address to any IP address on the external network.
	Schedule	Always	The policy schedule. Always means that the policy is valid at any time.
	Service	ANY	The policy service. ANY means that this policy processes connections for all services.
	Action	ACCEPT	The policy action. ACCEPT means that the policy allows connections.
	☑ **NAT**		NAT is selected for the NAT/Route mode default policy so that the policy applies network address translation to the traffic processed by the policy. NAT is not available for Transparent mode policies.
	☐ **Traffic Shaping**		Traffic shaping is not selected. The policy does not apply traffic shaping to the traffic controlled by the policy. You can select this option to control the maximum or minimum amount of bandwidth available to traffic processed by the policy.
	☐ **Authentication**		Authentication is not selected. Users do not have to authenticate with the firewall before connecting to their destination address. You can configure user groups and select this option to require users to authenticate with the firewall before they can connect through the firewall.
	☐ **Antivirus & Web Filter**		Antivirus & Web Filter is not selected. This policy does not include a content profile that applies antivirus protection, web content filtering, or email filtering to content traffic processed by this policy. You can select this option and select a content profile to apply different levels of content protection to traffic processed by this policy.
	☐ **Log Traffic**		Log Traffic is not selected. This policy does not record messages to the traffic log for the traffic processed by this policy. You can configure FortiGate logging and select Log Traffic to record all connections through the firewall that are accepted by this policy.

(Annotated)

Netskope Exhibit 1002

272. Further, as described above in Section III.B.2, limitation 1[e][ii] and as shown above in Table 4 (annotated above), the firewall processed traffic according to the configuration settings. (FortiGate, 34-35.). This is further discussed in limitation 1[d] of Ground I discussed above. Thus, FortiGate disclosed that the received traffic was processed in view of configuration data.

1[e][iii]/4[b][ii]/7[b][ii]:

273. Fang disclosed that content detection module (108) (network gateway) enforced policies that were based on content detection data (e.g., virus lists). FortiGate disclosed that its FortiGate firewall applied policies to traffic based on configuration data. Fang also disclosed that its policy enforcement included identifying (e.g., defining) suspicious traffic.

274. Fang disclosed that content detection module (108) "perform[ed] policy enforcement." (*Id.*, ¶31.) Fang explained that policy enforcement included, *inter alia*, "virus scanning, content scanning (e.g., scanning for undesirable content), and intrusion detection (e.g., detecting undesirable content, such as worms, porno website, etc.)." (*Id.*) Fang described that each content detection module "determines whether the electronic content contains undesirable content based on the content detection data it receives from update station…." (Fang, ¶31.)

275. As addressed above in element 1[d], Fang did not disclose that the policy enforcement included configuration data.

Netskope Exhibit 1002

276. However, FortiGate described configuration settings, which included allowing an administrator to configure "Traffic Shaping" and "Authentication," for example. (FortiGate, 34-35; *see also id.* 16.)

277. Fang also disclosed that content detection module (108) detected "suspicious" content. For example, Fang disclosed that "content detection module 108 may detect a content that [was] suspicious (or that requires further processing)…." (*Id.*, ¶ 33.) While Fang did not disclose that what constituted "suspicious" was defined in a policy, a POSITA would have found that obvious because a network administrator would need to define what constitutes "suspicious" so the content detection module could detect the data.

1[e][iv]/4[b][iii]/7[b][iii]:

278. Fang disclosed that content detection module (108) performed policy enforcement. (Fang, ¶31.) Fang explained that "'policy enforcement' refer[ed] to a process or procedure, an execution of which ***create[d] a result that can be used to determine whether to pass data to user***…." (*Id.* (emphasis added).)

1[e][v]/4[b][iv]/7[b][iv]:

279. Fang disclosed that policy enforcement included "source verification, destination verification … virus scanning, content scanning (e.g., scanning for undesirable content), and intrusion detection (e.g., detecting undesirable content, such as worms, porno website, etc.)." (Fang, ¶31.)

Netskope Exhibit 1002

280. While Fang did not explicitly disclose user identification, a POSITA would have found it obvious to add the parameter based on FortiGate, as discussed above. (*See* supra, Section III.B.2, Limitation 1[e][v].)

1[e][vi]/4[c][i]/7[c][i]:

281. Fang disclosed that when the content detection module (108) detected suspicious content, it "sen[t] the electronic content to processing station 104 for processing." (Fang, ¶ 33; *see also id.*, cl. 1 (disclosing "determine[ing] whether received content contain[ed] suspicious content data or the suspicious content data [was] a threat to detect.").)

1[e][vii]/4[c][ii]/7[c][ii]:

282. Fang described that when network traffic contained a threat, the analysis process generated additional content detection data to detect the threat in subsequently received network traffic. (Fang, ¶ 34.)

283. Fang disclosed that, after the procession station received the traffic, it "analyze[d] such information to determine whether the content contain[ed]/[was] a threat (e.g., a virus, a worm, a spam, etc.) that [was] desired to be detected." (*Id.*, ¶34.) If so, "processing station 104 then generate[d] content detection data for the electronic content." (*Id.*) More specifically, Fang disclosed "[i]f processing station 104 determine[d] that the set of content data include[d] content that [was] undesirable,…processing station then generate[d] content detection data, which

Netskope Exhibit 1002

[could] be used by content detection modules 108 to detect the undesirable content.: (*Id.*)

1[f]/4[d]/7[d]:

284. Fang disclosed that content detection module (108) received additional content detection data indirectly from the central station (102) via update stations (106).

285. Fang disclosed that "[u]pdate stations 106 receive[d] the content detection data from central station 102, and sen[t] the content detection data to content detection modules 108 (e.g., through the Internet)." (Fang, ¶30.) Figure 1 of Fang is also illustrative of the update station acting as an intermediary device distributing the content detection data to the content detection modules.

Netskope Exhibit 1002

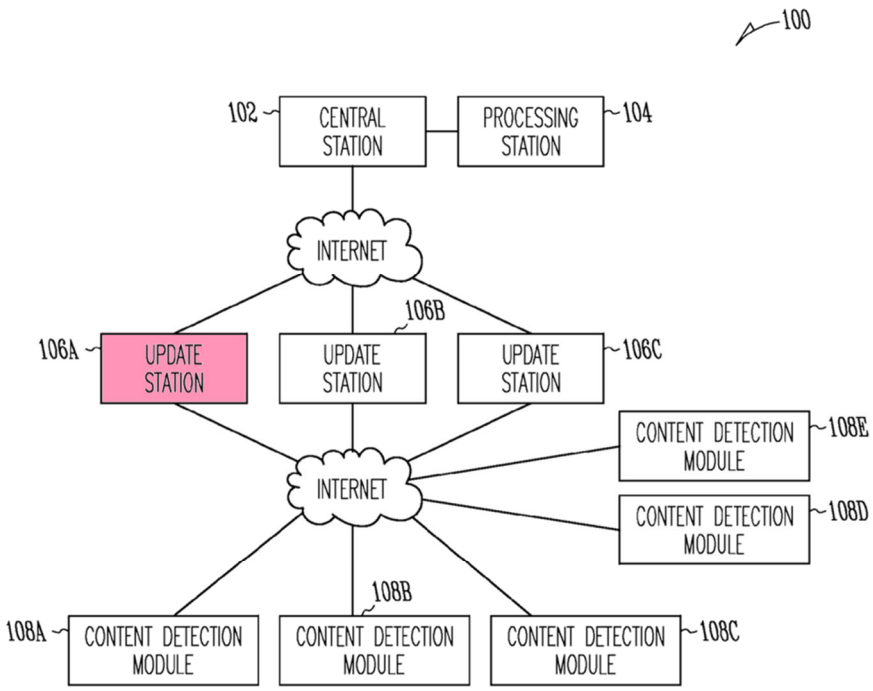


Fig. 1
(Annotated)

1[g]/4[e]/7[e]:

286. Fang did not explicitly disclose that the additional content detection data would be stored on the at least one memory device. However, a POSITA would find it obvious that it was.

287. Fang disclosed that the content detection module (108) received content detection data from an update station and determined whether content contained undesirable content based on the content detection data. (*Id.*, ¶31.) For example, content detection module (108) could use a virus signature received from an update station to detect a virus. (*Id.*). A POSITA would have found it obvious that the content detection data was stored in memory because it would need to be for the

Netskope Exhibit 1002

network gateway (content detection module) to access and use the information when processing traffic.

1[h]/4[f]/7[f]:

288. Fang disclosed that the content detection module processed traffic using content detection data from the processing station. (Fang, ¶34 ("processing station then generates content detection data, which can be used by content detection modules 108 to detect the undesirable content.").) For example, Fang disclosed that after the processing station received and analyzed the content, the "processing station then generate[d] content detection data which [could] be used by content detection modules 108 to detect the undesirable content." (*Id.*, *see also, id.*, ¶42.)

1[i]/4[g]/7[g]:

289. Fang did not explicitly disclose that the content detection modules "block" traffic, but a POSITA would find it obvious. Fang described that the content detection module that performed policy enforcement to determine whether to pass the data (e.g., traffic) to a user. (Fang, ¶31.) A POSITA would understand that by determining whether to "pass" data means that data may alternatively be blocked.

1[j]/4[h]/7[h]:

290. Fang described that the content detection module allowed traffic that did not violate a policy to pass. (Fang, ¶31.)

Netskope Exhibit 1002

3. Claims 2, 5, and 8:

291. Fang disclosed that the analysis process was executed on a distinct device. For example, Figure 1 of Fang disclosed that the processing station was a separate device from the content detection module and was separated by the Internet.

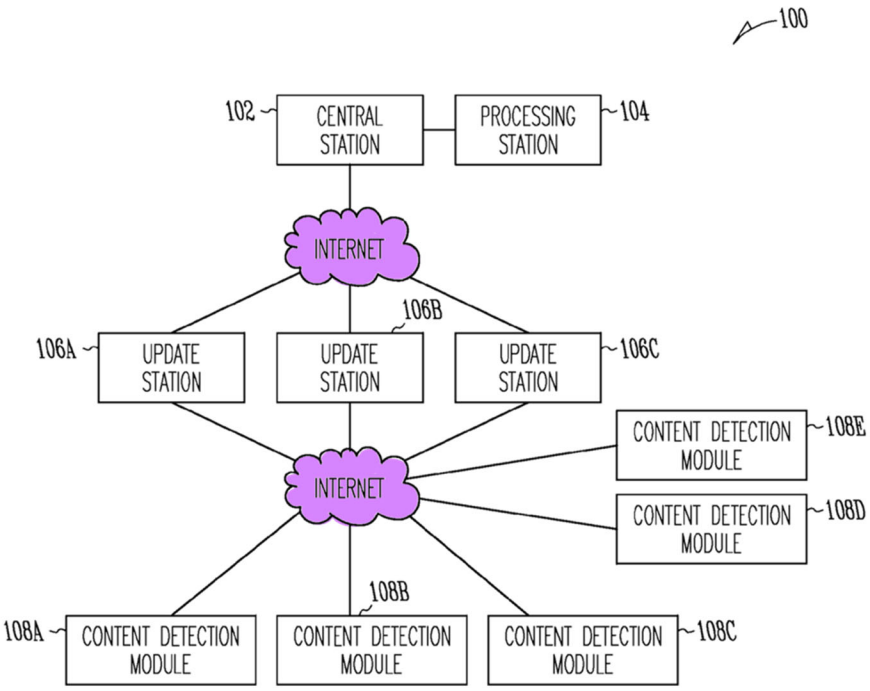


Fig. 1
(Annotated)

4. Claims 3, 6, and 9:

292. Fang disclosed the analysis processing facilitated an analysis that included receiving administrator input. (Fang, ¶34 ("Alternatively, the set of content data can be analyzed by an administrator, a separate device, or a separate software, and the result of the analysis is then input to processing station 104.").)

Netskope Exhibit 1002

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001; and further that such willful false statements may jeopardize the validity of the application or any patent issued thereon. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed on February 1, 2023, in Atlanta, GA.



EXPERT

Netskope Exhibit 1002

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 **10**
A. Theses 10
B. Published Journal Papers 10
C. Published Books and Parts of Books 11
D. Edited Proceedings and Books 11
E. Conference Presentations (refereed) 12
E.1 with Proceedings (refereed) 12
F. Workshop Presentations (refereed) 26
F.1 with Proceedings 26
G. Other 27
H. Research Proposals and Grants 29
I. Research Honors and Awards 36

III. SERVICE **38**
A. Professional Activities 38
A.1 Membership and Activities in Professional Societies 38
A.2 Journal Edit-oral Board Activities 38
A.3 Conference Committee Activities 38
B. On-campus Georgia Tech Committees 42

IV. NATIONAL AND INTERNATIONAL PROFESSIONAL RECOGNITION **44**
A. Patents 44

V. OTHER CONTRIBUTIONS **46**
A. Seminar Presentations 46

Netskope Exhibit 1002 - Appendix A

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) Georgia Institute of Technology	7/2015–7/2021
Director	Georgia Tech Information Security Center (GTISC) Georgia Institute of Technology	8/2012–6/2015
Professor	College of Computing Georgia Institute of Technology	2/2009–present
Associate Professor	College of Computing Georgia Institute of Technology	2/2005–2/2009
Assistant Professor	College of Computing Georgia Institute of Technology	8/2001–2/2005
Assistant Professor	Department of Computer Science North Carolina State University	8/1999–8/2001
Research Assistant	Department of Computer Science Columbia University	1994–1999
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-

Netskope Exhibit 1002 - Appendix A

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

Netskope Exhibit 1002 - Appendix A

I. TEACHING

A. Courses Taught

Semester/Year	Course	Number of Students	Comments
Fall 2021	OMS CS/Cybersecurity 6264 Information Security Labs: System and Network Defenses	25	Online
Fall 2021	OMS CS/Cybersecurity 6262 Network Security	360	Online
Fall 2021	OMS CS/Cybersecurity 6035 Introduction to Information Security	980	Online
Fall 2021	CS 6262 Network Security	40	At CoC
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	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
Fall 2020	CS 6262 Network Security	40	At CoC
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
Fall 2019	OMS CS/Cybersecurity 6262 Network Security	156	Online
Fall 2019	OMS CS/Cybersecurity 6035 Introduction to Information Security	770	Online
Fall 2019	CS 6262 Network Security	43	At CoC
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

Netskope Exhibit 1002 - Appendix A

Semester/Year	Course	Number of Students	Comments
Fall 2018	OMSCS 6262 Network Security	154	Online
Fall 2018	OMSCS 6035 Introduction to Information Security	525	Online
Fall 2018	CS 6262 Network Security	47	At CoC
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	OMSCS 6262 Network Security	300	Online
Fall 2017	OMSCS 6035 Introduction to Information Security	350	Online
Fall 2017	CS 4235/6035 Introduction to Information Security	120	At CoC
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	OMSCS 6262 Network Security	300	Online
Fall 2016	OMSCS 6035 Introduction to Information Security	350	Online
Fall 2016	CS 4235/6035 Introduction to Information Security	80	At CoC
Summer 2016	OMSCS 6035 Introduction to Information Security	350	Online
Spring 2016	OMSCS 6035 Introduction to Information Security	350	Online
Fall 2015	OMSCS 6035 Introduction to Information Security	230	Online
Fall 2015	CS 4235/6035 Introduction to Information Security	80	At CoC
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

Netskope Exhibit 1002 - Appendix A

Semester/Year	Course	Number of Students	Comments
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

Netskope Exhibit 1002 - Appendix A

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.

Netskope Exhibit 1002 - Appendix A

C. Individual Student Guidance

1. Postdoctoral Fellows Supervised

Erkam Uzun: 2021 -

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; Juniper Networks

Yian Huang: Graduated in Summer 2006; Google

Prahlad Fogla: Graduated in Summer 2007; 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; Director of Product & Application Security, Netflix

Monirul Sharif: Graduated in Fall 2010; Engineering Manager III (L7), Google

Kapil Singh: Graduated in Summer 2011; Research Satff Member, 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; Staff 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; Associate Professor with tenure in the College of Computer and Information Science, Northeastern University

Netskope Exhibit 1002 - Appendix A

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

Yacin Nadji: Graduated in Summer 2015; Corelight

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

Chengyu Song: Graduated in Summer 2016; Tenure-track Assistant Professor 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; Post-doc in EECS, UC Berkeley

Yang Ji: Graduated in Fall 2019; Research Scientist in Palo Alto Networks

Ruian Duan: Graduated in Fall 2019; Research Scientist in Palo Alto Networks

Erkam Uzun: Graduated in Summer 2021; Post-doc at Georgia Tech

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

Evan Downing: In Progress

Kyuhong Park: In Progress

Carter Yagemann: In Progress

Joey Allen: In Progress

Matthew Landen: In Progress

Burak Sahin: In Progress

Yongheng Chen: In Progress

Feng Xiao: In Progress

Haoran Wang: In Progress

Ding Zhang: In Progress

Zhang Yang: In Progress

Netskope Exhibit 1002 - Appendix A

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 and Wenke Lee. The Creation and Detection of Deepfakes: A Survey. *ACM Computing Surveys*. 54(1), 2021.

2. 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.

3. 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.

4. 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.

5. 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.

6. 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).

7. Prahlad Fogla and Wenke Lee. q-Gram Matching Using Tree Models. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 4 (April 2006).

8. 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.

9. 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.

Netskope Exhibit 1002 - Appendix A

10. 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.

11. 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.

12. 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.

13. 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.

14. 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.

Netskope Exhibit 1002 - Appendix A

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. 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.

2. Carter Yagemann, Mohammad Noureddine, 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.

3. 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.

4. 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.

5. 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.

6. 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.

7. 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).

8. 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).

9. 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).

Netskope Exhibit 1002 - Appendix A

10. 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.

11. 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.

12. 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.

13. 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.

14. 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.

15. 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.

16. 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.

17. 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.

18. 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.

19. 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.

Netskope Exhibit 1002 - Appendix A

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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

Netskope Exhibit 1002 - Appendix A

30. 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

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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**).

38. 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.

39. 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.

Netskope Exhibit 1002 - Appendix A

40. 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.

41. 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.

42. 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.

43. 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.

44. 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.

45. 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.

46. 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.

47. 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.

48. 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.

49. 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)

50. 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.

Netskope Exhibit 1002 - Appendix A

51. 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.

52. 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.

53. 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.

54. 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.

55. 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.

56. 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.

57. 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.

58. 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.

59. 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.

60. 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.

61. 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.

Netskope Exhibit 1002 - Appendix A

62. 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.

63. 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.

64. 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.

65. 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).

66. 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.

67. 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.

68. 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.

69. 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.

70. 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.

71. 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.

72. 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.

Netskope Exhibit 1002 - Appendix A

73. 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.

74. 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.

75. 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.

76. 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.

77. 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.

78. 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.

79. 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.

80. 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.

81. 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.

82. 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.

83. 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.

Netskope Exhibit 1002 - Appendix A

84. 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.

85. 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.

86. 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.

87. 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.

88. 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.

89. 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.

90. 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.

91. 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.

92. 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.

93. 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.

94. 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.

Netskope Exhibit 1002 - Appendix A

95. 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.

96. 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**)

97. 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.

98. 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.

99. 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.

100. 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.

101. 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.

102. 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.

103. 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.

104. 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.

105. 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.

Netskope Exhibit 1002 - Appendix A

106. 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.

107. 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.

108. 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.

109. 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.

110. 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.

111. 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.

112. 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.

113. 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.

114. 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.

115. 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.

116. David Cash, Yan Zong Ding, Yevgeniy Dodis, Wenke Lee, Richard Lipton, and Shabsi Walfish. 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.

Netskope Exhibit 1002 - Appendix A

117. 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.

118. 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.

119. 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.

120. 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.

121. 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.

122. 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.

123. 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.

124. 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.

125. 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.

126. 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.

127. 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.

Netskope Exhibit 1002 - Appendix A

128. 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.

129. 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.

130. 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.

131. 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.

132. 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.

133. 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.

134. 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.

135. 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.

136. 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.

137. 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.

138. 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.

Netskope Exhibit 1002 - Appendix A

139. 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.

140. 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.

141. 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.

142. 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.

143. Wenke Lee, Joao B.D. Cabrera, Ashley Thomas, Niranjan 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.

144. 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.

145. 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.

146. Wenke Lee, Sal Stolfo, Phil Chan, Eleazar Eskin, Wei Fan, Matt Miller, Shlomo Hershkop, 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.

147. 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.

148. 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.

149. 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.

150. 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.

Netskope Exhibit 1002 - Appendix A

151. 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.

152. 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.

153. 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.

154. 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.

155. 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.

156. 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.

157. 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.

158. 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.

Netskope Exhibit 1002 - Appendix A

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. <u>Other</u>

1. Matt Blaze, Sampath Kannan, Insup Lee, Oleg Sokolsky, Jonathan Smith, Angelos Keromytis, and Wenke Lee. Dynamic Trust Management. *IEEE Computer*, February 2009.

Netskope Exhibit 1002 - Appendix A

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.

Netskope Exhibit 1002 - Appendix A

H. Research Proposals and Grants

a. Approved and Funded

1. **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.

2. **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.

3. **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.

4. **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

5. **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

6. **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

7. **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.

Netskope Exhibit 1002 - Appendix A

8. **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.

9. **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.

10. **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.

11. **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.

12. **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.

13. **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.

14. **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.

15. **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

Netskope Exhibit 1002 - Appendix A

Foundation.
Funded at $1,873,078 for project total, with $673,078 to CoC/GT, 10/1/12–9/30/15.

16. **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.

17. **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.

18. **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.

19. **Dimensions of Network Intelligence**
HSARPA of DHS
Wenke Lee (PI) and Paul Royal (co-PI)
Funded at $1,022,000, 9/2012-9/2017.

20. **Dynamic Reputation for DNS**
Google
Wenke Lee (PI)
Funded as unrestricted gift at $75,000, 7/2010.

21. **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.

22. **Research in Virtual Machine Monitoring Techniques**
Sandia National Labs/Sandia Corp.
Wenke Lee (PI)
Funded at $50,000, 9/15/2010–9/15/2011.

23. **TC: SMALL: A Foundational and Practical Platform for Host Security Applications**
NSF CNS/Trustworthy Computing

Netskope Exhibit 1002 - Appendix A

Wenke Lee (PI)
Funded at $429,571, 7/15/2010–7/15/2013.

24. **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.

25. **Collaborative Research: CT-L: CLEANSE: Cross-Layer Large-Scale Efficient Analysis 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.

26. **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

27. **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.

28. **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)

29. **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)

Netskope Exhibit 1002 - Appendix A

30. **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)

31. **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)

32. **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)

33. **Cyber Threat Analytics**
    ARO via subcontract from SRI International
    Funded: $80,000, (7/15/2006-7/15/2007)

34. **ARL CTA**
    ARL via subcontract from SPARTA/Telcordia
    Funded: $75,000, (7/15/2006-7/15/2007)

35. **Computer Network Defenses Technologies**
    AFRL via subcontract from Scientific Systems Co. Funded: $30,000, (9/13/2006-3/13/2007)

36. **ARO Workshop on Research in Botnets**
    ARO
    Funded: $25,000, (4/1/2006-6/1/2007)

37. **Next-Generation Botnet Detection and Response**
    DARPA (via ARO)
    Funded: $291,346, (12/15/2005-6/14/07)

38. **An Information-Theoretic Framework for Evaluating and Optimizing Intrusion Detection Performance**
    Army Research Office
    Funded: $199,745, (4/1/2005-1/30/07)

39. **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)

Netskope Exhibit 1002 - Appendix A

40. **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)

41. **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)

42. **A Course Module on Wireless Security**
Microsoft, Curriculum Development Grant
Funded: $30,000 (unrestricted gift)

43. **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)

44. **High-Speed Intrusion Detection Sensors**
Army Research Lab (Scientific Systems Company Inc. subcontract)
Funded: $65,000 (8/7/2003-8/6/2005)

45. **Distributed Intrusion Detection Feasibility Study**
U.S. Government (Columbia University subcontract)
Funded: $45,000 (6/1/2003-6/30/2004)

46. **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)

47. **CAREER: Adaptive Intrusion Detection Systems**
NSF, CISE/CAREER
Funded: $350,000 (8/15/2002–7/31/2007)

48. **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)

49. **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

Netskope Exhibit 1002 - Appendix A

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)

50. **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)

Netskope Exhibit 1002 - Appendix A

I. Research Honors and Awards

1. IEEE Fellow, 2021.
2. 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.
3. 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.
4. ACM Fellow, 2017.
5. 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.
6. 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.
7. 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.
8. Outstanding Achievement in Research Program Development Award, (awarded to Mustaque Ahamad, Wenke Lee, Paul Royal, and David Dagon), Georgia Institute of Technology, 2015.
9. Outstanding Community Service Award, the IEEE Technical Committee on Security and Privacy, 2013
10. Outstanding Faculty Leadership for the Development of GRAs Award, Georgia Institute of Technology, 2012.
11. Sigma Xi Faculty Best Paper Award, Georgia Institute of Technology, 2010.
12. 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.
13. Outstanding Senior Faculty Research Award, College of Computing, Georgia Institute of Technology, 2009.
14. NSF CAREER Award, 2002.

Netskope Exhibit 1002 - Appendix A

15. 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).

16. 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.

17. 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.

Netskope Exhibit 1002 - Appendix A

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

Netskope Exhibit 1002 - Appendix A

18. The 22nd ACM Conference on Computer and Communications Security (CCS 2015).
19. Co-chair, Security and Privacy Track, WWW 2015.
20. Annual Network and Distributed System Security Symposium (NDSS) 2015.
21. European Symposium on Research in Computer Security (ESORICS) 2014.
22. The 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2014).
23. Security and Privacy Track, WWW 2014.
24. *PC Chair*, The 2013 IEEE Symposium on Security and Privacy.
25. The 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2013).
26. The 20th ACM Conference on Computer and Communications Security (CCS 2013).
27. The 16th International Symposium on Recent Advances in Intrusion Detection (RAID 2013).
28. The 19th ACM Conference on Computer and Communications Security (CCS 2012).
29. The 9th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DMIVA 2012).
30. The 7th USENIX Workshop on Hot Topics in Security (HotSec '12).
31. *PC co-Chair*, The 2012 IEEE Symposium on Security and Privacy.
32. Financial Cryptography and Data Security (FC 2012).
33. The 19th Annual Network and Distributed System Security Symposium (NDSS 2012).
34. The 8th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2011).
35. The 41st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2011).
36. The 20th USENIX Security Symposium (Security '11).
37. The 6th USENIX Workshop on Hot Topics in Security (HotSec '11).
38. The 2011 IEEE Symposium on Security and Privacy.
39. The 13th International Symposium on Recent Advances in Intrusion Detection (RAID 2010).
40. The 15th European Symposium on Research in Computer Security (ESORICS 2010).

Netskope Exhibit 1002 - Appendix A

41. The 7th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2010).
42. Security and Privacy Track, ICDCS 2010.
43. The 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2010).
44. Security and Privacy Track, WWW 2010.
45. *PC Chair*, The 17th Annual Network and Distributed System Security Symposium (NDSS 2010).
46. Financial Cryptography and Data Security (FC 2010).
47. The 25th Annual Computer Security Applications Conference (ACSAC 2009).
48. The 16th ACM Conference on Computer and Communications Security (CCS 2009).
49. 14th European Symposium on Research in Computer Security (ESORICS 2009).
50. The 18th USENIX Security Symposium (Security '09).
51. *PC Chair*, The Second USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), 2009.
52. *PC Co-Chair*, The Second ACM Conference on Wireless Network Security (WiSec), 2009.
53. The 2009 IEEE Symposium on Security and Privacy.
54. The 16th Annual Network and Distributed System Security Symposium (NDSS 2009).
55. The 24th Annual Computer Security Applications Conference (ACSAC 2008).
56. Internet Measurement Conference (IMC) 2008.
57. 13th European Symposium on Research in Computer Security (ESORICS 2008).
58. The 15th ACM Conference on Computer and Communications Security (CCS 2008).
59. The 17th USENIX Security Symposium (Security '08).
60. 3rd USENIX Workshop on Hot Topics in Security (HotSec '08).
61. The 2008 IEEE Symposium on Security and Privacy.
62. The 15th Annual Network and Distributed System Security Symposium (NDSS 2008).
63. The 23rd Annual Computer Security Applications Conference (ACSAC 2007).
64. The 14th ACM Conference on Computer and Communications Security (CCS 2007).
65. The 10th International Symposium on Recent Advances in Intrusion Detection (RAID 2007).

Netskope Exhibit 1002 - Appendix A

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

Netskope Exhibit 1002 - Appendix A

89. Steering Committee and the 6th International Symposium on Recent Advances in Intrusion Detection (RAID 2003).

90. The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003.

91. The 2003 IEEE Information Assurance Workshop.

92. Steering Committee for the 5th International Symposium on Recent Advances in Intrusion Detection (RAID 2002).

93. The 2002 ACM New Security Paradigms Workshop.

94. The 2002 IEEE Symposium on Security and Privacy.

95. *PC Co-Chair*, The 4th International Symposium on Recent Advances in Intrusion Detection (RAID 2001).

96. 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. 5-Year Review Committee for the Chair of School of Computer Science, College of Computing, 2016.

2. Faculty Recruiting Committee, School of Computer Science, 2014.

3. Faculty Recruiting Committee, School of Computer Science, 2012-13.

4. RPT Committee, College of Computing, 2010-2012.

5. College of Computing Dean Search Committee, 2009-2010, College of Computing.

6. Chair, Faculty Recruiting Committee, 2008-2009, School of Computer Science, College of Computing.

7. Chair Advisory Committee, 2008-2009, School of Computer Science, College of Computing.

8. Co-Chair, Faculty Recruiting Committee, 2007-2008, School of Computer Science, College of Computing.

9. RPT Executive Committee, 2007-2008, College of Computing.

10. Faculty Recruiting Committee, 2006-2007, College of Computing.

11. Faculty Recruiting Committee, 2005-2006, College of Computing.

12. Undergraduate Curriculum Task Force, 2004-2005, College of Computing.

13. Ph.D. Admissions Committee, 2003-2004, College of Computing.

Netskope Exhibit 1002 - Appendix A

14. Faculty Recruiting Committee, 2001-2003, College of Computing.

E. Consulting and Advisory Appointments

1. Advisory Board of the NUS-Singtel Cyber Security Research & Development Laboratory, National University of Singapore. 2019-2021.
2. Secure, Accessible & Fair Elections (SAFE) Commission, State of Georgia, 2018-2019
3. Advisory Board of the Faculty of Engineering, The Chinese University of Hong Kong. 2017-
4. Board of Trustees, Pace Academy. 2017-

Netskope Exhibit 1002 - Appendix A

IV. NATIONAL AND INTERNATIONAL PROFESSIONAL RECOGNITION

A. Patents

1. "Systems and methods for using video for user and message authentication". Simon Chung, Wenke Lee, and Yeongjin Jang. U.S. Patent Number: 10,476,888, November 2019.

2. "Method and System for Detecting Malware". Manos Antonakakis, Roberto Perdisci, Wenke Lee, and Gunter Ollmann. U.S. Patent Number: 10,257,212, April 2019.

3. "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.

4. "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.

5. "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.

6. "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.

7. "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.

8. "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.

9. "Method and System for Detecting Malware". Manos Antonakakis, Roberto Perdisci, Wenke Lee, and Gunter Ollmann. U.S. Patent Number: 9,525,699, December 2016.

10. "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.

11. "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.

12. "Systems and methods for secure in-VM monitoring", Monirul Sharif and Wenke Lee. U.S. Patent Number: 9,129,106, September 2015.

13. "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.

Netskope Exhibit 1002 - Appendix A

14. “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.

15. “Method and System for Detecting Malware”. Manos Antonakakis, Roberto Perdisci, Wenke Lee, and Gunter Ollmann. U.S. Patent Number: 8,578,497, November 2013.

16. “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.

17. “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.

18. “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.

Netskope Exhibit 1002 - Appendix A

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, Daejon, 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

Netskope Exhibit 1002 - Appendix A

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

Netskope Exhibit 1002 - Appendix A