

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

FORCEPOINT LLC,

Petitioner

v.

SECURITY PROFILING, LLC,

Patent Owner

IPR2023-00989

U.S. Patent No. 10,609,063

**DECLARATION OF A.L. NARASIMHA REDDY, PH.D.,
UNDER 37 C.F.R. § 1.68 IN SUPPORT OF PETITION
FOR *INTER PARTES* REVIEW**

Ex.1003

FORCEPOINT LLC. / Page 1 of 116

TABLE OF CONTENTS

	Page
I. INTRODUCTION	3
II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE	5
III. LEVEL OF ORDINARY SKILL IN THE ART	8
IV. RELEVANT LEGAL STANDARDS	9
V. OVERVIEW OF THE '063 PATENT	11
VI. CLAIM CONSTRUCTION	12
VII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE....	13
VIII. W-L RENDERS OBVIOUS CLAIMS 10, 11, 39, AND 58.....	13
A. Summary of W-L.....	13
B. Detailed Analysis of Claims.....	18
1. Claim 10.....	18
2. Claim 11	60
3. Claim 39.....	63
4. Claim 58.....	69
IX. THE COMBINATION OF GUPTA AND GRAHAM RENDERS OBVIOUS CLAIMS 1 AND 5.....	69
A. Summary of Gupta	69
B. Summary of Graham	71
C. Reasons to Combine Gupta with Graham.....	72
D. Similarity to US 8,984,644.....	73
E. Detailed Analysis of Claims.....	76
1. Claim 10.....	76
2. Claim 11	103
3. Claim 39	107
4. Claim 58.....	115
X. CONCLUSION.....	116

I, Narasimha Reddy, Ph.D., do hereby declare as follows:

I. INTRODUCTION

1. I am making this declaration at the request of Forcepoint LLC. in the matter of the *Inter Partes* Review of U.S. Patent No. 10,609,063 (“the ’063 patent”) to Oliphant et al.

2. I am being compensated for my work in this matter at my standard hourly rate of \$650. I am also being reimbursed for reasonable and customary expenses associated with my work and testimony in this proceeding. My compensation is not contingent on the outcome of this matter or the specifics of my testimony.

3. I have been asked to provide my opinions regarding whether the subject matter of claims 10, 11, 39 and 58 (“the Challenged Claims”) of the ’063 patent would have been obvious to a person having ordinary skill in the art (“POSITA”) at the time of the alleged invention, in light of the prior art. It is my opinion that the Challenged Claims would have been obvious to a POSITA.

4. In the preparation of this declaration, I have studied:

- a. the ’063 patent, Ex.1001;
- b. the prosecution history of the ’063 patent (“’063 File History”), Ex.1002;

Ex.1003

FORCEPOINT LLC/ Page 3 of 116

c. U.S. Patent No. 7,359,962 to Willebeek-LeMair et al. (“W-L”),
Ex.1005;

d. U.S. Patent Publication No. 2003/0004689 to Gupta et al.
 (“Gupta”), Ex.1006; and

e. U.S. Patent No. 7,237,264 to Graham et al. (“Graham”), Ex.
1007.

5. In forming the opinions expressed below, I have considered: the documents listed above; the relevant legal standards, including the standard for obviousness; and my own knowledge and experience based upon my work in the field of network communications and security as described below, as well as portions of the following additional materials:

a. the prosecution history of U.S. Patent No. 9,117,069 (“’069 File History”), great-grandparent of the ’063 patent, Ex.1008;

b. the prosecution history of U.S. Patent No. 9,100,431 (“’431 File History”), grandparent of the ’063 patent, Ex.1009;

c. the prosecution history of U.S. Patent No. 10,050,988 (“’988 File History”), parent of the ’063 patent, Ex.1010;

d. U.S. 6,493,871 to McGuire et al. (“Mcguire”), Ex.1014;

e. U.S. Pub. 2003/0084340 to Schertz et al. (“Schertz”), Ex.1016;

Ex.1003

f. U.S. Patent No. 6,735,766 to Chamberlain et al.

(“Chamberlain”), Ex.1018; and

g. U.S. 8,205,161 to King et al (“King”), Ex.1020.

6. Unless otherwise noted, all **emphasis** in any quoted material has been added. Claim terms are *italicized*.

II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE

7. My complete qualifications and professional experience are described in my *Curriculum Vitae*, a copy of which can be found in Exhibit 1004. The following is a brief summary of my relevant qualifications and professional experience.

8. I am currently the J.W. Runyon Professor of Electrical and Computer Engineering at Texas A&M University in College Station, Texas. I have over 25 years of experience in a wide variety of technologies and industries relating to data communications, storage systems, and distributed systems, including packet-switched network communications.

9. My academic credentials include a Bachelor of Technology degree in Electronics and Electrical Communications Engineering from the Indian Institute of Technology, Kharagpur, India, in August 1985. I then received a Master of Science and a Ph.D. in Computer Engineering from the University of Illinois at Urbana-Champaign in May 1987 and August 1990, respectively.

Ex.1003

FORCEPOINT LLC/ Page 5 of 116

10. I have worked for over 25 years in the field of Electrical Engineering. My primary focus and research interest has been on computer networks, storage systems, multimedia systems, and computer architecture. I have authored and co-authored over a hundred technical papers and several book chapters related to several of these interests, including on such topics as multipath routing, route control, high-speed networks, network congestion, packet management, quality of service regulation, network security, network modeling, differentiated services, storage system enhancements, caching strategies, and multimedia system enhancements to name a few examples. I am listed as an inventor on five U.S. patents in the field of communication networks.

11. My employment history following my graduation from the University of Illinois at Urbana-Champaign began at the IBM Almaden Research Center in San Jose, California in 1990. At IBM, I worked on projects related to disk arrays, multiprocessor communication, hierarchical storage systems and video servers.

12. In 1995, I joined the faculty of the department of Electrical Engineering at Texas A&M University initially as an Associate Professor and was later promoted to full, tenured Professor. At Texas A&M, I was the Associate Agency Director for Strategic Initiatives and Centers for the Texas A&M Engineering Experiment Station (TEES) during 2016-21, which engages in engineering and technology-oriented research and educational collaborations.

Further, I served as Associate Dean for Research for College of Engineering during 2016-21.

13. At Texas A&M, I have taught dozens of courses related to computer networking and communications, as well as computer architecture, multimedia systems and networks, topics in networking security, multimedia storage and delivery, as well as networking for multimedia applications. I have done research on various topics of network security including anomaly detection, botnet detection, building mechanisms in routers and switches to improve security, spammer detection in social networks and improving security of cyberphysical systems. I have also served on various committees for the benefit of the scientific community and the Texas A&M University community.

14. I am a member of a number of professional societies, including the Institute of Electrical and Electronic Engineers (IEEE), where I have been elevated to an IEEE Fellow, and the Association for Computing Machinery (ACM). I have been responsible for chairing or co-chairing numerous conferences and programs, as well as presenting research at major IEEE and ACM conferences. For example, I served as program co-chair for the 2008 5th International Conference on Broadband Communications, Networks and Systems, panels co-chair for the 2008 3rd International Conference on Communication Systems Software & Middleware,

and panel chair of the IEEE Conference of High Performance Computer Architecture.

15. My presentations include a Keynote speech at International Conference on Information Technology-New Generations in 2013, a Keynote speech at IEEE International Symposium on Computers and Communications 2010, and several invited talks including at Georgia Tech (2013), COMSNETS Conference (2013), Int. Conf. on Networking and Communications (2012), Samsung (2011), Korea University (2011), Aijou University (2011), Catedra Series at University of Carlos III, Madrid (2009), Thomson Research, Paris (2009), Telefonica Research, Barcelona (2009), and a Distinguished Seminar at IBM Austin Research Lab (2008).

16. I have received multiple awards in the field of networks and computer architecture. I received the NSF Career Award from 1996-2000. I received an outstanding professor award by the IEEE student branch at Texas A&M during 1997-1998, an outstanding faculty award by the department of Electrical and Computer Engineering during 2003-2004, a Distinguished Achievement award for teaching from the former students' association of Texas A&M University, and a citation "for one of the most influential papers from the 1st ACM Multimedia conference."

III. LEVEL OF ORDINARY SKILL IN THE ART

17. I understand there are multiple factors relevant to determining the level of ordinary skill in the pertinent art, including (1) the levels of education and experience of persons working in the field at the time of the invention; (2) the sophistication of the technology; (3) the types of problems encountered in the field; and (4) the prior art solutions to those problems.

18. A Person of Ordinary Skill in The Art (“POSITA”) in July 2003 would have had (i) a working knowledge of the network communications art pertinent to the ’063 patent, including network security and (ii) (a) a bachelor’s degree in computer science, computer engineering, or an equivalent and two years of professional experience relating to network communications, (b) a higher relevant level of education (e.g., a Master’s degree) with less professional experience or, (c) more professional experience and less education.

19. For purposes of this Declaration, in general, and unless otherwise noted, my statements and opinions, such as those regarding my own experience and what a POSITA would have understood or known generally (and specifically related to the references I consulted herein), reflect the knowledge that existed in the relevant field as of the priority date of the ’063 patent.

IV. RELEVANT LEGAL STANDARDS

20. I am not an attorney. In preparing and expressing my opinions and considering the subject matter of the '063 patent, I am relying on certain basic legal principles that Forcepoint's counsel has explained to me.

21. I understand that prior art to the '063 patent includes patents and printed publications in the relevant art that predate the priority date of the '063 patent. For purposes of this Declaration, I am applying July 1, 2003, as the priority date of the '063 patent.

22. I have been informed by Forcepoint's counsel that a claimed invention is unpatentable under 35 U.S.C. § 103 if the differences between the claimed invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a POSITA. I have also been informed by Forcepoint's counsel that the obviousness analysis considers factual inquiries, including the level of ordinary skill in the art, the scope and content of the prior art, and the differences between the prior art and the claimed subject matter.

23. I have been further informed by Forcepoint's counsel that there are several recognized rationales for combining references or modifying a reference to show obviousness. These rationales include: (a) combining prior art elements according to known methods to yield predictable results; (b) simple substitution of one known element for another to obtain predictable results; (c) use of a known

technique to improve a similar device (method, or product) in the same way; (d) applying a known technique to a known device (method, or product) ready for improvement to yield predictable results; (e) choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; and (f) some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify the prior art or to combine prior art teachings to arrive at the claimed invention.

V. OVERVIEW OF THE '063 PATENT

24. The '063 patent focuses on the “management of security of computing and network devices that are connected to other such devices.” '063 patent at 1:23-25. To accomplish this, “technology is provided to meet the following market requirements: integrate network security products to share information; provide system intelligence; and remediate network vulnerabilities.” '063 patent at 11:37-40. Integrating products is done “to intelligently reference and share information from the same vulnerability data set.” '063 patent at 11:41-43.

25. The system in the '063 patent has a “security server 135” which “collects certain information and provides certain data services.” '063 patent at 2:30-38. The server collects data about devices within the network, including operating system information and other configuration and policy settings. '063 patent at 2:43-45. The system in the '063 patent “may easily integrate with and

enable network security products such as IDS [intrusion detection system], scanners, or firewalls to intelligently reference and share the same vulnerability data set.” ’063 patent at 8:26-33.

26. The server obtains a list of security vulnerabilities from a vulnerability remediation database 110. ’063 patent at 2:46-49. The server also includes a database 146, which maintains remediation techniques for vulnerabilities. ’063 patent at 2:56-57, 4:62-64. When traffic arrives at the network, such as at a firewall 131, the security server determines whether the traffic “is attempting to take advantage of a particular known vulnerability” by using the information from the database. ’063 patent at 4:9-11, 4:21-29. If an attack is detected, the security server “selects one or more remediation techniques from database 146 that remediate the particular vulnerability.” ’063 patent at 4:62-64.

27. As I will explain below, these concepts were well known as of the priority date of the ’063 patent.

VI. CLAIM CONSTRUCTION

28. It is my understanding that in order to properly evaluate the ’063 patent, the terms of the claims must first be interpreted. It is my understanding that for the purposes of this *inter partes* review, the claims are to be construed under the so-called *Phillips* standard, under which claim terms are given their ordinary and customary meaning as would have been understood by a POSITA in light of

the specification and prosecution history, unless the inventor has set forth a special meaning for a term. I have also been informed that claim terms only need to be construed to the extent necessary to resolve the obviousness inquiry. I have reviewed the entirety of the '063 patent, as well as its prosecution history. It is my opinion that for purposes of applying the prior art presented herein to evaluate the patentability of the Challenged Claims, no terms require requires express construction.

VII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

29. The discussion in this Declaration provides a detailed analysis of how the asserted prior art references teach each limitation of the Challenged Claims.

30. As part of my analysis, I have considered, and discuss in detail, the scope and content of the prior art and any differences between the alleged invention and the prior art.

31. It is my opinion that the alleged invention recited in the Challenged Claims would have been obvious in view of the teachings of the asserted prior art and the knowledge of a POSITA.

VIII. W-L RENDERS OBVIOUS CLAIMS 10, 11, 39, AND 58

A. Summary of W-L

32. U.S. Patent No. 7,359,962 to Willebeek-LeMair et al. (Ex.1005, “W-L”) was filed April 30, 2002, and issued April 15, 2008. W-L is titled “Network Security System Integration.”

33. W-L was first cited by Applicant in the great-great-grandparent application, in what would become the '069 patent, after the issue fee was paid with a petition to withdraw from issuance, along with 27 other patents, 23 patent publications, and 11 non-patent literature citations in June 2015. See '069 File History, pp. 12-18. The Examiner mailed a new notice of allowance with an Examiner's Amendment in July 2015. '069 File History, pp. 28-85 (see also p. 84, commenting that no reason for allowance is necessary). W-L was cited in the '431 patent on December 4, 2014, along with 794 other patent/patent publication prior art (and 17 non-patent literature references) during prosecution of the '431 patent (before a different Examiner). Compare '069 File History, pp. 84, 93-100 (Examiner Dant Shaifer Harriman) to '431 File History, pp. 481-514 (IDS filing), 546-579 (IDS considered by Examiner Madhuri Herzog).¹ W-L was also cited in the '063 patent file history by the Examiner as pertinent to applicant's disclosure

¹ I have also reviewed the file histories of U.S. Pat. No. 8,266,699 and U.S. App. Nos. 14/499,226, 14/499,227, and 14/499,239, and, to best of my knowledge, neither W-L, Gupta, nor Graham was substantively discussed in any of these file histories.

but not relied upon. '063 File History, p. 495. Applicant did not address this in any subsequent response.

34. Like the '063 patent, W-L “relates to network security” that integrates functionality “to provide for a unified network defense structure.” W-L at 1:7-10. This includes integrating “the functionalities performed by a firewall, IDS and VAS [vulnerability assessment scanner] for network security into one system or appliance supported on a single platform.” W-L at 3:14-18. W-L’s unified system 10 is illustrated in Figure 1, and an “exemplary integrated architecture” of W-L’s unified system 10 is illustrated in Figure 2, W-L at 4:37-39.

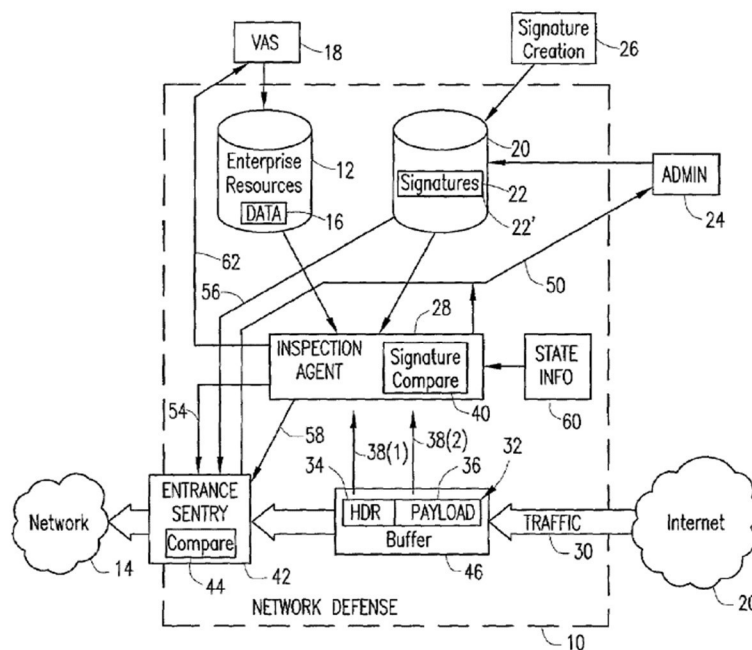


FIG. 1

W-L at FIG. 1

35. The unified system includes “an enterprise resource database” that contains data that identifies “machines (hosts) in the network, services provided by the hosts, and potential computer system and network device vulnerabilities associated with those machines and services in the context of the network configuration.” W-L at 5:9-15. The system also includes a “signature database” that stores “detection signatures” that comprise “security rules, policies and algorithms” designed to mitigate or avert network damage from vulnerabilities. W-L at 5:20-24. For example, a detection signature includes, among other objects, an “action set” which is a “definition of the action or actions” for the system to perform if a threat is detected. W-L at 10:58-60.

36. The system also includes an “agent 126 that functions to configure, tune and monitor the operation of the intrusion detector functionality 116 and the firewalling functionality 118,” as shown in FIG. 2, reproduced below. W-L at 9:36-41.

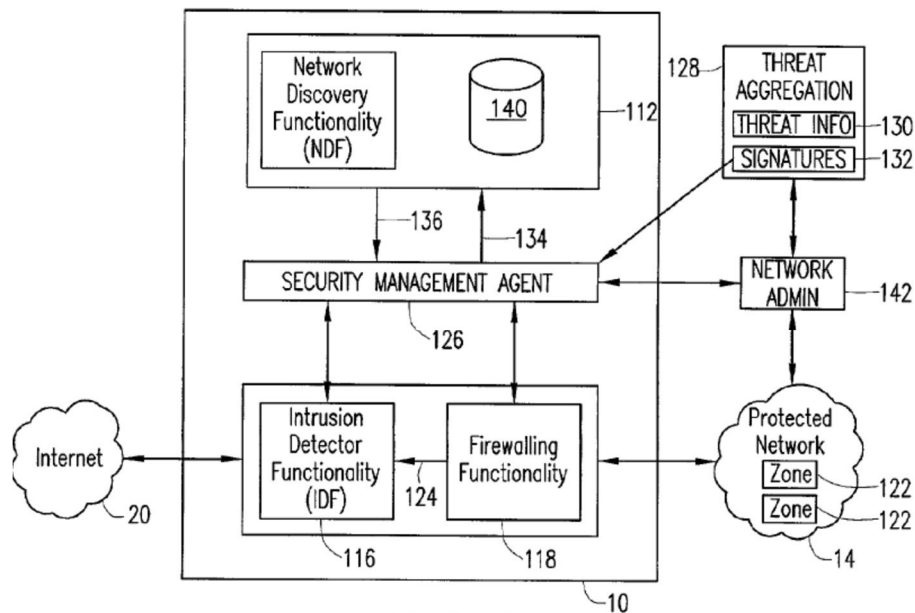


FIG. 2

W-L at FIG. 2

37. Tuning involves the agent accessing the enterprise database and tailoring the relevant detection signature(s) “based on enterprise specific data” to minimizing the chance of false positives. W-L at 10:3-19. The agent instantiates the tailored detection signature (which has been determined to be relevant to at least some part of the network) at the intrusion detector functionality and/or the firewalling functionality “to be sensitive to the specific recognized vulnerabilities of the network 14 being protected.” W-L at 11:40-54.

38. With the detection signatures so instantiated, in operation if either (or both) of the intrusion detector functionality and the firewalling functionality “subsequently detects traffic that matches the criteria of the detection signature,”

then “the threat response actions defined by that signature are then invoked” in order to mitigate the effect of any attack. W-L at 13:35-39. Some examples of actions taken include logging or blocking traffic, generating an alert, terminating a session, and so forth. W-L at 13:40-42.

B. Detailed Analysis of Claims

1. Claim 10

- a. **[10.0] *A non-transitory computer-readable media storing instructions that, when executed by one or more processors, cause the one or more processors to:***

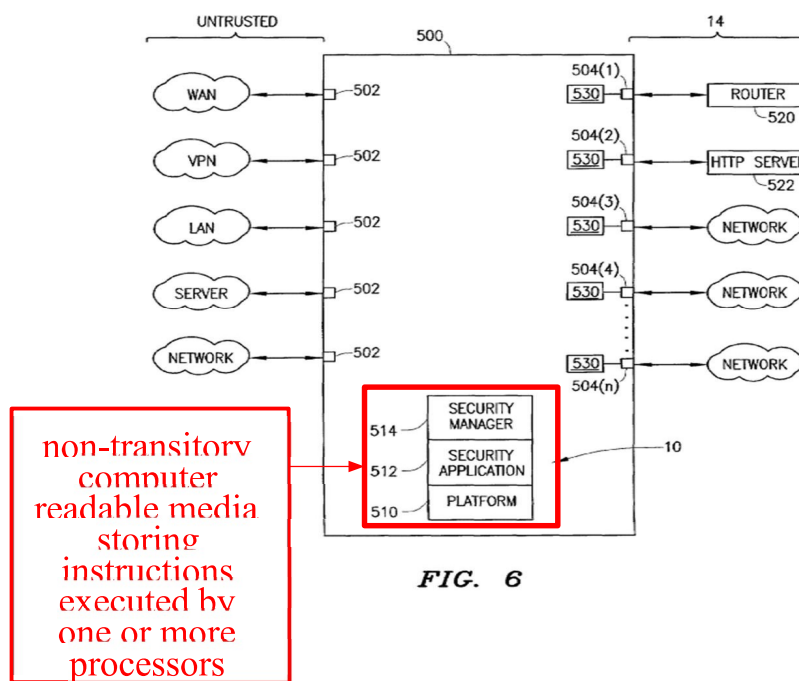
39. To the extent that the preamble is limiting, W-L renders it obvious.

40. W-L discloses a “*non-transitory computer-readable media storing instructions*” and “*one or more processors*” executing the instructions by disclosing its appliance 500.

41. W-L teaches using an appliance with underlying hardware, operating system (software), and other facilities required to perform the operations of its disclosure. “The appliance **500** includes a platform **510** supporting its operation. **The platform 510 comprises the underlying hardware, operation system and core infrastructure facilities necessary to allow the appliance 500 to provide an execution environment for security application.**” W-L at 16:1-5.

42. W-L teaches that the “appliance 500 further includes a security application functionality 512 that executes on the platform and which, in the

preferred embodiment is implemented as the unified network defense system 10 shown in FIGS. 1 and 2 and described in detail herein.” W-L at 16:11-15. W-L continues that it “comprises the processes and functions necessary to have the platform 510 function as a network security appliance 500 as opposed to a generic network platform.” W-L at 16:15-19; see also FIG. 6:



W-L at FIG. 6 (annotated).

43. In other words, W-L teaches that the platform 510 includes the “operating system ... necessary” – as well as the “underlying hardware” – to perform its operations in support of the “security application functionality 512.” W-L at 16:2-5. It would have therefore been obvious to a POSITA that the platform 510’s “underlying hardware” would include one or more “processors” to execute the security application functionality 512. *See, e.g.,* Chamberlain at

4:20-43 (showing that “multiprocessor systems” and “processing unit 21” connected to the system memory were known).

44. Further, the security application functionality 512 executes on that platform and is “implemented as the unified network defense system 10.” W-L at 16:11-15. Therefore, it would have been obvious to a POSITA that the “security application functionality 512” that W-L discloses is embodied on the “platform 510” as the non-transitory computer readable medium, further since it was well-known and common for executable applications to be stored on non-transitory computer readable media.² See, e.g., Chamberlain at Abstract (computer-readable medium). Moreover, a POSITA would have recognized that the “operating system” of the platform 510 would have been stored on a non-transitory computer readable media – and, as such, that the platform 510 has a non-transitory computer readable media.

45. Therefore, W-L renders obvious “*a non-transitory computer-readable media storing instructions that, when executed by one or more processors, cause the one or more processors to,*” as recited in the preamble.

b. [10.1] receive first vulnerability information from at least one first data storage that is generated utilizing second

² As the “security application functionality 512 ... is implemented as the unified network defense system 10 shown in FIGS. 1 and 2,” W-L at 16:11-15, the discussion of network defense system 10 herein applies to security application functionality 512 in Figure 5.

vulnerability information from at least one second data storage that is used to identify a plurality of potential vulnerabilities;

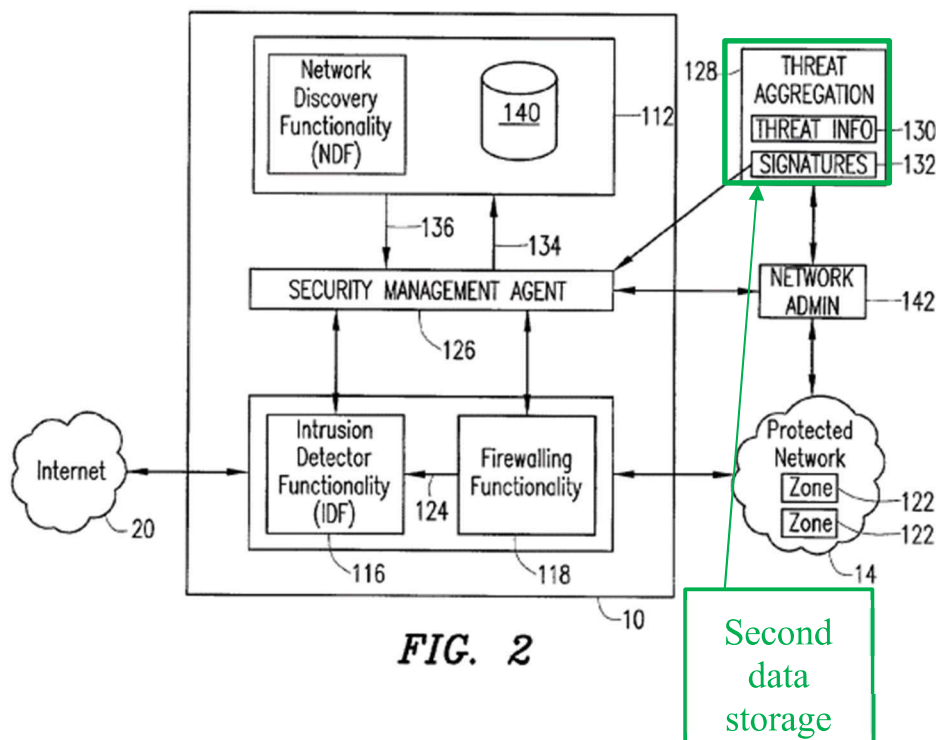
46. W-L renders obvious this limitation. Note that claim element [10.1] is addressed according to the temporal sequence of operations listed, which starts with the “*second data storage*,” rather than according to the sequence suggested by the nomenclature of the claim (i.e., “*first*” followed by “*second*”). Also, as shown below, claim element [10.1] is rendered obvious in two different ways: (1) by the embodiment illustrated in Figure 2 of W-L along with the associated description; and (2) by the embodiment illustrated in Figure 1 of W-L along with the associated description.

47. First, W-L teaches a “*second data storage*” that is used to “*identify a plurality of potential vulnerabilities.*” W-L does so with (1) threat aggregation functionality 128 (of Figure 2) and the information it stores; and (2) entity 26 (of Figure 1) and the information it stores.

48. W-L discloses a “threat aggregation functionality 128 [that] **stores threat information 130** (for example worm, virus, trojan, DoS, Access, Failure, Reconnaissance, other suspicious traffic, and the like) collected from around the world.” W-L at 10:36-40. W-L teaches that this “threat information” is “analyzed and utilized by the network administrator 142 to design the detection signatures 132.” W-L at 10:40-42. The “threat information 130” is one example of “*second*

vulnerability information” that is stored by “threat aggregation functionality 128”
(“*at least one second data storage*”).

49. The “detection signatures 132” are another example of “*second vulnerability information.*” These “detection signatures 132,” also stored by the threat aggregation functionality 128, include “security rules, policies and algorithms ... that can be used by the system 10 to mitigate or avert network damage from the collected threats (see, also, signatures 22 and database 20 of FIG. 1).” W-L at 10:42-46.



W-L at FIG. 2 (annotated).

50. Moreover, the “threat information 130” or the “detection signatures 132,” both stored in the threat aggregation functionality 128, are “*used to identify a*

Ex.1003

plurality of potential vulnerabilities.” As W-L notes, “[b]efore the detection signature 132 ... is installed in the intrusion detector functionality 116 and/or firewalling functionality 118, the agent 126 may first query 134 the network discovery functionality 112” and evaluate “for the purpose of determining whether the detection signature 132 is relevant to the particular network 14 being protected.” W-L at 11:12-29. Since it is unknown whether the detection signature 132 (and, by extension, the threat information 130 on which the detection signature is based) before evaluation, it would have been obvious to a POSITA that the information thus stored in the threat aggregation functionality 128 identifies *potential vulnerabilities*.

51. The example describing FIG. 1 also discloses the “*second vulnerability information*” with its description of entity 26. Signatures stored in database 20 of FIG. 1

may be obtained from any one of a number of well known sources, including, for example, machine (host) manufacturers, service suppliers, the Internet, and the like. Additionally, the signatures 22 may be created by an administrator 24 of the protected network 14. Still further, the signatures 22 may be supplied by a entity 26 in the business of signature creation, where that entity operates to collect threat information (for example, worm, virus, trojan, DoS, Access, Failure, Reconnaissance, other

suspicious traffic, and the like) from around the world, analyze that information and design detection signatures 22 that can be used by others to mitigate or avert network damage from the collected threats.

W-L at 5:24-36.

52. The signatures 22, before being stored in database 20, are signatures that have been created with respect to “*potential vulnerabilities*” because they have not yet taken into account the “detected vulnerabilities” of the network 14 itself. It would have been obvious to a POSITA that the system 10 would obtain those signatures from a data storage storing “*a plurality of potential vulnerabilities*” before storing the relevant signatures in database 20.

including “instantiat[ing] detection signatures ... that are tailored to a certain perceived threat to the network 14.” W-L at 9:37-48. The tuning includes the “agent 126 confer[ring] with the network discovery functionality 112 to ensure that the detection signatures supplied to the intrusion detector functionality 116 and firewalling functionality 118 are tailored to the collected enterprise (i.e., network 14) specific data.” W-L at 10:5-9. To this end, “the enterprise specific data is considered by the agent when issuing a detection signature so that the signature (either through its criteria or its response instructions) is designed in a way that minimizes the likelihood that false positive alarms will be generated.” W-L at 10:9-14.

55. W-L’s tailored signatures teach “*first vulnerability information.*” These signatures are generated from the signatures obtained from the threat aggregation functionality, as tailored by the enterprise specific data. Therefore, they are “*generated utilizing second vulnerability information.*” Moreover, it would have been obvious to a POSITA that the tailored signatures would have been stored by the agent 126 at least temporarily (e.g., in a first data storage). As W-L teaches, the agent 126 evaluates data 136 retrieved from the network discovery functionality 112, after receiving the detection signature from the threat aggregation functionality, “for the purpose of determining whether the detection signature 132 is relevant.” W-L at 11:11-29; *see also* 13:8-11 and 13:27-35

(discussing agent 126 tailoring a signature received from database 132). Thus, it would have been obvious for agent 126 to retain (and thus *store*) detection signatures that are determined relevant.

56. W-L also teaches that the platform 510 includes the “underlying hardware” necessary to perform its operations in support of the “security application functionality 512,” which includes the agent 126. W-L at 16:2-5. It would have therefore been obvious to a POSITA that the platform 510’s “underlying hardware” would include a data storage to store the detection signature 132 while evaluating the data 136 for relevance and tailoring them to enterprise specific data. Thus, W-L teaches a “*first data storage*” for the “*first vulnerability information*” that is “*generated utilizing second vulnerability information.*” See W-L at FIGs. 2 and 6:

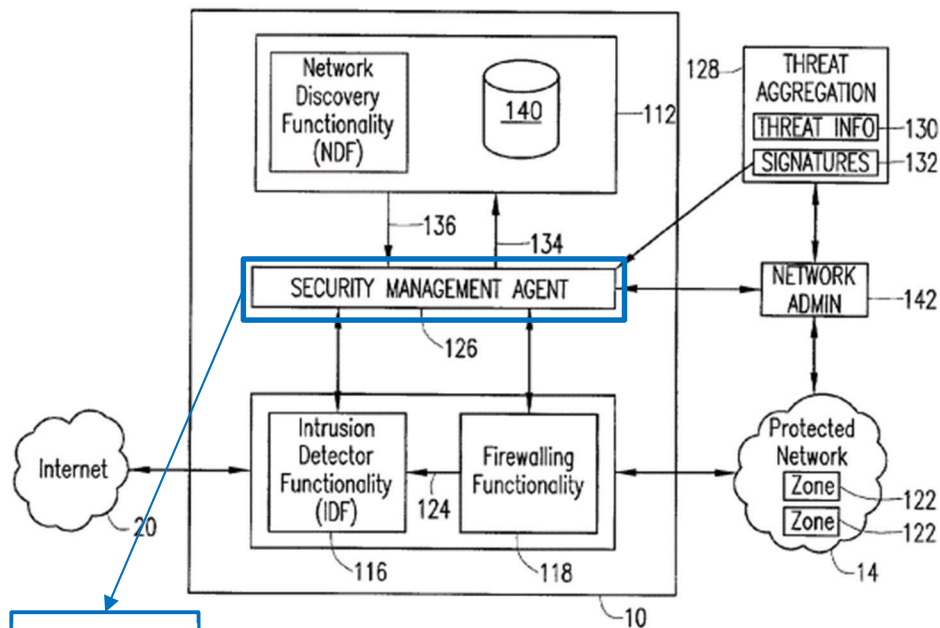


FIG. 2

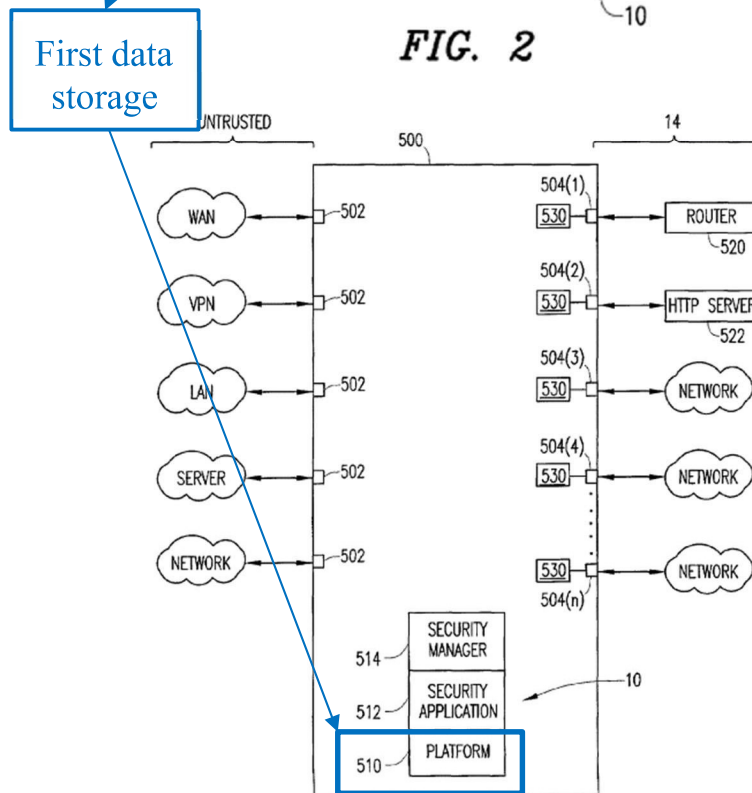


FIG. 6

W-L at FIGs. 2 and 6 (annotated).

Ex.1003

Database 20 (FIG. 1)

57. W-L further teaches generating the “*first vulnerability information*” with the database 20 in the FIG. 1 example. The signature database 20 “stores detection signatures 22 (comprising, for example, security rules, policies and algorithms) **that are designed to mitigate or avert network damage from detected vulnerabilities.**” W-L at 5:20-24. W-L further teaches that the signatures 22 thus stored in the database 20

may be obtained from any one of a number of well known sources, including, for example, machine (host) manufacturers, service suppliers, the Internet, and the like. Additionally, the signatures 22 may be created by an administrator 24 of the protected network 14. Still further, the **signatures 22** may be **supplied by a entity 26** in the business of signature creation, where that entity operates to collect threat information (for example, worm, virus, trojan, DoS, Access, Failure, Reconnaissance, other suspicious traffic, and the like) from around the world, analyze that information and design detection signatures 22 that can be used by others to mitigate or avert network damage from the collected threats.”

W-L at 5:24-36.

58. It would have been obvious to a POSITA that a detection signature designed to mitigate damage from “detected vulnerabilities” is an example of “*first*

vulnerability information” (the detection signature 22 in database 20) “*that is generated utilizing second vulnerability information*” (all available detection signatures from known sources, administrator-created, or a business entity 26).

This is at least because the signatures stored in the database 20 are only those that are for “detected vulnerabilities,” not just any “threat information ... from around the world” in general. In other words, the signatures 22 of FIG. 1 are generated from the signatures obtained from the external, “around the world” source(s). They are then stored in the database 20 for the system 10. Thus, W-L teaches a “*first data storage*” for the “*first vulnerability information*” that is “*generated utilizing second vulnerability information.*”

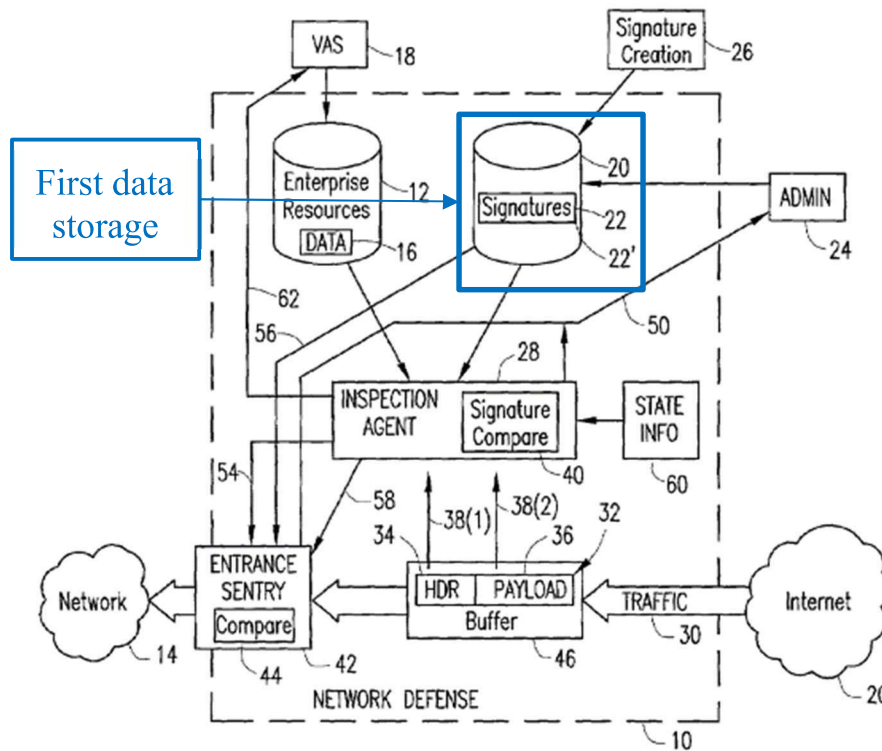


FIG. 1

W-L at FIG. 1 (annotated).

59. Third, W-L teaches receiving “*first vulnerability information*” (detection signatures) from the “*first data storage*” (either the storage of platform 510 supporting agent 126 or the database 20).

Receiving from agent 126 (FIG. 2)

60. As a first example, W-L teaches the intrusion detector functionality, alone or together with firewalling functionality, receiving the tailored signatures from the agent 126. After the detection signatures are supplied to the agent 126, and tailored based on enterprise data, W-L teaches that the tailored detection

signature is “supplied to the intrusion detector functionality 116 and/or firewalling functionality 118 to effectuate the tuning of the system 10 against a certain perceived threat by filtering of the packets (traffic).” W-L at 11:1-10; *see also* 11:11-29.

61. Receipt of the tailored signatures at either of the intrusion detector functionality 116 and the firewalling functionality 118 from the agent 126, which at least temporarily stores signatures while evaluating and tailoring them, renders obvious receiving “*first vulnerability information*” from a “*first data storage*,” where the signatures are generated from information received from the threat aggregation functionality 128.

Receiving from database 20 (FIG. 1)

62. As a second example, W-L teaches the agent 28 of FIG. 1 receiving detection signatures 22 from database 20. “The inspection operation performed by **the inspection agent 28** next involves comparing 40 the extracted packet features against **the detection signatures 22 obtained from the signature database 20.**” W-L at 5:50-53. “The detection signatures 22 may be applied by the inspection agent 28 **as they are obtained from the database 20** (i.e., without alteration or change).” W-L at 6:5-7.

63. W-L further teaches another example with the agent 28 instantiating detection signatures 22 at the “comparison functionality 40 and/or the sentry’s comparison functionality 44.” W-L at 8:7-11. In this protection mode, the signatures are downloaded to one or both of the agent 28 and “entrance sentry 42” for use in comparing against traffic. W-L at 5:50-53 (signatures obtained from database), 6:54-58 (signatures downloaded to entrance sentry 42 via agent 28 or directly from database 20).

64. Receipt of the signatures at either of the agent 28 and the entrance sentry 42 from the database 20 (either directly or indirectly), renders obvious receiving “*first vulnerability information*” from a “*first data storage*,” where the signatures are generated from the entity 26.

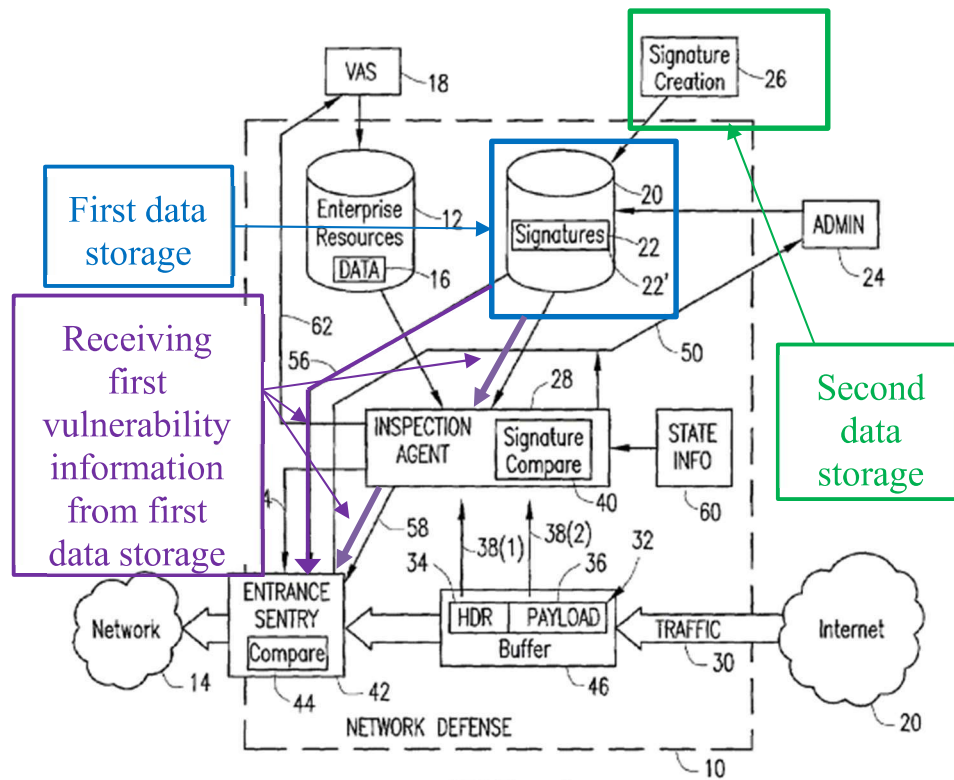


FIG. 1

W-L at FIG. 1 (annotated)

65. W-L therefore renders obvious “receive first vulnerability information from at least one first data storage that is generated utilizing second vulnerability information from at least one second data storage that is used to identify a plurality of potential vulnerabilities,” as claimed.

c. [10.2] said first vulnerability information generated utilizing the second vulnerability information, by:

66. W-L renders this limitation obvious. As already explained at [10.1], W-L renders obvious “first vulnerability information ... that is generated utilizing second vulnerability information.” W-L further renders obvious the “first

vulnerability information” being generated as demonstrated at [10.3] and [10.4] below.

- d. [10.3] *identifying at least one configuration associated with a plurality of devices including a first device, a second device, and a third device, and*

67. W-L renders obvious this limitation.

68. First, W-L discloses “*identifying at least one configuration*” by disclosing determining an operating system configuration (of an environment). As noted at [10.1], W-L teaches checking the conditions of the network, as obtained by the enterprise specific data.

The tuning operation that is performed by the agent 126 further occurs in a manner that is optimized to the network 14 being protected. **To that end, the agent 126 confers with the network discovery functionality 112 to ensure that the detection signatures supplied to the intrusion detector functionality 116 and firewalling functionality 118 are tailored to the collected enterprise (i.e., network 14) specific data.** More specifically, **the enterprise specific data is considered by the agent when issuing a detection signature so that the signature** (either through its criteria or its response instructions) is designed in a way that minimizes the likelihood that false positive alarms will be generated. This cooperative defense effort between the network discovery

functionality 112, intrusion detector functionality 116 and firewalling functionality 118, **through the agent 126 tailoring the detection signatures based on enterprise specific data**, effectuates a self-hardening system 10.

W-L at 10:3-19. A POSITA would have recognized that an obvious example of a false positive alarm would be an alarm based on a signature that corresponds to a vulnerability that does not apply to any machine in the network.

69. This includes determining an operating system configuration in some examples:

As another example illustrating the operation of the agent 126, assume that the network 14 includes a Linux host running an Apache web server.³ The threat aggregation functionality 128 collects and stores a piece of threat information 130 relating to a network vulnerability relating the malicious code intended to exploit a Microsoft IIS web server running on a Microsoft operating system. The network administrator 142 may design a detection signature 132 for this particular threat. The detection signature 132 is then provided to the agent 126. **The agent, noting the query object specification of Microsoft IIS web servers and Microsoft operating**

³ Note this sentence likely includes typographical errors. What it likely meant to say is that “assume that the network 14 includes a Microsoft operating system ~~Linux~~ host running a Microsoft IIS ~~Apache~~ web server.”

systems, then queries 134 the network discovery functionality 112 to retrieve information as to whether any Microsoft IIS web servers and/or Microsoft operating systems exist on the network 14. Responsive enterprise specific data 136, identifying the machines of the network using Microsoft IIS web servers and/or Microsoft operating systems, is then supplied to the agent 126.

W-L at 12:44-61.

70. This also includes both the embodiments of FIG. 1, where W-L teaches that the system 10 obtains specifically those signatures “that are designed to mitigate or avert network damage from detected vulnerabilities.” W-L at 5:20-24. Such “detected vulnerabilities” included the enterprise specific data just noted. *See, e.g.*, W-L at 5:9-15 (enterprise specific data), 5:15-19 (vulnerability assessments to obtain enterprise specific data). A POSITA would have recognized that the enterprise specific data includes operating system configuration information since it is a well-known type of information about the enterprise and is relevant to tailoring detection signatures to a particular enterprise’s network.

71. **Second**, W-L teaches identifying that the configuration is “*associated with a plurality of devices*.” Continuing with the operating system example, W-L teaches:

Using the supplied data 136, the agent 126 decides whether to instantiate the detection signature on either or both the intrusion detector functionality 116 and firewalling functionality 118 to address the perceived threat. **This decision is made in view of the supplied enterprise specific data 136.** If the data 136 indicates that there are no machines in the network 14 that are susceptible to the threat (for example, there are no machines with using Microsoft IIS web servers and/or Microsoft operating systems), the agent could choose not to instantiate the detection signature 132 at all. **If there is, however, a risk from the threat in the network 14 environment, or if there would be some benefit from tracking attempts to exploit the network using this threat, the agent 126 may decide to instantiate.** However, before instantiating the detection signature 132, the agent 126 tailors the action set so that the actions taken responsive to the detection of the threat are appropriate. For example, if there are no machines with using Microsoft IIS web servers and/or Microsoft operating systems, the threat is not valid in the particular network 14 configuration being protected and the agent 126 may tailor the action set to merely log the attack rather than take action against the attack. In this way, no false positive alerts are generated.

W-L at 12:61-13:17.

72. Here, W-L explains that the agent 126 identifies whether the operating system configuration is associated with any devices, “machines” in W-L’s description. While described generally with respect to FIG. 2’s example, W-L also taught that the aspects of FIG. 2 have parallels to the aspects disclosed in FIG. 1. *See* W-L at 8:54-58.

73. W-L generally also teaches identifying whether a configuration is associated with specific machines in the network, which a POSITA would recognize as teaching at least a “*first device, a second device, and a third device.*” It was well-known to POSITAs that enterprise networks, such as that disclosed in W-L, commonly included hundreds or thousands of computers. Further, Microsoft operating systems were one of the most commonly used operating systems for computers, including those in enterprise networks.

74. “Responsive to the detection signature, the agent 126 queries 134 the network discovery functionality 112 to retrieve information as to whether any IP ports of a host on the network 14 provide HTTP services. Responsive enterprise specific data 136, identifying the IP ports, hosts, and related machine data that provide HTTP services, is then supplied to the agent 126.” W-L at 12:9-15. The “enterprise specific data,” including the ports, hosts, and related machine data, are examples of “*at least one configuration*” in addition to the operating system example already noted.

75. W-L continues, explaining that the agent decides whether to instantiate a signature based on the supplied data, the decision being made in view of the supplied enterprise specific data 136. If the data 136 indicates that there are no machines in the network 14 that are susceptible to the attack, there would be no need to instantiate the detection signature 132 at all. **If there is, however, a risk from the attack in the network 14 environment, the agent 126 next decides whether to restrict the signature to a specific set or segment of machines in the network 14.** Again, this decision is made in view of the supplied enterprise specific data 136. **If only certain machines in the network 14, or certain zones within the network, are susceptible to the attack, then the detection signature 132 is instantiated on only the intrusion detector functionality 116 and/or firewalling functionality 118 that protect the threatened portions of the network.**

W-L at 12:15-33.

76. As W-L explains, the agent determines whether to instantiate a detection signature based on the supplied enterprise specific data, which is “*at least one configuration.*” W-L’s “certain machines in the network” that are susceptible to attack are identified by that configuration information in the enterprise specific data, such as computers using a Microsoft operating system. Therefore, W-L

teaches that the configuration is “associated with a plurality of devices including a first device, a second device, and a third device.” Specifically, it would have been obvious to a POSITA that the “certain machines in the network” teaches the claimed first, second, and third devices constituting the plurality of devices, since enterprise networks generally contained numerous machines.

77. Thus, W-L renders obvious “identifying at least one configuration associated with a plurality of devices including a first device, a second device, and a third device,” as claimed.

- e. ***[10.4] determining that the plurality of devices is vulnerable to at least one accurately identified vulnerability based on the identified at least one configuration, utilizing the second vulnerability information that is used to identify the plurality of potential vulnerabilities;***

78. W-L renders obvious this limitation.

79. **First**, W-L teaches that a plurality of devices are vulnerable to a vulnerability that is identified based configuration information, thereby reducing false positives. As noted in [1.3], W-L teaches that

the enterprise specific data is considered by the agent when issuing a detection signature so that the signature (either through its criteria or its response instructions) **is designed in a way that minimizes the likelihood that false positive alarms** will be generated. This cooperative defense effort between the network discovery

functionality 112, intrusion detector functionality 116 and firewalling functionality 118, **through the agent 126 tailoring the detection signatures based on enterprise specific data**, effectuates a self-hardening system 10.

W-L at 10:3-19.

80. Minimizing false positives by using the enterprise specific data when tailoring a detection signature results in accurately identifying a vulnerability of the plurality of devices. Indeed, **“[i]f only certain machines in the network 14, or certain zones within the network, are susceptible to the attack, then the detection signature 132 is instantiated on only the intrusion detector functionality 116 and/or firewalling functionality 118 that protect the threatened portions of the network.”** W-L at 12:15-33. W-L teaches that the susceptibility to attack is determined by considering the enterprise specific data (which includes “*at least one configuration*”).

81. Further, W-L teaches that a specific example of enterprise specific data includes data identifying the operating system of machines in the network. *See* W-L at 12:44-61. This includes teaching that some machines in the network are using an operating system that is susceptible to an identified threat. *See* W-L at 12:61-13:17 (“The threat aggregation functionality 128 collects and stores a piece of threat information 130 relating to a network vulnerability relating the malicious code intended to exploit a Microsoft IIS web server running on a Microsoft

operating system. The network administrator 142 may design a detection signature 132 for this particular threat.”). Thus, W-L teaches that “certain machines” (*“the plurality of devices”*) in the network are susceptible to an attack based on the enterprise specific data considered by the agent (*“vulnerable to at least one accurately identified vulnerability based on the identified at least one configuration”*).

82. Second, W-L renders obvious that the determination that the plurality of devices is vulnerable uses the *“second vulnerability information.”* As noted at [1.1], W-L teaches that the “threat information 130” and “detection signatures 132” are both used to *“identify the plurality of potential vulnerabilities.”* See W-L at 10:40-42 (detection signatures 132 based on threat information 130); 11:12-29 (unknown whether detection signatures 132 are relevant to the network until compared to enterprise specific data).

83. Thus, when W-L teaches determining whether to instantiate a detection signature 132 from threat aggregation functionality 128, it is determining whether the plurality of devices are vulnerable by using the detection signature 132, and threat information 130 the signature 132 is based on. See W-L at 10:3-19, 12:15-33.

84. W-L therefore renders obvious “determining that the plurality of devices is vulnerable to at least one accurately identified vulnerability based on the

identified at least one configuration, utilizing the second vulnerability information that is used to identify the plurality of potential vulnerabilities,” as claimed.

f. **[10.5] *display information that is based on the first vulnerability information;***

85. W-L renders obvious this limitation.

86. W-L discloses an intrusion alert that is sent to and reviewed by a system administrator, which renders obvious “*displaying*” vulnerability information.

87. **First**, W-L discloses generating an alert when intrusion is detected. As described in [10.1]-[10.4], W-L teaches how the intrusion detector functionality, alone or together with the firewalling functionality, receives tailored signatures (“*the first vulnerability information*”). Referring to FIG. 2, a tailored detection signature may be instantiated toward the intrusion detector functionality and/or firewalling functionality. W-L at 13:30-33. If the intrusion detector functionality and/or firewalling functionality subsequently detects traffic that matches the criteria of the detection signature, “**the threat response actions defined by that signature are then invoked** in step 212 (**for example**, the traffic is logged or blocked, **an alert is generated to the network administrator** 142, a session is terminated, or the like).” W-L at 13:35-42. W-L teaches how the intrusion detector functionality itself issues the alert. W-L at 13:35-42, cl. 9 (“**an**

intrusion detector functionality analyzing network traffic received ... in view of the instantiated detection signature ... and **issuing alerts.**”).

88. **Second**, it was well known at the time of the invention that intrusion prevention systems included graphical user interfaces to display information about an intrusion event, such as alerts, to a network manager. Schertz at ¶18 (“FIG. 2 is a more detailed functionality block diagram of **an intrusion protection system 14 with a user interface system 10** according to an embodiment of the present invention.”); ¶21 (“**User interface system 10 comprises a graphical user interface 12**, which is capable of displaying real-time status information as well as archived data.”); ¶24 (“**The signature associated with the current intrusion event is displayed graphically 102 to the user**, as shown in block 44. The decoded event data, such as Ethernet header summary 104 and IP header summary 106, and also the IP header data in hexadecimal format 108 are also **displayed** as shown in block 46.”).

89. Thus, referring to W-L’s FIG. 2, since intrusion detector functionality 116 communicates with security management agent 126, which in turn communicates directly with network administrator 142, and since it was common for computers to provide information to human users via a visual display interface, it would have been obvious to a POSITA for the alert generated by the intrusion detector functionality 116 to be displayed to the network administrator 142 from

unified network defense system 10. It further would have been obvious to a POSITA that W-L's platform and security functionality includes a computer readable media storing the instructions to “display” the alert information from the intrusion detector functionality and/or firewalling functionality to the network administrator (user).

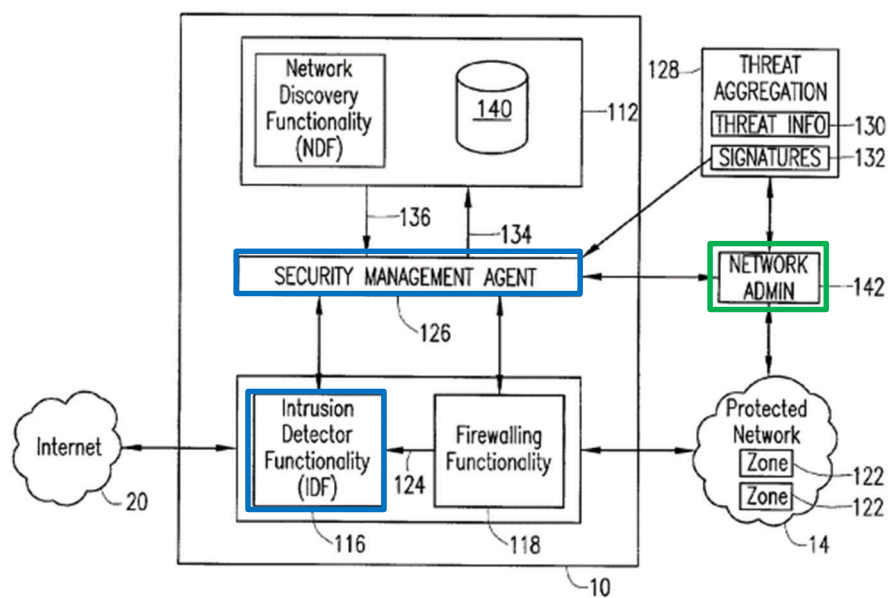


FIG. 2

W-L at FIG. 2 (annotated)

90. Third, as another example, as described in [10.1-10.4], an inspection agent may receive the tailored detection signatures. Referring to FIG. 1, an inspection agent may inspect network traffic “using threat detection signatures to detect the existence of potentially harmful content in the traffic.” W-L at 3:41-46. “In the event that the comparison 40 operation performed by the inspection agent

Ex.1003

28 is satisfied (i.e., there is a criteria match), anyone (or more than one) of a number of possible actions (specified by an object within the detection signature 22 itself) may be taken by the system 10. For example, **the inspection agent 28 may issue an alarm report 50 to the administrator** 24. The identification of the threatened machine (host) or service is provided using information contained in the report 50.” W-L at 5:59-67; *see also* 7:4-13 (“In the event that the comparison 44 operation performed by the sentry 42 with respect to the detection (blocking) signature 22’ is satisfied (i.e. there is a criteria match) ... **the sentry 42 may issue an alarm report 50 to the administrator** 24. The identification of the threatened machine (host) or service is typically available from the information contained in the report 50”). For the reasons stated in [10.5] *supra*, it would have been obvious to a POSITA to employ a graphical user interface to “*display*” an alarm report issued from the inspection agent 28 to the network administrator 24 (or from the entrance sentry 42 to the network administrator 24). Such “*display*” functionality was common and well-known to those familiar with working with computers. *See* Schertz at ¶¶18, 21, 24.

91. Furthermore, referring to FIG. 1, since inspection agent 28 (or entrance sentry 42) communicates directly with network administrator 24, it would have been obvious to a POSITA that the alarm report issued by inspection agent 28

(or entrance sentry 42) is displayed to the network administrator 24 by unified network defense system 10, as described in [10.0].

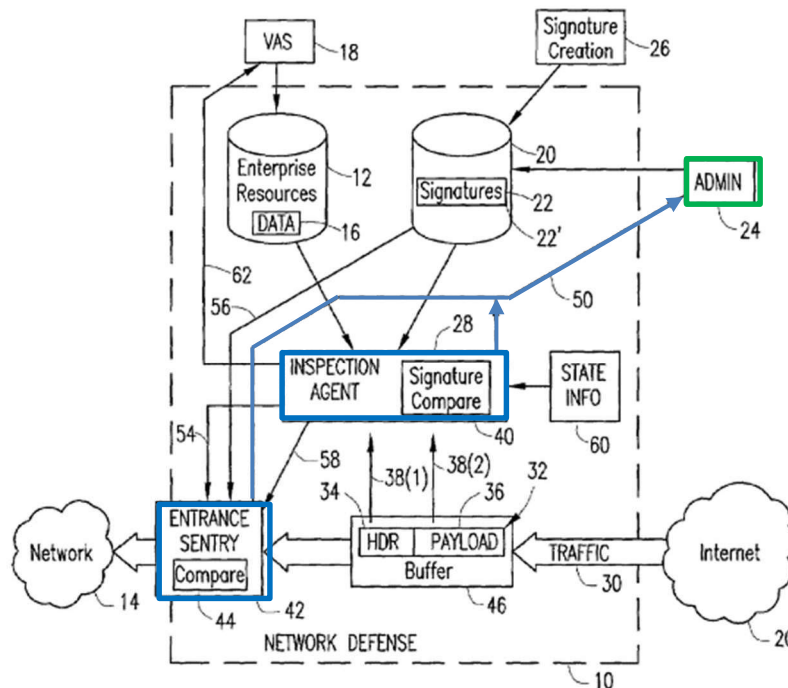


FIG. 1

Ex.1005, FIG. 1 (annotated)

92. A POSITA would have understood that W-L’s platform and security functionality include a computer-readable memory storing instructions to cause one or more processors to “*display*” the alarm report information from the inspection agent (or entrance sentry) (“*information that is based on the first vulnerability information*”).

93. Therefore, W-L renders obvious [10.5].

g. [10.6] *cause utilization of different occurrence mitigation actions of diverse occurrence mitigation types, including a firewall-based occurrence mitigation type and an intrusion*

Ex.1003

mitigation system-based occurrence mitigation type, across the plurality of devices for occurrence mitigation by preventing advantage being taken of accurately identified vulnerabilities utilizing the different occurrence mitigation actions of the diverse occurrence mitigation types across the plurality of devices;

94. W-L renders obvious this limitation.

95. First, W-L discloses “*caus[ing] utilization of different occurrence mitigation actions.*” W-L generally teaches its system taking actions to mitigate a detected threat in traffic:

Subsequently, the intrusion detector functionality 116 **detects in step 310 an attempt from outside the network 14 to exploit the noted vulnerability** (for example, a certain IP address is noted as being a source of traffic relating to the IIS buffer overflow vulnerability as detected by the code red signature). This is reported back to the agent 126 in step 312. The agent 126 responds in step 314 by reporting to the network administrator 142 that an attempt to exploit has occurred. The agent 126 further instructs the firewalling functionality 118 in step 316 to **drop the packet traffic relating to the attempted exploitation**. Additionally, the agent 126 instantiates a policy on the firewalling functionality 118 in step 318 **instructing the firewall to block all traffic originating from the noted IP address of the attacker**. The network

administrator 142 **is then updated** in step 320 that traffic from the noted IP address has been blocked.

W-L at 14:46-15:8; *see also* 15:9-21.

96. W-L discloses “*diverse occurrence mitigation types*.” For example, W-L teaches actions including “**block and terminate actions to be taken**. ... the agent 126 may establish a **block action to be taken by the firewalling functionality 118** to block the attack-related traffic, **and a terminate action to be taken by the intrusion detector functionality 116** to terminate any session associated with a possible attack.” W-L at 12:28-43. The “action taken by the firewalling functionality 118” is an example of “*a firewall-based occurrence mitigation type*,” which in this example is a “block” action. Further, the “action taken by the intrusion detector functionality 116” is an example of “*an intrusion prevention system-based occurrence mitigation type*,” which includes in this example a terminate action (to terminate the session).

97. In particular, W-L’s disclosure of the “intrusion detector functionality 116” renders obvious the “*intrusion prevention system*.” Indeed, the ’063 patent refers to intrusion detection and intrusion prevention systems interchangeably. *See* ’063 patent at 19:62-63. The ’063 patent further describes operations performed by “IPS” (Intrusion Prevention System) including outputting “alert data,” and terminating an attack at the IPS. *See* ’063 patent at 20:1-7 (the “in-line sensor”

referring to an in-line IPS sensor, see 19:59). Since W-L describes the intrusion detection functionality 116 taking actions including “a terminate action” as well as alerts, W-L’s intrusion detector functionality 116 discloses an IPS. *See, e.g.*, W-L at 12:28-43 (terminate action), 9:21-27 (alerts). More specifically, W-L’s actions taken by the intrusion detector functionality 116 teach “*an intrusion prevention system-based occurrence mitigation type.*”

98. Since W-L teaches causing several different actions in response to detecting a threat in traffic, W-L teaches to “*cause utilization of different occurrence mitigation actions of diverse occurrence mitigation types, including a firewall-based occurrence mitigation type and an intrusion prevention system-based occurrence mitigation type.*”

99. Second, W-L teaches applying the mitigation actions “*across a plurality of devices.*” W-L teaches that the system 10 that includes intrusion detector functionality 116 and firewalling functionality 118 is located at the edge of a network, such that any mitigation actions apply “*across a plurality of devices.*” For example, responsive to an alert from intrusion detector functionality 116, “the firewalling functionality 118 adapts in real-time to the active threats and stops malicious traffic at the edge of the network 14 before it can enter, infect and debilitate the network systems.” W-L at 9:27-31. This is because the system 10, with the intrusion detector and firewalling functionalities, operates as an “in-line”

component such that “all traffic 30 must pass through ... before entering the protected network 14.” W-L at 6:26-29. Thus, W-L’s protective actions (such as block and terminate, by firewalling and intrusion detector functionalities respectively), by applying to the network in general, applies to the “*plurality of devices*” within that network.

100. As another example, W-L teaches implementing the firewalling functionality 118 on a zone-by-zone basis: “[t]he firewalling functionality 118 manages advanced firewall functionalities including: encryption acceleration; NAT/PAT; traffic shaping and URL filtering. Independent management of plural zones 122 within the network 14 may be employed. In this way, **the firewalling functionality 118 may be independently tuned to provide threat protection on a zone-by-zone basis.**” W-L at 9:8-14. W-L’s use of “zones” refers to coverage of something larger than individual computers: “the elements of the protected network 14 may include: a router; special server types (for example, HTTP, SMTP, FTP, DNA, and the like); an intranet; **personal computers; and, network zones.**” W-L at 15:60-63.

101. It would have been obvious to a POSITA that a network zone would include anywhere from one to many different computing devices (even on the order of thousands). Accordingly, W-L’s teaching of using of the firewalling functionality to prevent attacks on a zone-by-zone basis (which was known to

often include many computing devices in a protected network), also renders obvious applying the mitigation actions “*across the plurality of devices.*”

102. Third, W-L teaches “*preventing advantage being taken of the accurately identified vulnerabilities*” by performing the caused actions.

103. As noted at [10.4], the “*accurately identified vulnerability*” was identified from the enterprise specific data (used to tailor a detection signature). See W-L at 10:3-19 and [10.4]. As demonstrated, the detection signatures were tailored to address the actual vulnerability (or vulnerabilities) of one or more machines in the network. W-L’s use of such tailored detection signatures in the “intrusion detector functionality 116 and/or firewalling functionality 118” means that the actions taken based on those signatures prevent advantage being taken of the identified vulnerabilities.

104. Moreover, W-L’s use of multiple mitigation techniques shows that the system is using “*the different occurrence mitigation actions*” (the actions specified in one or more detection signatures 132 instantiated at the “intrusion detector functionality 116 and/or firewalling functionality 118”) “*of the diverse occurrence mitigation types*” (the actions taken by each of the functionalities 116 and 118, including terminate and block, for example) “*across the plurality of devices*” (one or both of intrusion detector functionality 116 and firewalling functionality 118

being applied at the network edge to protect the devices within the network, or within specific zones of the network).

105. W-L therefore renders obvious “*caus[ing] utilization of different occurrence mitigation actions of diverse occurrence mitigation types, including a firewall-based occurrence mitigation type and an intrusion prevention system-based occurrence mitigation type, across the plurality of devices for occurrence mitigation by preventing advantage being taken of accurately identified vulnerabilities utilizing the different occurrence mitigation actions of the diverse occurrence mitigation types across the plurality of devices,*” as claimed.

- h. [10.7] receive an indication that an occurrence has been identified in connection with at least one of the plurality of devices utilizing one or more monitors;**

106. W-L discloses this limitation.

107. First, W-L discloses an (intrusion) *occurrence in connection* with a device is identified. W-L teaches the intrusion detector functionality 116 and firewalling functionality 118 receive and inspect packets using instantiated detection signature(s). In W-L once a detection signature(s) has been instantiated at intrusion detector functionality 116 and/or firewalling functionality 118, both receive and inspect packets. W-L teaches that “network protection is provided by having an inspection agent **extract features (for example, packet features) from**

entering traffic.” W-L at 3:41-44; *see also* 5:37-41 (traffic is inspected as it enters protected network).

108. The receipt of one or more packets, and generally traffic, at the device discloses “*identify an occurrence.*” This is because, after features are extracted, the inspection agent evaluates the features “using threat detection signatures to detect the existence of potentially harmful content in the traffic. If harmful content is detected, an entrance sentry responds to the agent detection by denying the traffic admission to the network.” W-L at 3:44-49. This is consistent with how the ’063 patent treats an “occurrence,” where in dependent claim 33 the ’063 patent further recites that “said occurrence includes at least one of a request, traffic, at least one packet, or a potential attack.” ’063 patent at 39:18-19.

109. W-L’s teaching of receiving network traffic to subsequently analyze and identify potentially harmful content, combined with W-L’s teaching that the network includes “certain machines in the network” that may be vulnerable to an attack that may be present in the traffic, renders obvious “*an occurrence*” is identified (the arrival of one or more packets in “traffic”) “*in connection with at least one of the plurality of devices*” (the “certain machines in the network”).

110. Second, W-L renders obvious receiving “*an indication*” that “*an occurrence is identified.*” A POSITA would have further understood that for the inspection agent to “extract features (for example, packet features) from entering

traffic,” W-L at 3:41-44; *see also* 5:37-41, the agent must first “*receive an indication*” that one or more packets of network traffic have been actually received so that it can extract those features.

111. Alternatively, it would have been obvious to a POSITA for the inspection agent to receive a signal (“*indication*”) when a packet arrives so that the inspection agent would be able to promptly begin analyzing the packet. Such an arrangement was known in the art as event-driven programming and was well-known in the computing and networking fields before the earliest priority date. See King at 3:67-4:5 (“In event-driven programming, the program is code is written so that the program can respond to various types of external events as they occur. The program itself has little or no control over the sequence and choice of events generated, but simply responds by carrying out the appropriate actions in response to the generated events.”).

112. Furthermore, following the detection of threatening traffic, the inspection agent may act to block the threatening traffic from entering the network. W-L at 6:21-23. “Responsive to the detection of threatening traffic 30, the inspection agent 28 issues a blocking command 54 to the entrance sentry 42.” This command 54 includes sufficient information to allow the entrance sentry 42 to identify the threatening traffic 30 and deny it entry to the protected network 14.”

W-L at 6:29-34. Thus, when the entrance sentry receives a blocking command

from the inspection agent, the entrance sentry “*receiv[es] an indication*” that the inspection agent has detected network traffic (“*that an occurrence has been identified*”) and that the traffic has been determined to be threatening. Thus, W-L also discloses the entrance sentry “*receiving an indication*” from the inspection agent “*that an occurrence has been identified.*”

113. As another example, as described in [10.5], an inspection agent may inspect network traffic “using threat detection signatures to detect the existence of potentially harmful content in the traffic.” W-L at 3:41-46. In the event of a criteria match between network traffic and a tailored signature, “**the inspection agent 28 may issue an alarm report 50 to the administrator** 24.” W-L at 5:59-67; *see also* 7:4-13 (“**the sentry 42 may issue an alarm report 50 to the administrator** 24.”). Thus, when the network administrator receives an alarm report, it “*receiv[es] an indication*” that the inspection agent (or entrance sentry) has detected network traffic that is potentially harmful. *See* [10.5] (describing an intrusion detector functionality and/or firewalling functionality that generates an alert to the network administrator (“*receiving an indication*”) after it detects traffic (“*that an occurrence has been identified*”) that matches the criteria of a tailored detection signature). In these multiple ways, W-L renders obvious “*receiving an indication that an occurrence has been identified.*”

114. Second, W-L's description of the intrusion detector functionality 116, alone or in combination with the firewalling functionality 118 operating on incoming traffic, teaches "*utilizing one or more network monitors.*" Either functionality 116 or 118 teaches one "*network monitor,*" and both together teaches multiple such "*monitors.*" The intrusion detector functionality 116 "monitors traffic to and from the network being protected." W-L at 15:24-26. As noted above, W-L teaches receiving packets, and extracting features from those packets. While described with respect to an "inspection agent" of FIG. 1, W-L clarifies that the "intrusion detector functionality (IDF) 116" of FIG. 2 performs this same function. W-L at 8:54-58 ("see, also, inspection agent 28 and compare functionality in FIG. 1"). Further, the firewalling functionality 118 implements "a stateful, full packet inspection of passing traffic" and makes "packet-by-packet" decisions. W-L at 8:66-9:3. The intrusion detector functionality 116 and the firewalling functionality 118 therefore both disclose "*one or more network monitors.*"

115. W-L therefore discloses "*receiv[ing] an indication that an occurrence has been identified in connection with at least one of the plurality of devices utilizing one or more monitors,*" as claimed.

- i. **[10.8] wherein the at least one configuration involves at least one operating system.**

116. W-L discloses this limitation.

117. As noted at [10.3], the configuration may be an operating system configuration.

The threat aggregation functionality 128 collects and stores a piece of threat information 130 relating to a network vulnerability relating the malicious code intended to exploit a Microsoft IIS web server running on a Microsoft operating system. The network administrator 142 may design a detection signature 132 for this particular threat. The detection signature 132 is then provided to the agent 126. **The agent, noting the query object specification of Microsoft IIS web servers and Microsoft operating systems, then queries 134 the network discovery functionality 112 to retrieve information as to whether any Microsoft IIS web servers and/or Microsoft operating systems exist on the network 14.** Responsive enterprise specific data 136, **identifying the machines of the network using Microsoft IIS web servers and/or Microsoft operating systems,** is then supplied to the agent 126.

W-L at 12:44-61.

118. W-L therefore discloses “*wherein the at least one configuration involves at least one operating system,*” as claimed.

2. Claim 11

Ex.1003

- a. ***[11.0] The non-transitory computer-readable media of claim 10, wherein the instructions are configured such that the utilization of the different occurrence mitigation actions is caused for different devices prior to the occurrence.***

119. W-L discloses this limitation.

120. As described in [10.6], W-L renders obvious instructions configured to cause “*utilization of different occurrence mitigation actions*” for “*different devices*” (i.e. “*the plurality of devices*”).

121. W-L further teaches utilizing the different occurrence mitigation actions “*prior to*” the occurrence. W-L provides a flow diagram in Figure 3 showing how the tailored signature is instantiated on the intrusion detector functionality and firewalling functionality “*prior to*” detecting traffic that matches the criteria of the detection signature.

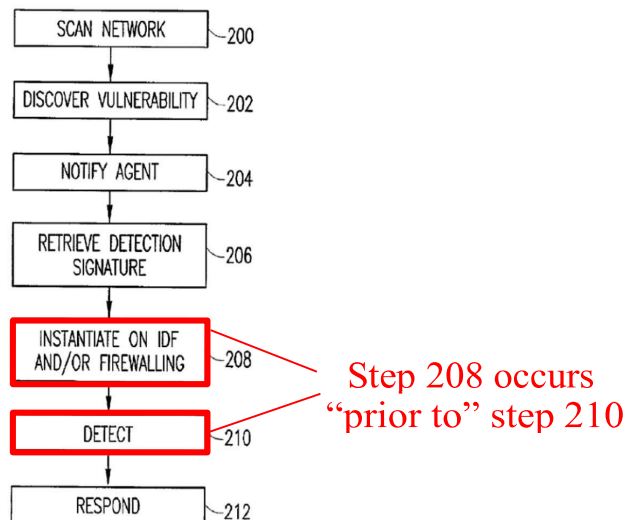


FIG. 3

W-L at FIG. 3 (annotated).

Ex.1003

122. If a network vulnerability is discovered in step 202, the agent is notified in step 204. W-L at 13:25-27. The agent then retrieves from enterprise vulnerabilities database a detection signature associated with the vulnerability. W-L at 13:27-29.

The detection signature 132 is tailored in the context of the network configuration **and then instantiated in step 208 toward the intrusion detector functionality 116 and/or firewalling functionality** 118 for the purpose of tuning (or configuring) their operation to protect the network 14 against the recognized vulnerability. If the intrusion detector functionality 116 and/or firewalling functionality 118 **subsequently detects traffic that matches the criteria of the detection signature** 132 (step 210), the threat response actions defined by that signature are then invoked in step 212 (for example, the traffic is logged or blocked, an alert is generated to the network administrator 142, a session is terminated, or the like).

W-L at 13:30-42.

123. Thus, W-L teaches the utilization of different occurrence mitigation actions (i.e. instantiating signatures on the intrusion detector and/or firewalling functionality as discussed in [10.6]) “*prior to*” detecting traffic that matches the detection signature.

124. W-L therefore discloses “*the instructions are configured such that the utilization of the different occurrence mitigation actions is caused for different devices prior to the occurrence,*” as claimed.

3. Claim 39

- a. **[39.0] *A non-transitory computer-readable media storing instructions that, when executed by one or more processors, cause the one or more processors to:***

125. See analysis at [10.0].

- b. **[39.1] *receive first vulnerability information from at least one first data storage that is generated utilizing second vulnerability information from at least one second data storage that is used to identify a plurality of potential vulnerabilities,***

126. See analysis at [10.1].

- c. **[39.2] *by including: at least one first potential vulnerability, and at least one second potential vulnerability;***

127. See analysis at [10.1], which recites a similar limitation including a “*plurality of potential vulnerabilities.*” A POSTIA would have recognized that a plurality of potential vulnerabilities includes “*at least one first potential vulnerability, and at least one second potential vulnerability.*”

- d. **[39.3] *said first vulnerability information generated utilizing the second vulnerability information, by:***

128. See analysis at [10.2].

- e. ***[39.4] identifying at least one configuration associated with a plurality of devices including a first device, a second device, and a third device, and***

129. See analysis at [10.3].

- f. ***[39.5] determining that the plurality of devices is actually vulnerable to at least one actual vulnerability based on the identified at least one configuration, utilizing the second vulnerability information that is used to identify the plurality of potential vulnerabilities;***

130. See analysis at [10.4]. Claim element [39.5] differs because it recites:

“actually vulnerable” in place of “vulnerable,” and at least one “actual vulnerability” in place of “accurately identified vulnerability.” A POSITA would understand that the same analysis stated in [10.4] is applicable to [39.5] despite the minor language differences. For example, because W-L discloses “determining that the plurality of devices is vulnerable to ... accurately identified vulnerability,” a POSITA would understand that W-L similarly discloses “*determining that the plurality of devices is actually vulnerable ... actual vulnerability.*” For the same reasons stated in [10.4], W-L renders obvious [39.5].

- g. ***[39.6] identify an occurrence in connection with at least one of the plurality of devices, utilizing one or more monitors;***

131. As discussed in [10.7], W-L discloses “*that an occurrence has been identified*” in connection with at least one of the plurality of devices *utilizing one or more monitors*. W-L thus renders obvious “*identify an occurrence*” of [39.6].

- h. *[39.7] based on a packet analysis, determine that the at least one actual vulnerability of the at least one of the plurality of devices is susceptible to being taken advantage of by the occurrence identified in connection with the at least one of the plurality of devices, utilizing the first vulnerability information; and*

132. W-L discloses this limitation.

133. First, W-L discloses *packet analysis* by teaching matching packet contents to at least one detection signature. The intrusion detector functionality 116 “analyzes network traffic to identify content that is potentially harmful to the network 14.” W-L at 8:54-58. Analyzing the network traffic includes, per W-L, packets, “with each packet including a header portion 34 and a payload portion 36.” W-L at 5:39-41. W-L teaches generally “**comparing 40 the extracted packet features against the detection signatures 22** [FIG. 1, obtained from the signature database 20].” W-L at 5:50-53. The comparison is done to determine if “there is a criteria match” between content of the packet and content of a signature. W-L at 5:53-63. Thus, W-L teaches a “*packet analysis*.”

134. W-L teaches an example operation after a detection signature is tailored and instantiated to functionality 116 and/or 118, using the packet analysis. “If the intrusion detector functionality 116 and/or firewalling functionality 118 subsequently **detects traffic that matches the criteria of the detection signature 132** (step 210), the threat response actions defined by that signature are then

invoked.” W-L at 13:30-39. The “detection signature 132” which the traffic matches is an example of the “*first vulnerability information.*” See [1.1]. Thus, W-L teaches “*utilizing the first vulnerability information*” when making its determination “*based on a packet analysis.*”

135. Second, W-L discloses determining an “*actual vulnerability of the at least one of the plurality of devices is susceptible to being taken advantage of.*” Gupta teaches that the criteria in the detection signature 132 has been tailored based on enterprise specific data identifying configuration information of the machines in the network. See W-L at 10:3-19 (tailoring detection signatures based on enterprise specific data); *see also* [1.4].

136. Thus, when W-L teaches detecting a match between content of a packet (traffic) and the criteria in a tailored detection signature, W-L teaches determining that an “*accurately identified vulnerability*” (from the enterprise specific data) is “*susceptible to being taken advantage of*” (indicated by the match) “*by the occurrence identified in connection with the at least one of the plurality of devices*” (the receipt of traffic intended for one or more machines in the network 14).

137. W-L therefore discloses “*based on a packet analysis, determine that the at least one accurately identified vulnerability of the at least one of the plurality of devices is susceptible to being taken advantage of by the occurrence*

identified in connection with the at least one of the plurality of devices, utilizing the first vulnerability information,” as claimed.

- i. ***[39.8] permit selective utilization of different occurrence mitigation actions of diverse occurrence mitigation types, including a firewall-based occurrence mitigation type and a other occurrence mitigation type, across the plurality of devices for occurrence mitigation by preventing advantage being taken of actual vulnerabilities utilizing the different occurrence mitigation actions of the diverse occurrence mitigation types across the plurality of devices;***

138. See analysis of [10.6], which recites a similar “*different occurrence mitigation actions*” limitation. Claim element [39.8] differs because it recites “*permit selective utilization*” in place of “cause utilization” of difference occurrence mitigation actions, “*a other*” in place of “an intrusion mitigation system-based” occurrence mitigation type, “*actual vulnerabilities*” in place of “accurately identified vulnerabilities. As discussed in [10.6], W-L renders obvious to “cause utilization” of different occurrence mitigation actions of diverse mitigation types carried out by the firewalling functionality 118 and the intrusion detector functionality 116, such as drop, block, and terminate actions.

139. W-L also renders obvious to “*permit selective utilization*” of the different occurrence mitigation actions of diverse mitigation types. For example, W-L teaches that when there is risk of an attack, detection signatures may be instantiated to specific sets or segments of machines, selectively:

[T]he agent 126 next decides **whether to restrict the signature to a specific set or segment of machines in the network 14**. Again this decision is made in view of the supplied enterprise specific data 136. **If only certain machines in the network 14, or certain zones within the network, are susceptible to the attack, then the detection signature 132 is instantiated on only the intrusion detector functionality 116 and/or firewalling functionality 118** that protect the threatened portions of the network. Lastly, before instantiating the detection signature 132 as needed, the agent 126 tailors the action set so that actions taken responsive to the detection of the attack are appropriate....In implementing these actions, the agent 126 may establish a block action to be taken by the firewalling functionality 118 to block the attack-related traffic, and a terminate action to be taken by the intrusion detector functionality 116 to terminate any session associated with a possible attack.

Ex.1005, 12:23-43.

140. Because signatures may be restricted to specific sets or segments of machines in the network, and may also be tailored with different action sets to be performed by the intrusion detector functionality and/or firewalling functionality, W-L renders obvious to “*permit selective utilization*” of different occurrence mitigation actions of diverse mitigation types.

Ex.1003

FORCEPOINT LLC/ Page 68 of 116

141. Furthermore, as discussed in [10.6], the actions taken by the intrusion detector functionality (e.g. terminate action) may be of a different (“a other”) type than the actions taken by the firewalling functionality (e.g. block action). And as further discussed in [10.6], the accurately identified vulnerabilities identified from enterprise specific data are “*actual vulnerabilities*” of one or more machines in the network.

142. Therefore, W-L renders obvious claim element [39.8].

j. [39.9] *wherein the at least one configuration involves at least one operating system.*

143. See analysis at [10.8].

4. Claim 58

a. [58.0] *The non-transitory computer-readable media of claim 39, wherein the instructions are configured such that the utilization of the different occurrence mitigation actions is selectively permitted for different devices prior to the occurrence.*

144. See analysis at [11.0].

IX. THE COMBINATION OF GUPTA AND GRAHAM RENDERS OBVIOUS CLAIMS 1 AND 5

A. Summary of Gupta

145. U.S. Patent Application No. 2003/0004689 to Gupta (Ex. 1006, “Gupta”), entitled “Hierarchy-based Method and Apparatus for Detecting Attacks on a Computer System,” is directed toward a hierarchical organization of

Ex.1003

threat/vulnerability information, which represents potential vulnerabilities. This potential information is matched against device configuration data in order to produce actual vulnerability data. As stated in the abstract:

A method of provisioning a computer against computer attacks includes constructing a **hierarchy characterizing different computer attacks** and counter measures, and traversing this hierarchy to **identify computer attacks and countermeasures relevant to a target platform**. Detection and protection measures are collected in response to this traversing. These detection and protection measures are then downloaded to a security sensor associated with the target platform.

Gupta, Abstract.

146. In more detail, Gupta discloses creating customized “attack files” designed to protect a specific target.

Detection and protection measures for the target platform are then collected (block 164). This can result in an attack file 149. A sensor is then supplied, through a download, with the protective software (e.g., the attack file) for the target platform (block 166).

The hierarchical approach of the invention allows one to clearly describe, in one data structure, what attacks a given environment is vulnerable to, what protection means

(mechanisms and signatures) are available, and how detection alerts are correlated.

Gupta, [0164]-[0165].

147. The identified protection measures are used to protect the system and include IPS-based mitigation actions such as dropping the connection:

The response processor 54 responds to AIDs. The response processor 54 generates alerts that provide notification of an attack. The response processor 54 also attempts to prevent the attack. Short-term responses include terminating TCP connections.

Gupta, [0087].

B. Summary of Graham

148. U.S. Patent No. 7,237,264 to Graham et al. (Ex. 1007, “Graham”), entitled “System and Method for Preventing Network Misuse,” is also directed to identifying potential vulnerability information and deriving actual vulnerability data from it. Graham teaches correlating “target fingerprint data ... with the context-based and/or state-based data signature (as indicated at 255) to determine whether the target is actually vulnerable to the suspicious data signature.”

Graham at 7:14-19.

149. Graham also discloses using firewall-based mitigation actions to prevent an attack from taking advantage of an actual vulnerability.

If the system determines that the target is not vulnerable, one embodiment of the system generates a low-level alert and/or simply logs the event signature (or may take no action whatsoever, depending on the configuration). **If the system determines that the target is vulnerable,** however, then one embodiment will generate an alert condition (or increase a preexisting alert level) and/or **may take certain precautionary measures** (at 235). For example, at this stage **the system may attempt to block incoming data transmissions from the suspect node (e.g., by commanding the firewall to do so).**

Graham at 7:58-67.

C. Reasons to Combine Gupta with Graham

150. One of ordinary skill in the art would have been motivated to combine Gupta and Graham. Both disclosures are directed towards the same problem, and they share a number of common elements in their solutions. For example, both disclosures make use of potential vulnerability data to determine actual vulnerability data.

151. Gupta, however, is broader in its view of vulnerabilities. While it discloses software vulnerabilities (*see, e.g.,* buffer overflow attacks, Gupta, [0156]), it also discloses protocol and network vulnerabilities as well. Graham discloses greater detail and specificity with remediating and mitigating such weaknesses. One of ordinary skill in the art as of July 2003 would have looked to

Ex.1003

a reference, such as Graham, to provide details appropriate to their technological environment. Such a combination would have been prompted by Gupta's suggestion to identify target-specific attacks. Gupta, Abstract. Graham enables the skilled artisan, in possession of Gupta, to employ a wider range of mitigations and more clearly identify actual vulnerability data within its narrower purview.

152. One of ordinary skill in the art would have been motivated to combine Gupta and Graham. Both disclosures are directed towards the same problem (network security), including the use of signatures to detect attacks and taking mitigating action. Such a combination is merely applying a known technique (implementing corrective actions for identified vulnerabilities) to a known method (identifying vulnerabilities) with the predictable outcome of identifying and implementing corrective action for the identified vulnerabilities.

153. Further, the combination permits but does not require physical incorporation of elements from Graham into Gupta.

D. Similarity to US 8,984,644

154. Claim 10 of the '063 patent is substantially identical to claim 1 of the '644 patent, with only minor differences. Most differences relate to simple re-phrasings, such as:

- first vulnerability information <--> actual vulnerability information,
- second vulnerability information <--> potential vulnerability information,

Ex.1003

FORCEPOINT LLC/ Page 73 of 116

- vulnerable to at least one accurately identified vulnerability <=> actually vulnerable, and
- one accurately identified vulnerability <=> one actual vulnerability.

'063 Patent - Claim 10	'644 Patent - Claim 1
10. A non-transitory computer-readable media storing instructions that, when executed by one or more processors, cause the one or more processors to:	1. A computer program product embodied on a non-transitory computer readable medium, comprising:
receive first vulnerability information from at least one first data storage that is generated utilizing second vulnerability information from at least one second data storage that is used to identify a plurality of potential vulnerabilities;	code for receiving actual vulnerability information from at least one first data storage that is generated utilizing potential vulnerability information from at least one second data storage that is used to identify a plurality of potential vulnerabilities, by including at least one first potential vulnerability, and at least one second potential vulnerability;
said first vulnerability information generated utilizing the second vulnerability information, by:	said actual vulnerability information generated utilizing the potential vulnerability information, in response to code execution by at least one processor, by:
identifying at least one configuration associated with a plurality of devices including a first device, a second device, and a third device, and	identifying at least one configuration associated with a plurality of devices including a first device, a second device, and a third device, and
determining that the plurality of devices is vulnerable to at least one accurately identified vulnerability based on the identified at least one	determining that the plurality of devices is actually vulnerable to at least one actual vulnerability based on the identified at least one configuration,

'063 Patent - Claim 10	'644 Patent - Claim 1
configuration, utilizing the second vulnerability information that is used to identify the plurality of potential vulnerabilities;	utilizing the potential vulnerability information that is used to identify the plurality of potential vulnerabilities;
	code for identifying an occurrence in connection with at least one of the plurality of devices;
	code for determining that the at least one actual vulnerability of the at least one of the plurality of devices is susceptible to being taken advantage of by the occurrence identified in connection with the at least one of the plurality of devices, utilizing the actual vulnerability information;
display information that is based on the first vulnerability information;	
and cause utilization of different occurrence mitigation actions of diverse occurrence mitigation types, including a firewall-based occurrence mitigation type and an intrusion mitigation system-based occurrence mitigation type, across the plurality of devices for occurrence mitigation by preventing advantage being taken of accurately identified vulnerabilities utilizing the different occurrence mitigation actions of the diverse occurrence mitigation types across the plurality of devices;	and code for providing a user with one or more options to selectively utilize different occurrence mitigation actions of diverse occurrence mitigation types, including a firewall-based occurrence mitigation type and an intrusion prevention system-based occurrence mitigation type, across the plurality of devices for occurrence mitigation by preventing advantage being taken of actual vulnerabilities utilizing the different occurrence mitigation actions of the diverse occurrence mitigation types across the plurality of devices.
receive an indication that an occurrence has been identified in connection with at least one of the plurality of devices utilizing one or more monitors;	

'063 Patent - Claim 10	'644 Patent - Claim 1
wherein the at least one configuration involves at least one operating system.	

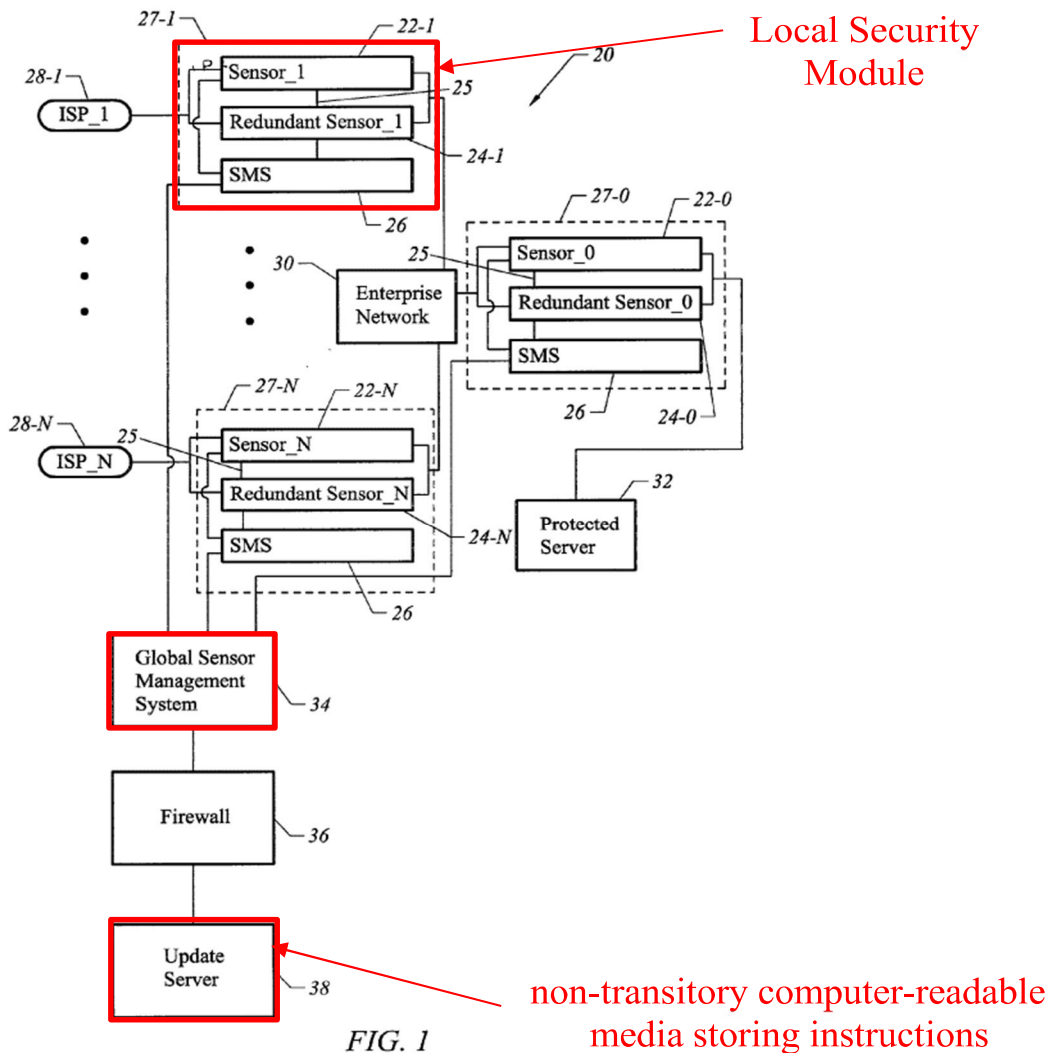
E. Detailed Analysis of Claims

1. Claim 10

- a. [10.0] *A non-transitory computer-readable media storing instructions that, when executed by one or more processors, cause the one or more processors to:*

155. To the extent that the preamble is limiting, Gupta discloses this limitation.

156. First, Gupta discloses a “*non-transitory computer-readable media*” that stores “*instructions.*” For example, Gupta’s FIG. 1 depicts a local security module 27 which includes a sensor 22 and a sensor management system 26, and a global sensor management system 34, all of which are connected to an update server 38. Gupta at FIG. 1, [0032]-[0033]. Gupta further discloses that the update server 38 includes “standard computer components” including a “memory 146” storing “a set of executable programs to implement the functions of the update server.” Gupta, [0150].



Gupta, FIG. 1 (annotated)

157. Second, Gupta discloses the “instructions” stored at the update server 38 are downloaded to the local security module 27 and global sensor management system 34. For example, Gupta discloses the update server 38 “coordinate[s] the delivery” of “**software updates** to the local sensor security module 27.” Gupta, [0034]. Gupta further discloses the update server is “used to notify customers of

new software,” and “provides the **software images** and signature files **for download.”** Gupta, [0138]. It is well known in the art that computing devices can download and install software from a software update server (a “download server downloads updating files to the client” computer and the “desired revised software product is then installed on the client computer”). McGuire, Abstract. Therefore, a POSITA would understand that Gupta’s update server 38 stores software (*instructions*) that is downloadable by local security module 27 and global management system 34.

158. Third, Gupta discloses the (downloaded) “*instructions,*” when “*when executed by one or more processors,*” cause “*the one or more processors*” to perform certain operations. Gupta discloses the “*instructions*” are downloaded from update server 38 by “processor[s] 40_1 through 40_N” of sensor 22, CPU 100 of the sensor management system 26, and CPU 120 of the global management system 34. Gupta, [0036], [0129], [0133]. The processors 40 are connected to the memory 50 (storing the downloaded software from update server 38) via a system bus, and therefore the executable programs in memory 50 are executed by “*one or more processors*” 40. Gupta, [0036]-[0037]. Similarly, Gupta discloses the “sensor management system 26” is in the “form of a general-purpose computer, including a central processing unit 100” and a “memory 106” storing “a set of executable programs utilized to implement features of the invention,” which are

connected via a “system bus 104.” Gupta, [0129]-[0130]. The “global sensor management system 34” is also “in the form of a general-purpose computer, including a central processing unit 120” and a “memory 126.” Gupta, [0133][0134]. Thus, a POSITA would understand that the processors 40, central processing unit 100 or 120 (*one more processors*) executes software *instructions* downloaded from update server 38, respectively.

159. Therefore, Gupta discloses “*a non-transitory computer-readable media storing instructions that, when executed by one or more processors, cause the one or more processors to,*” as claimed.

- b. **[10.1] *receive first vulnerability information from at least one first data storage that is generated utilizing second vulnerability information from at least one second data storage that is used to identify a plurality of potential vulnerabilities;***

160. Gupta renders obvious this limitation.

161. Gupta discloses two data storages, including a hierarchical attach categorization module in a memory (“*second data storage*”) that stores potential vulnerabilities (“*second vulnerability information*”) and a secure update download module in a memory (“*first data storage*”) that stores and provides for download actual vulnerabilities (“*first vulnerability information*”), as described in detail below.

162. First, Gupta discloses “*first vulnerability information*” by disclosing an “attack file.” The “attack file” is generated to identify actual vulnerabilities in the network such as “computer attacks and counter measures relevant” to the network. Gupta, [0164]. The attack file “clearly describe[s], in one data structure, what attacks a given environment is vulnerable to”:

The hierarchy is then traversed to **identify computer attacks and countermeasures relevant to the target platform** (block 162). Detection and protection measures for the target platform are then collected (block 164). This can **result in an attack file** 149. A sensor is then supplied, through a download, with the protective software (e.g., the attack file) for the target platform (block 166).

The hierarchical approach of the invention allows one to **clearly describe, in one data structure, what attacks a given environment is vulnerable to, what protection means (mechanisms and signatures) are available, and how detection alerts are correlated.**

Gupta, [0164]-[0165]. Thus, the “attack file” in Gupta containing the actual vulnerabilities in the network teaches the “*first vulnerability information.*”

163. Gupta discloses that the sensor management system 26 downloads the attack file from a secure update download module 148 in memory 146:

The secure update download module 148 includes executable code to download an attack file 149 to **a sensor management system 26** ...

The attack file 149 can be downloaded using different approaches. The secure update download module 148 can periodically download the attack file 149. Alternately, the secure update download module can download the attack file 149 in response to a request from a sensor management system 26. Alternately, email alerts may also be used to deliver updated attack files 149. The SMS alert notification module 152 may be used to send alerts by secure e-mail (e.g., SMIME) when a new signature update is available for download ...

Gupta, [0151]-[0152]. As the secure update download module 148 in memory 146 provides the attack file for download, a POSITA would understand that the secure update download module 148 in memory 146 maintains a memory unit for storing the attack file. This memory unit in the update download module 148 for storing the attack file discloses “*the at least one first data storage.*”

164. Second, Gupta discloses “*second vulnerability information*” by disclosing a broader set of potential vulnerability information that is used to generate the attack file. Gupta discloses a hierarchical attack categorization module 150 that generates the attack file:

The attack file 149 may be formed utilizing the hierarchical attack categorization module 150 of the invention. The hierarchical attack categorization module 150 is used to develop, deploy and manage information on different computer attacks and countermeasures.

Gupta, [0154]. Gupta further discloses that this hierarchical attack categorization module creates a hierarchy of potential vulnerabilities and attacks; the hierarchy is then used to generate actual vulnerability information for the target platform that is then stored in the attack file:

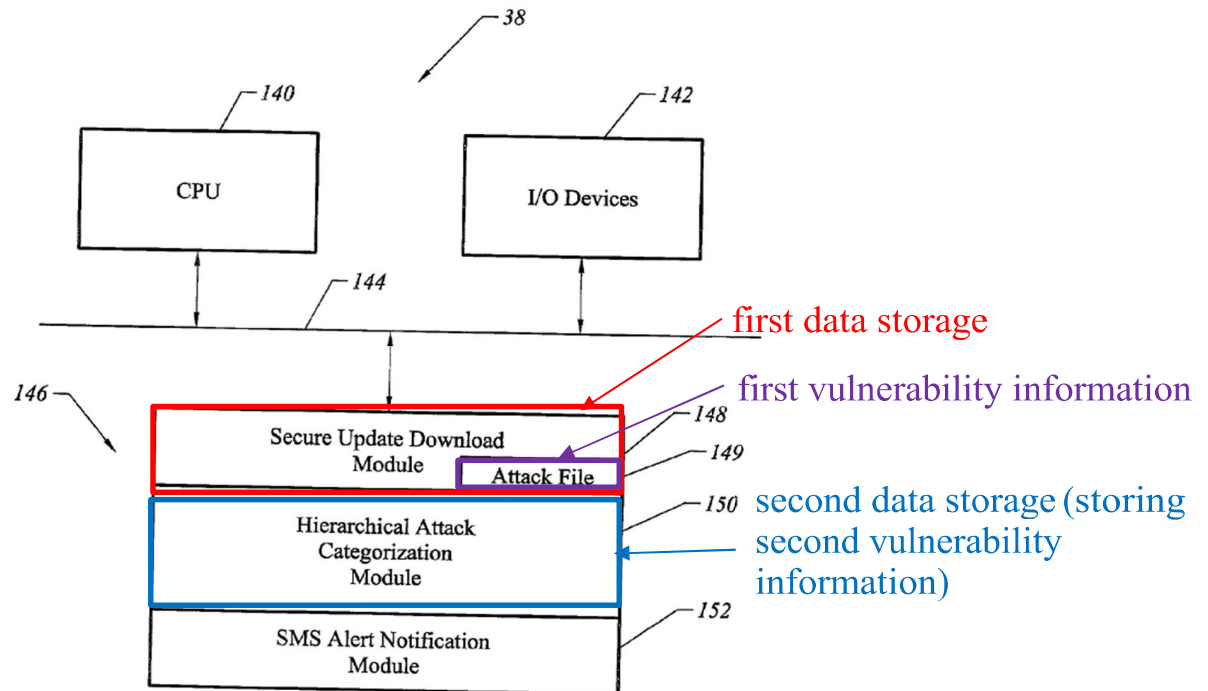
The module **initially** constructs a hierarchy characterizing different computer attacks and countermeasures (block 160) ... The hierarchy is then traversed to identify computer attacks and countermeasures relevant to the target platform (block 162). Detection and protection measures for the target platform are then collected (block 164). This can result in an attack file 149. A sensor is then supplied, through a download, with the protective software (e.g., the attack file) for the target platform (block 166).

The hierarchical approach of the invention allows one to clearly describe, in one data structure, what attacks a given environment is vulnerable to, what protection means (mechanisms and signatures) are available, and how detection alerts are correlated.

Gupta, [0164]-[0165]. The information about “different computer attacks and countermeasures” that is used to generate the “attack file” (“*first vulnerability information*”) in Gupta teaches the “*second vulnerability information*.”

165. As the hierarchical attack categorization module 150 in memory 146 collects information of “different computer attacks and countermeasures,” a POSITA would understand that the hierarchical attack categorization module 150 maintains a memory unit in memory 146 that stores such collected information. This memory unit teaches “*the at least one second data storage that is used to identify a plurality of potential vulnerabilities.*”

166. Third, Gupta discloses receiving the “*first vulnerability information*” from “*first data storage.*” As previously discussed, Gupta discloses that the sensor management system 26 downloads (“*receiving*”) the attack file (“*first vulnerability information*”) from a secure update download module 148 in memory 146 (“*first data storage*”). For example, Gupta’s FIG. 15 shows the attack file 149 is stored at the secure update download module 148 (“*first data storage*”) and “[t]he secure update download module 148 includes executable code to download an attack file 149 to a sensor management system 26.” Gupta, [0151], FIG. 15.



Gupta, FIG. 15 (annotated)

167. As previously discussed in [10.0], update server 38 stores software for sensor management system 26 to download and to be executed by the central processing unit 100. Therefore, a POSITA would understand that Gupta's update server 38 stores programs executed for the central processing unit 100 at sensor management system 26 to receive the "attack file" (*"first vulnerability information"*).

168. Therefore, Gupta renders obvious *"receive first vulnerability information from at least one first data storage that is generated utilizing second*

vulnerability information from at least one second data storage that is used to identify a plurality of potential vulnerabilities,” as claimed.

c. [10.2] said first vulnerability information generated utilizing the second vulnerability information, by:

169. Gupta renders this limitation obvious.

170. As already explained at [10.1], Gupta renders obvious “*first vulnerability information ... that is generated utilizing second vulnerability information.*” Gupta further renders obvious the “*first vulnerability information*” being generated as demonstrated at [10.3] and [10.4] below.

d. [10.3] identifying at least one configuration associated with a plurality of devices including a first device, a second device, and a third device, and

171. Gupta renders obvious this limitation.

172. First, Gupta discloses *identifying at least one configuration* by disclosing an attack file that includes configuration information of a device or a target platform. As explained above, Gupta describes creating an attack file of actual vulnerabilities that are relevant to an individual sensor (for example, the vulnerabilities relevant to target devices in an environment or platform protected by that sensor). *See* Gupta, [164]-[165]. Gupta further explains that the information about potential vulnerabilities (attacks) can be categorized based on various information, such as an “operating system” targeted by an attack:

In addition to defining attack categories, it is useful to define attack target platforms. In one embodiment of the invention, the following hierarchy is identified: (1) vendor (e.g., SUN, IBM, HP, SGI, CISCO), (2) device (e.g., workstation, server, router, firewall), (3) **operating system (e.g., Solaris, Linux, MAC OS)**, and (4) version/revisions.

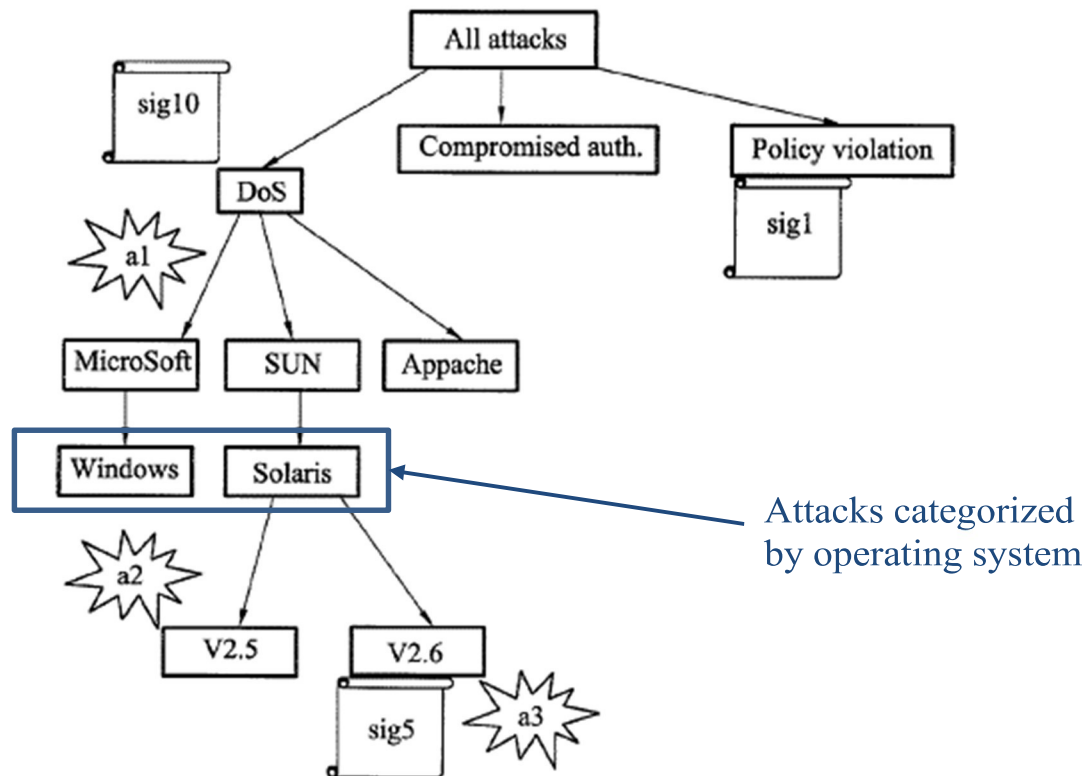
Gupta, [0160].

173. Gupta gives a specific example of a potential attack that “only affects SUN Solaris 2.6,” which a POSITA would have recognized as a specific version of a specific operating system. Gupta, [162]. Defining and categorizing potential attack information based on an “operating system” of potential target devices discloses “*at least one configuration.*”

174. Second, Gupta renders obvious that the operating system “*configuration*” is associated “*with a plurality of devices.*” One of ordinary skill in the art as of July 2003 would readily understand that Gupta would include embodiments wherein the network disclosed therein includes a plurality of at least three different devices. *See, e.g.,* Gupta, [0032] (positioning the sensor security modules 27 between internet service providers and an “enterprise network”), [0057] (distinguishing an “enterprise network” from a “single host/server”), [0068] (“The target system can be a whole enterprise network, an individual server or host

machine ...”). Enterprise networks at that time routinely included hundreds or thousands of computers.

175. Gupta further discloses the “attack file” is generated at the “hierarchical attack categorization module 150” which belongs to the memory 146 of update server 38: “[d]etection and protection measures for the target platform are then collected (block 164),” which “result in an attack file.” Gupta, [0150], [0164]. Gupta notes that some attacks may “appl[y] to all the platforms,” while others may be specific to only some operating systems or specific versions of operating systems. Gupta, [162]. This concept is shown in Figure 16, where some attacks are shown categorized by the operating system (e.g., Windows or Solaris) that they apply to:



Gupta at Fig. 16.

176. Based on Gupta’s statement that the hierarchy of known attacks (like that shown in Fig. 16) is traversed to identify those that are “relevant” to a particular target platform, it would have been obvious to a POSITA for Gupta’s system to identify configuration information such as the operating system of the computers to be protected. See Gupta, [0164]. For example, if an enterprise network to be protected did not have any computers using the SUN Solaris 2.6 operating system, then an attack file for protecting that network would not need to

include information about attack a3 because there would be no computers vulnerable to attack a3. See Gupta, [0162] & Fig. 16.

177. Thus, a POSITA would understand that the resulting “attack file” includes “detection and protection measures for the target platform,” which is based on *identifying... configuration* information of the target platform (e.g., an enterprise network including a “*a plurality of devices*”).

178. Therefore, Gupta renders obvious “*identifying at least one configuration associated with a plurality of devices including a first device, a second device, and a third device,*” as claimed.

- e. **[10.4] *determining that the plurality of devices is vulnerable to at least one accurately identified vulnerability based on the identified at least one configuration, utilizing the second vulnerability information that is used to identify the plurality of potential vulnerabilities;***

179. Gupta renders obvious determining that the plurality of devices is *vulnerable to accurately identified vulnerability* based on the devices’ configuration(s), *utilizing the second vulnerability information*.

180. **First**, Gupta discloses “*at least one accurately identified vulnerability.*” As discussed at [10.1], Gupta discloses an attack file that identifies actual vulnerabilities of a target platform (“*at least one accurately identified vulnerability*”) is generated using Gupta’s hierarchical attack categorization

module 150 storing potential vulnerabilities (“*second vulnerability information*”). For example, Gupta discloses that the update server 38 contains a hierarchical attack categorization module 150, which generates the attack file: “[t]he attack file 149 may be formed utilizing the hierarchical attack categorization module 150 of the invention. The hierarchical attack categorization module 150 is used to develop, deploy and manage information on different computer attacks and countermeasures.” Gupta, [0154].

181. Second, Gupta discloses *determining that a plurality of devices are vulnerable to at least one accurately identified vulnerability based on the identified configuration*. Gupta discloses its system using a hierarchical approach to determine what actual attacks a given environment is vulnerable to:

The hierarchical approach of the invention allows one to clearly describe, in one data structure, **what attacks a given environment is vulnerable to**, what protection means (mechanisms and signatures) are available, and how detection alerts are correlated.

Gupta, [0165]. Gupta further describes how an attack file is generated to contain “computer attacks and countermeasures **relevant** to the target platform.” Gupta, [0164]. Thus, a POSITA would understand that Gupta discloses determining that devices in a given environment are vulnerable to an actual attack (*accurately identified vulnerability*).

182. As previously discussed in [10.3], a POSITA would understand that Gupta's attack file includes information specific to a "target platform" or computing "environment" based on a *configuration associated with a plurality of devices*. The target platform or the computing "environment" are within an "enterprise network 30." Gupta, FIG. 1, [0032]. Therefore, a POSITA would have recognized that the attack file is directed to a target platform that includes different devices within the enterprise network. Thus, a POSITA would understand that Gupta discloses determining the different devices are vulnerable to attacks based on the configuration information. Gupta, [0164] ("identify computer attacks and countermeasures **relevant** to the target platform") & [0165] ("describe... what attacks a given environment is vulnerable to").

183. Therefore, Gupta renders obvious "*determining that the plurality of devices is vulnerable to at least one accurately identified vulnerability based on the identified at least one configuration, utilizing the second vulnerability information that is used to identify the plurality of potential vulnerabilities,*" as claimed.

f. [10.5] display information that is based on the first vulnerability information;

184. Gupta discloses this limitation.

185. Gupta discloses a “graphical user interface (GUI)” that “*display[s]*” intrusion information (“*first vulnerability information*”) for a user to view, as described below.

186. As discussed previously in [10.1], Gupta discloses “*first vulnerability information*” by disclosing an “attack file” that identifies “computer attacks and counter measures relevant to the target platform (block 162)” and “clearly describe[s], in one data structure, what attacks a given environment is vulnerable to.” Gupta, [0164]-[0165].

187. Gupta renders obvious that information in the “attack file” is displayed to a human user to review. Gupta discloses that users such as an “intrusion expert who is a person” is “responsible for configuration and monitoring of intrusion related matters.” Gupta, [0142]. To monitor intrusion related matters, the memory 126 at the global sensor management system 34 contains a virtual intrusion detection system (VIDS) 130. Gupta, [0133], [0134]. Specifically, the VIDS 130 has a graphical user interface (GUI) for a user to monitor the VIDS: “VIDS provisioning module 130 implements a graphical user interface (GUI). The GUI enables the subscriber to access the sensor management system 34 to manage and monitor the VIDS. The GUI can be implemented within a browser, either as an applet or using HTML.” Gupta, [0131], [0143]. Gupta further discloses the VIDS displays intrusion information for a subscriber to view,

e.g., “VIDS opens a window for subscribers to view information related to their own systems” and “[a] VIDS view restricts the intrusion information shared with a subscriber to the subset related to the subscriber configuration.” Gupta, [0137], [0144].

188. Gupta renders obvious that the displayed information on the GUI is “*based on*” the attack file (“*first vulnerability information*”). For example, a POSITA would find it obvious that Gupta’s intrusion information displayed for a user to view includes information relating to computer attacks to the network. As discussed previously in [10.1], Gupta’s attack file identifies “computer attacks and counter measures relevant to the target platform.” Gupta, [0164]. Thus, Gupta’s displayed intrusion information is “*based on*” at least the computer attack information in the attack file.

189. It would have been obvious to POSITA for a user desiring to view “configuration” information for Gupta’s system to access and view the attack file (including *first vulnerability information*) that was downloaded to a sensor. See Gupta, [0142], [137]. For example, Gupta notes that the “intrusion signature and response configuration” is accessible via VIDS. Gupta, [0139]. Such configuration information is provided as part of an attack file:

The attack file 149 specifies attacks and counter measures.

Preferably, the file also identifies unknown attacks and

suggests responses for such attacks. The attack file 149 includes information forming the **intrusion signatures** 70 processed by the classification and pattern-matching module 68.

Gupta, [0151].

190. Therefore, a POSITA would understand that Gupta's VIDS 130 in the global sensor management system 34 displays intrusion information from the attack file (containing *first vulnerability information*) via a GUI for a user to view.

191. As previously discussed in [10.0], update server 38 stores software for the global sensor management system 34 to download and to be executed by the central processing unit 120 to implement Gupta's features. Therefore, a POSITA would understand that Gupta's update server 38 stores programs executed by the central processing unit 120 to display intrusion information via the GUI.

192. Therefore, Gupta renders obvious "*display[ing] information that is based on the first vulnerability information.*"

- g.** **[10.6] cause utilization of different occurrence mitigation actions of diverse occurrence mitigation types, including a firewall-based occurrence mitigation type and an intrusion prevention system-based occurrence mitigation type, across the plurality of devices for occurrence mitigation by preventing advantage being taken of accurately identified vulnerabilities utilizing the different occurrence mitigation actions of the diverse occurrence mitigation types across the plurality of devices;**

193. Gupta in combination with Graham renders obvious this limitation.

194. **First**, Gupta discloses “*caus[ing] utilization of different occurrence mitigation actions.*” Gupta discloses providing a user with the ability to select among different options of security policies and preferences. Gupta discloses that the VIDS in the global sensor management system 34 provides a user options to configure and manage the security system: “[p]referably, the VIDS provisioning module 130 implements a graphical user interface (GUI). The GUI enables the subscriber to access the sensor management system 34 to manage and monitor the VIDS. The GUI can be implemented within a browser, either as an applet or using HTML.” Gupta, [0131], [0143]. Gupta further discloses:

The VIDS provisioning module 130 facilitates protection for multiple traffic streams for different subscribers using a single sensor platform while maintaining security separation. The module 130 supports **different security policies, protection objectives, and security management preferences** effectively and efficiently. The module 130 reduces the management task and total cost of providing intrusion detection security services.

Gupta, [0136].

195. The availability of multiple “security policies, protection objectives, and security management preferences” in Gupta’s system would have suggested to

a POSITA that Gupta's system would be able to apply a variety of *different occurrence mitigation actions* in response to different threats and attacks.

196. Second, Gupta discloses "*diverse occurrence mitigation types*." Gupta discloses providing different options of occurrence mitigation actions, including terminating TCP connections, which a POSITA would understand to be an "*intrusion prevention system-based occurrence mitigation type*" as recited in this claim. For example, Gupta discloses that memory 50 of the local sensor 22 includes a "response processor 54." Gupta, FIG. 2. The response processor 54 "generates alerts that provide notification of an attack." Gupta, [0087]. Gupta further discloses permitting system administrators to configure response processors to respond to detected occurrences:

The overall processing is supervised through sensor management 86, which is implemented with the sensor management module 52. A response module handles response processing 85. In particular, the response processor 54 determines the response actions for the specific attack. **The response processor 54 is configured by the system administrator.** Once configured, the response processor 54 responds to specific attacks.

Gupta, [0048].

197. Gupta's response processor 54 may then utilize one or more mitigation actions to prevent the attack. In particular, the response processor can respond with an IPS-based mitigation action of closing a connection:

The response processor 54 responds to AIDs. The response processor 54 generates alerts that provide notification of an attack. **The response processor 54 also attempts to prevent the attack. Short-term responses include terminating TCP connections.** Long-term responses include packet logging for further analysis to improve detection and response.

Gupta, [0087].

198. It would have been further obvious to a POSITA to respond with other action types in order to address an attack, including a firewall-based mitigation action. Gupta's sensor 22 includes a "fixed-field detector 63" that "[o]nce the packet is received by the sensor 22," scans the packet "for attacks that leave characteristic traces in certain specific fields." Gupta, [0084], [0101]. Specifically, for a "LAND attack" which "consists of streams of TCP SYN packets whose source IP address and TCP port number are set equal to the destination IP address and TCP port number," the fixed-field detector can scan the "source and destination address fields" to detect the attack. Gupta, [0101]. "A web server suddenly seeing a lot of requests from new IP addresses may also be considered an anomaly."

[0051]. A POSITA would understand that the best way to prevent such DOS attacks is to drop these packets at the Firewall.

199. Graham also discloses “*firewall-based occurrence mitigation type.*”

Graham discloses providing occurrence mitigation actions, such as firewall techniques, that a POSITA would recognize would beneficially be utilized among the plurality of mitigation options provided in Gupta. Graham discloses responding to detected occurrences, including the user of firewall-based mitigation actions:

If the system determines that the target is not vulnerable, one embodiment of the system generates a low-level alert and/or simply logs the event signature (or may take no action whatsoever, depending on the configuration). **If the system determines that the target is vulnerable,** however, then one embodiment will generate an alert condition (or increase a preexisting alert level) and/or **may take certain precautionary measures** (at 235). For example, at this stage **the system may attempt to block incoming data transmissions from the suspect node (e.g., by commanding the firewall to do so).**

Graham at 7:58-67.

200. It would have been obvious to a POSITA to combine Gupta and Graham, as both teach the use of multiple, known mitigation techniques and the benefit of using the technique that best fits the circumstances of a given occurrence

or device. A POSITA would have recognized the benefit of combining Graham's additional mitigation technique, such as firewall-based precautionary measures to block all incoming traffic from a suspect node, with Gupta's provision of multiple mitigation options.

201. As previously discussed in [10.0], update server 38 stores software for local security module 27 or the global sensor management system 34 to download and to be executed by the processors 40 or central processing unit 120. Therefore, a POSITA would understand that Gupta's update server 38 stores programs that are downloaded to the local sensor 22 to be executed by *one or more processors to cause utilization of different occurrence mitigation actions of diverse occurrence mitigation types* at the response processor 54 (part of sensor 22, see Gupta, [0036], [0038], [0049]).

202. Therefore, Gupta in combination with Graham renders obvious "*allow selective utilization of different occurrence mitigation actions of diverse occurrence mitigation types, including a firewall-based occurrence mitigation type and an intrusion prevention system-based occurrence mitigation type, across the plurality of devices for occurrence mitigation by preventing advantage being taken of accurately identified vulnerabilities utilizing the different occurrence mitigation actions of the diverse occurrence mitigation types across the plurality of devices,*" as claimed.

- h.** ***[10.7] receive an indication that an occurrence has been identified in connection with at least one of the plurality of devices utilizing one or more monitors;***

203. Gupta discloses this limitation.

204. **First**, Gupta discloses detecting an attack against a device within a network, which a POSITA would understand to be an “*occurrence in connection with at least one of the plurality of devices.*”

205. Gupta discloses performing intrusion detection to identify an occurrence. According to Gupta,

The anomaly detector 62 is one mechanism utilized by the invention to prevent attacks. Another mechanism utilized in accordance with the invention is signature processing. The signature processing system detects attacks by looking for signatures that are specific combinations of patterns (e.g., numerical field values, string matches, and the like) existing in monitored network traffic.

Gupta, [0083].

206. Gupta provides additional details about how these signatures lead to the identification of an AttackID (AID):

Both the fixed-field detector and the protocol parser 64 generate Signature Sub-IDs (SSIDs) that indicate the presence of subsets of the patterns required for detection of a particular signature. The classification and pattern matching module 68 accepts the SSIDs in the order in

Ex.1003

FORCEPOINT LLC/ Page 100 of 116

which they occur and determines if all the SSIDs required to form the complete pattern of a signature occurred. The pattern signatures are stored as intrusion signatures 70. For each complete signature, the classification and pattern matching module 68 raises an AttackID (AID) that indicates the attack that was found in the communications being monitored.

Gupta, [0086].

207. As discussed above at [10.1], Gupta describes using signatures from an attack file that identify attempts to take advantage of a vulnerability in connection with devices being protected on the network.

208. Further, Gupta teaches that these actions, with respect to individual sensors 22, are “used to implement a shared-resource virtual intrusion detection system.” Gupta, [0031]. Gupta further teaches implementing a “global shared-resource virtual intrusion detection system” with the “global sensor management system 34.” Gupta, [0033]. Gupta contemplates that multiple sensors are used to protect an enterprise network. For example, Gupta refers to “Sensors 27_1 through 27_N [that] monitor the links that connect the enterprise network 30 to Internet service providers.” Gupta, [0145]. Each of Gupta’s sensors act as a “*network monitor*,” thus the multiple sensors render obvious “*one or more network monitors*.”

209. Therefore, Gupta discloses that “*an occurrence has been identified in connection with at least one of the plurality of devices, utilizing one or more network monitors,*” as claimed.

210. Second, Gupta also discloses *receiving an indication* about the identified occurrence. Gupta discloses the global sensor management system 34 receives *an indication* of attacks detected by the sensor 22 in sensor module 27. For example, Gupta’s FIG. 1 discloses a “global sensor management system 34 [that] **tracks information from** the local sensor security modules 22 [sic, 27] to identify and **respond to distributed denial of service attacks.**” Gupta, [0033]. Therefore, Gupta’s global sensor management system 34 receives at least an indication of “distributed denial of service attacks” (*occurrence ... in connection with at least one of the plurality of devices*) that have been *identified* by the sensor 22.

211. As previously discussed in [10.0], update server 38 stores software for global sensor management system 34 to download and to be executed by the central processing unit 120. Therefore, a POSITA would understand that Gupta’s update server 38 stores programs executed by the central processing unit 120 to receive an indication relating to attacks *identified* by the sensor 22.

212. Therefore, Gupta discloses “*receiving an indication that an occurrence has been identified in connection with at least one of the plurality of devices utilizing one or more monitors.*”

- i.** **[10.8] *wherein the at least one configuration involves at least one operating system.***

213. As noted at [10.3], the configuration may be an operating system configuration for target platforms. See Gupta, [0160] (target platforms identified, including “operating system”). Therefore, Gupta discloses [10.8].

2. Claim 11

- a.** **[11.0] *The non-transitory computer-readable media of claim 10, wherein the instructions are configured such that the utilization of the different occurrence mitigation actions is caused for different devices prior to the occurrence.***

214. Gupta discloses this limitation.

215. As previously discussed in [10.6], Gupta in combination with Graham renders obvious utilizing the different occurrence mitigation actions of the diverse occurrence mitigation types across the plurality of devices. And as noted in [10.5], Gupta renders obvious a user interface through which its sensor system—including its “different security policies, protection objectives, and security management preferences”—may be configured by an administrator. See Gupta, [0136]-[0138]. Based on Gupta’s description of permitting the user to configure policies, objectives, and preferences, it would have been obvious to a POSITA for Gupta’s

Ex.1003

system to allow the user to configure different responses (“*occurrence mitigation actions*”) before an actual attack occurs (“*prior to the occurrence*”).

216. Gupta further discloses the *utilization of* at least one *occurrence mitigation action prior to the occurrence*. Specifically, Gupta’s sensor 22 includes a “fixed-field detector 63” that “[o]nce the packet is received by the sensor 22,” scans the packet “for attacks that leave characteristic traces in certain specific fields.” Gupta, [0084], [0101]. Specifically, for a “LAND attack” which “consists of streams of TCP SYN packets whose source IP address and TCP port number are set equal to the destination IP address and TCP port number,” the fixed-field detector can scan the “source and destination address fields” to detect the attack. Gupta, [0101]. Thus, Gupta’s fixed-field detector that scans **every** received **packet** for potential attacks discloses configuring *utilization of* a mitigation action “*prior to the occurrence*” of an attack.

217. As previously discussed in [10.0], update server 38 stores software for the local sensor 22 to download to implement features of the invention. A POSITA would understand that software *instructions* downloaded by the sensor 22 configure Gupta’s fixed-field detector to perform such scanning *prior to the* LAND attack.

218. Graham also discloses causing utilization of different occurrence mitigation actions, which a POSITA would have understood to be configured before the occurrence:

In one embodiment of the invention, one or more of the nodes—e.g., node 132—monitors network traffic across the LAN 140. When the node 132 identifies an incident directed to a particular client or server—e.g., node 134 it **may (or may not) log the incident, generate an alert condition and/or take certain precautionary measures.** The type and extent of action taken by the node 132 may depend on a variety of factors including, but not limited to, the type of incident detected; the “type” of target to which the incident is directed (e.g., the processor in the target, version of software running in the target, ... etc); the response of the target; the firewall response, if any (e.g., whether the incident has been filtered by the firewall), and/or the behavior of the target following the incident. Each of these variables, alone or in combination, may dictate the type and extent of a response.

Graham, 4:44-58.

219. For example, Graham states that a determination of which response or responses to take in response to an incident (“*occurrence*”) may depend on detailed information about the target. Graham explains that such “fingerprint” information may be gathered beforehand, for example, by receiving input from an administrator

Ex.1003

FORCEPOINT LLC/ Page 105 of 116

or by scanning devices over the network to gather information about them.

Graham, 6:43-51. It would have been obvious to a POSITA that fingerprint information is gathered “*prior to the occurrence*” because the information is used by Graham’s system to decide how (or whether) to respond to a potential incident:

Regardless of how/when the target fingerprint data is collected (determined at 215 and/or 220 of FIG. 2) one embodiment of the system correlates the fingerprint data with the context-based and/or state-based data signature (as indicated at 255) to determine whether the target is actually vulnerable to the suspicious data signature (at 230).

220. Graham further explains how the fingerprint data is used to configure a system to provide different responses to an attack or incident directed to different targets:

For example, only certain target configurations are vulnerable to the "phf" data signature described above. More specifically, the Apache.TM. server with a version number below 1.2 is vulnerable to the "phf" attack. Thus, **if the target fingerprint indicates Apache version of 1.2.4, then the system may only generate a low-level alert (or no alert) upon detecting the "phf" signature. If, however, the target is running version 1.0 of Apache, then the target is vulnerable and the system may generate a more severe alert condition.**

Similarly, the signature "09090909 . . . " is considered a suspicious signature because it represents code used to execute "no-ops" on an Intel.TM. processor (e.g., a Pentium.TM.-class processor). Although "no-ops" have certain legitimate uses for software developers, they may also be used by intruders to break into and/or disable Intel-based systems under certain circumstances. As such, **if the target contains an Intel processor and a "no-op" data signature is identified, then an alert condition may be generated** because the target is vulnerable to the "no-op" attack. However, **if the target is configured with a non-Intel processor (e.g., a Sun Sparc.TM. processor), then the target is not vulnerable, and the system may (or may not) merely log the event.**

Graham, 7:19-44.

221. Therefore, Gupta in combination with Graham renders obvious [11.0].

3. Claim 39

a. [39.0] *A non-transitory computer-readable media storing instructions that, when executed by one or more processors, cause the one or more processors to:*

222. See analysis at [10.0].

b. [39.1] *receive first vulnerability information from at least one first data storage that is generated utilizing second vulnerability information from at least one second data storage that is used to identify a plurality of potential vulnerabilities,*

223. See analysis at [10.1].

Ex.1003

- c. ***[39.2] by including: at least one first potential vulnerability, and at least one second potential vulnerability;***

224. Gupta discloses potential vulnerability information for a plurality of potential vulnerabilities. For example, Gupta discloses “[t]he hierarchical attack categorization module 150 is used to develop, deploy and manage information on different computer attacks and countermeasures.” Gupta, [0154].

225. Therefore, Gupta discloses claim element [39.2].

- d. ***[39.3] said first vulnerability information generated utilizing the second vulnerability information, by:***

226. See analysis at [10.2].

- e. ***[39.4] identifying at least one configuration associated with a plurality of devices including a first device, a second device, and a third device, and***

227. See analysis at [10.3].

- f. ***[39.5] determining that the plurality of devices is actually vulnerable to at least one actual vulnerability based on the identified at least one configuration, utilizing the second vulnerability information that is used to identify the plurality of potential vulnerabilities;***

228. See analysis at [10.4]. Claim element [39.5] differs because it recites: “actually vulnerable” in place of “vulnerable,” and at least one “actual vulnerability” in place of “accurately identified vulnerability.” A POSITA would understand that the same analysis stated in [10.4] is applicable to [39.5] despite the minor language differences. For example, because Gupta discloses “determining

that the plurality of devices is vulnerable to ... accurately identified vulnerability,” a POSITA would understand that Gupta similarly discloses “*determining that the plurality of devices is actually vulnerable ... actual vulnerability.*” For the same reasons stated in [10.4], Gupta renders obvious [39.5].

- g. [39.6] *identify an occurrence in connection with at least one of the plurality of devices, utilizing one or more monitors;***

229. Gupta discloses that an occurrence has been identified in connection with at least one of the plurality of device utilizing one or more monitors, as discussed above in [10.7].

230. Therefore, Gupta discloses this limitation.

- h. [39.7] *based on a packet analysis, determine that the at least one accurately identified vulnerability of the at least one of the plurality of devices is susceptible to being taken advantage of by the occurrence identified in connection with the at least one of the plurality of devices, utilizing the first vulnerability information; and***

231. Gupta discloses this limitation.

232. Gupta discloses using the target-specific attack file (“*the first vulnerability information*”) to determine if a monitored device is susceptible to being taken advantage of by the presently occurring threat (“*occurrence*”). For example, Gupta discloses:

The anomaly detector 62 is one mechanism utilized by the invention to prevent attacks. Another mechanism utilized

in accordance with the invention is signature processing.

The signature processing system detects attacks by looking for signatures that are specific combinations of patterns (e.g., numerical field values, string matches, and the like) **existing in monitored network traffic**.

Gupta, [0083]. Because the detected occurrence in network traffic may take advantage of at least one accurately identified vulnerability, a POSITA would understand that Gupta discloses determining if an accurately identified vulnerability of a device is susceptible to an “*occurrence*” based on “*first vulnerability information*.”

233. For example, Gupta discloses “[t]he attack file 149 may be formed utilizing the hierarchical attack categorization module 150 of the invention. The hierarchical attack categorization module 150 is used to develop, deploy and manage information on different computer attacks and countermeasures.” Ex.1006, [0154]. In addition, Gupta discloses:

The module initially constructs a hierarchy characterizing different computer attacks and countermeasures (block 160). ... The hierarchy is then traversed to identify computer attacks and countermeasures relevant to the target platform (block 162). Detection and protection measures for the target platform are then collected (block 164). This can result in an attack file 149. **A sensor is then supplied, through a download, with the protective**

Ex.1003

software (e.g., the attack file) for the target platform (block 166).

The hierarchical approach of the invention allows one to clearly describe, in one data structure, **what attacks a given environment is vulnerable to,** what protection means (mechanisms and signatures) are available, and how detection alerts are correlated.

Gupta, [0164]-[0165].

234. Moreover, Gupta explains that the sensors use their attack files (target specific) to respond to attacks:

FIG. 17 illustrates processing steps performed by an embodiment of the hierarchical attack categorization module 150 of the invention. The module initially constructs a hierarchy characterizing different computer attacks and countermeasures (block 160). This operation results in a structure of the type shown in FIG. 16. The hierarchy is then traversed to identify computer attacks and countermeasures relevant to the target platform (block 162). **Detection and protection measures for the target platform are then collected (block 164).** This can result in an attack file 149. A sensor is then supplied, through a download, with the protective software (e.g., the attack file) for the target platform (block 166).

Gupta, [0164].

235. Gupta further discloses *packet analysis* to generate the attack file.

Gupta teaches that the attack file generated based on the collected detection and protection measures, and “the update server 38 can be configured to update both sensors 22 and redundant sensors 24 with updated attack signatures, anomaly profiles, and the like.” Gupta, [0166]. The threat detection is implemented by the sensors for the “*packet analysis*,” based on which the system determines when “*at least one of the plurality of devices is susceptible to being taken advantage of by the occurrence*” using the signatures (“*utilizing the first vulnerability information*”).

236. It would have been obvious to a POSITA that “monitored network traffic” would include “*packet analysis*” with which to make the determination. For example, Gupta teaches generating SSIDs in response to comparing signatures to the content of packets (thus, an example of packet analysis). *See* Gupta, [0086]. A POSITA would understand that such signature comparison needs to be conducted by analyzing packet contents. Therefore, based on this packet analysis, the system determines when “*at least one of the plurality of devices is susceptible to being taken advantage of by the occurrence*” using the signatures (“*utilizing the first vulnerability information*”).

237. Therefore, Gupta discloses “*based on a packet analysis, determine that the at least one accurately identified vulnerability of the at least one of the*

plurality of devices is susceptible to being taken advantage of by the occurrence identified in connection with the at least one of the plurality of devices, utilizing the first vulnerability information,” as claimed.

- i. ***[39.8] permit selective utilization of different occurrence mitigation actions of diverse occurrence mitigation types, including a firewall-based occurrence mitigation type and a other occurrence mitigation type, across the plurality of devices for occurrence mitigation by preventing advantage being taken of actual vulnerabilities utilizing the different occurrence mitigation actions of the diverse occurrence mitigation types across the plurality of devices;***

238. Gupta in combination of Graham renders obvious this limitation.

239. Claim element [39.8] differs from [10.6] because it recites: “*permit selective utilization*” in place of “cause utilization” of difference occurrence mitigation actions, “*a other*” in place of “an intrusion mitigation system-based” occurrence mitigation type, and “*actual vulnerabilities*” in place of “accurately identified vulnerabilities.”

240. As discussed in [10.6], Gupta renders obvious to *cause utilization of different occurrence mitigation actions of diverse mitigation types* carried out by the local sensor 22. Gupta also renders obvious to “*permit selective utilization*” of these different occurrence mitigation actions. For example, as previously discussed in [10.6], Gupta discloses different options of occurrence mitigation actions, including at least short-term responses to terminate TCP connections or

long-term responses include packet logging at the response processor 54, or scanning every packet for attacks by the fix-field detector 63. A POSITA would understand that these different types of mitigation actions are not all carried out at the same time (e.g., when TCP connection is terminated, there is no packet arriving at the fix-field detector). It would have been obvious to a POSITA that when only a subset of Gupta's options of mitigation actions are carried out, Gupta is permitting selective utilization of its mitigation actions.

241. As also previously discussed in [10.6], Gupta's VIDS 130 permits a user to configure and select "security management preferences." Therefore, a POSITA would find it obvious that Gupta *permits* a user to select a security preference from different options, such as different options of occurrence mitigation actions, i.e., *selective utilization of different occurrent mitigation actions*.

242. And as discussed at [11.0], Graham describes responding to an incident in different ways based in part on fingerprint information about a target of the incident. Graham's tailoring of a response to the fingerprint (e.g., configuration) of an attack target also shows the "*selective utilization of different occurrence mitigation actions*."

243. Therefore, Gupta in combination of Graham renders obvious "*permit[ting] selective utilization of different occurrence mitigation actions of*

diverse occurrence mitigation types, including a firewall-based occurrence mitigation type and a other occurrence mitigation type, across the plurality of devices for occurrence mitigation by preventing advantage being taken of actual vulnerabilities utilizing the different occurrence mitigation actions of the diverse occurrence mitigation types across the plurality of devices.”

- j. [39.9] wherein the at least one configuration involves at least one operating system.**

244. See analysis at [10.8].

4. Claim 58

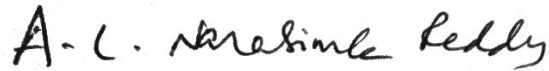
- a. [58.0] A non-transitory computer-readable media storing instructions that, when executed by one or more processors, cause the one or more processors to:**

245. See analysis at [11.0].

X. CONCLUSION

246. I 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, and 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 section 1001 of Title 18 of the United States Code.

Dated: 7 June, 2023

Handwritten signature of A. C. Narasimha Reddy in black ink.

Narasimha Reddy, Ph.D.