

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Fortinet, Inc.,

Petitioner,

v.

Netskope, Inc.,

Patent Owner.

Case No. 2026-00027

U.S. Patent 7,593,936

PETITION FOR *INTER PARTES* REVIEW

Mail Stop "PATENT BOARD"

Patent Trial and Appeal Board

U.S. Patent Trademark Office

P.O. Box 1450

Alexandria, VA 22313-1450

TABLE OF CONTENTS

	<u>Page</u>
TABLE OF ABBREVIATIONS	x
EXHIBIT LIST	xi
I. INTRODUCTION	1
II. GROUNDS FOR STANDING (37 C.F.R. §42.104(a)).....	1
III. THE PATENT AND STATE OF THE ART	2
A. Computer Malware and Other Attacks	2
B. Computer Security	3
C. Network and System Administration	4
D. The '936 Patent	5
E. The Challenged Claims	7
F. File History.....	8
G. POSITA.....	10
IV. Claim Construction.....	10
V. SUMMARY OF CHALLENGE AND RELIEF REQUESTED.....	12
A. 37 C.F.R. §42.104(b)(1): The Challenged Claims	12
B. 37 C.F.R. §42.104(b)(1): Prior Art Overview and Specific Grounds of Rejection	12
1. Honig.....	13
2. Oliphant.....	13
3. Vinberg	23
C. Grounds of Rejection	23

VI.	 GROUND S 1 AND 2 – CLAIMS 1, 7-12, 17-19, 21-22 ARE INVALID AS ANTICIPATED OR RENDERED OBVIOUS BY HONIG 	 24
A.	Overview of Honig.....	24
B.	Claim 1	26
	1. [1PRE]: "A method of detecting abnormal system states in computers, comprising:"	26
	2. [1A]: "receiving snapshots from a plurality of computers within a population of computers, wherein individual snapshots include data indicating a state of a respective computer;"	26
	3. [1B]: "automatically generating an adaptive reference model comprising a rule set customized to characteristics of the population of computers, the rule set being developed by identifying patterns among the snapshots from the plurality of computers such that the adaptive reference model is indicative of normal states in the computers within the population; and"	30
	4. [1C]: "comparing a snapshot from at least one of the computers to the adaptive reference model to determine whether an anomaly is present in the state of the at least one of the computers."	33
C.	Claim 7: "The method of claim 1, wherein individual snapshots include data associated with at least one of: system files, application files, a registry entry, a performance counter, a process, a communication port, a hardware configuration, a log file, a running task, services, and network connections."	35
D.	Claim 8: "The method of claim 1, further comprising: inserting rules into the rule set of the adaptive reference model to augment or override rules of the rule set automatically generated from the snapshots."	35

E.	Claim 9: "The method of claim 1, wherein the adaptive reference model is generated to include a value layer that determines whether an asset value contained in a snapshot is anomalous."	36
F.	Claim 10: "The method of claim 1, wherein the adaptive reference model is generated to include a cluster layer that tracks relationships between assets and identifies an anomaly in response to an asset being unexpectedly absent from or present in a set of assets in a snapshot."	37
G.	Claim 11: "The method of claim 1, wherein the adaptive reference model is generated to include a profile layer that identifies anomalies in response to violation of relationships of clusters of assets in a snapshot."	38
H.	Claim 12	39
I.	Claims 17-19, 21-22	40
VII.	GROUND 3 – CLAIMS 2-6, 13-16, 20 ARE INVALID AS RENDERED OBVIOUS BY HONIG OVER OLIPHANT	40
A.	Overview of Oliphant	40
B.	Motivation to combine Honig and Oliphant	41
C.	Claim 2: "The method of claim 1, further comprising: comparing the anomaly to a recognition filter to diagnose a trouble condition on the at least one of the computers; and in the event of a trouble condition, generating an automated response to the trouble condition."	45
D.	Claim 3: "The method of claim 2, wherein the automated response is a generic response not specific to a particular asset of the at least one computer, and wherein the method further comprises: sending the generic response and a set of anomalies found in the snapshot to the at least one computer, the set of anomalies indicating assets of the at least one computer whose states are anomalous; and applying the generic response to the anomalous assets to correct the trouble condition."	46

E.	Claim 4: "The method of claim 3, wherein the generic response includes at least one of: installing a missing software component, removing an undesirable software component, and restoring an incorrect registry setting."	47
F.	Claim 5: "The method of claim 2, wherein the recognition filter comprises a particular pattern of anomalies that indicates the presence of a particular root cause condition or a generic class of conditions."	47
G.	Claim 6: "The method of claim 1, further comprising: comparing a plurality of anomalies associated with a particular snapshot with a recognition filter to diagnose a trouble condition; and diagnosing a trouble condition on the at least one of the computers in response to at least a subset of the plurality of anomalies matching information in the recognition filter."	48
H.	Claims 13-16, 20	49
VIII.	GROUND 4 – CLAIMS 1, 7-12, 17-19, 21-22 ARE INVALID AS RENDERED OBVIOUS BY vinberg	49
A.	Overview of Vinberg.....	49
B.	Claim 1	51
1.	[1PRE]: "A method of detecting abnormal system states in computers, comprising:"	51
2.	[1A]: "receiving snapshots from a plurality of computers within a population of computers, wherein individual snapshots include data indicating a state of a respective computer;"	51
3.	[1B]: "automatically generating an adaptive reference model comprising a rule set customized to characteristics of the population of computers, the rule set being developed by identifying patterns among the snapshots from the plurality of computers such that the adaptive reference model is indicative of normal states in the computers within the population; and"	52

4.	[1C]: "comparing a snapshot from at least one of the computers to the adaptive reference model to determine whether an anomaly is present in the state of the at least one of the computers."	53
C.	Claim 7: "The method of claim 1, wherein individual snapshots include data associated with at least one of: system files, application files, a registry entry, a performance counter, a process, a communication port, a hardware configuration, a log file, a running task, services, and network connections."	54
D.	Claim 8: "The method of claim 1, further comprising: inserting rules into the rule set of the adaptive reference model to augment or override rules of the rule set automatically generated from the snapshots."	54
E.	Claim 9: "The method of claim 1, wherein the adaptive reference model is generated to include a value layer that determines whether an asset value contained in a snapshot is anomalous."	55
F.	Claim 10: "The method of claim 1, wherein the adaptive reference model is generated to include a cluster layer that tracks relationships between assets and identifies an anomaly in response to an asset being unexpectedly absent from or present in a set of assets in a snapshot."	56
G.	Claim 11: "The method of claim 1, wherein the adaptive reference model is generated to include a profile layer that identifies anomalies in response to violation of relationships of clusters of assets in a snapshot."	57
H.	Claim 12	58
I.	Claims 17-19, 21-22	59
IX.	GROUND 5 – CLAIMS 2-6, 13-16, 20 ARE INVALID AS RENDERED OBVIOUS BY Vinberg OVER OLIPHANT	60
A.	Motivation to combine Vinberg and Oliphant	60

B.	Claim 2: "The method of claim 1, further comprising: comparing the anomaly to a recognition filter to diagnose a trouble condition on the at least one of the computers; and in the event of a trouble condition, generating an automated response to the trouble condition."	63
C.	Claim 3: "The method of claim 2, wherein the automated response is a generic response not specific to a particular asset of the at least one computer, and wherein the method further comprises: sending the generic response and a set of anomalies found in the snapshot to the at least one computer, the set of anomalies indicating assets of the at least one computer whose states are anomalous; and applying the generic response to the anomalous assets to correct the trouble condition."	64
D.	Claim 4: "The method of claim 3, wherein the generic response includes at least one of: installing a missing software component, removing an undesirable software component, and restoring an incorrect registry setting."	65
E.	Claim 5: "The method of claim 2, wherein the recognition filter comprises a particular pattern of anomalies that indicates the presence of a particular root cause condition or a generic class of conditions."	65
F.	Claim 6: "The method of claim 1, further comprising: comparing a plurality of anomalies associated with a particular snapshot with a recognition filter to diagnose a trouble condition; and diagnosing a trouble condition on the at least one of the computers in response to at least a subset of the plurality of anomalies matching information in the recognition filter."	66
G.	Claims 13-16, 20	67
X.	MANDATORY Notices Under 37 C.F.R. §42.8(a)(1)	67
A.	37 C.F.R. §42.8(b)(1): Real Parties-In-Interest.....	67
B.	37 C.F.R. §42.8(b)(2): Related Matters	67
C.	37 C.F.R. §42.8(b)(3)-(4): Lead And Back-Up Counsel And Service Information.....	68

XI.	FEES UNDER 37 C.F.R. §42.103	68
XII.	CONCLUSION	68

TABLE OF AUTHORITIES

Cases

<i>Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.</i> , 800 F.3d 1375 (Fed. Cir. 2015)	13
<i>New Railhead Mfg., L.L.C. v. Vermeer Mfg. Co.</i> , 298 F.3d 1290 (Fed. Cir. 2002)	13
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005)	11

Statutes

35 U.S.C. §102(b)	23
35 U.S.C. §112	13
35 U.S.C. §311	1

Rules

37 C.F.R. §42.100(b)	10
37 C.F.R. §42.103	68
37 C.F.R. §42.104(a)	1
37 C.F.R. §42.104(b)(1)	12
37 C.F.R. §42.21	1
37 C.F.R. §42.8(b)(1)	67
37 C.F.R. §42.8(b)(2)	67
37 C.F.R. §42.8(b)(3)	67

37 C.F.R. §42.8(b)(4)	67
-----------------------------	----

TABLE OF ABBREVIATIONS

Abbreviation	Full Name
'936 Patent	U.S. Patent No. 7,593,936
Challenged Claims	Claims 1-22 of the '936 Patent
Patent Owner	Patent Owner Netskope, Inc.
Petitioner	Petitioner Fortinet, Inc.
POSITA	Person of Ordinary Skill In The Art

EXHIBIT LIST

Exhibit No.	Document
1001	U.S. Patent No. 7,593,936 ("the '936 patent")
1002	File History of the '936 patent
1003	Declaration of Dr. Kevin Almeroth
1004	U.S. Patent No. 7,225,343 to Honig et al. ("Honig")
1005	U.S. Patent Pub. No. 2005/0005171 to Oliphant et al. ("Oliphant")
1006	U.S. Patent No. 6,327,550 to Vinberg et al. ("Vinberg")
1007	<i>Netskope, Inc. v. Fortinet, Inc.</i> , No. 3:25-cv-2360 (N.D. Cal.), Dkt. 1 ("Complaint").
1008	<i>Netskope, Inc. v. Fortinet, Inc.</i> , No. 3:25-cv-2360 (N.D. Cal.), Dkt. 1, Ex. 15.
1009	Exhibit 15 to Netskope's Infringement Contentions, dated July 1, 2025
1010	U.S. Patent App. No. 60/484,085 ("Oliphant Provisional")

I. INTRODUCTION

Petitioner Fortinet, Inc. ("Petitioner") respectfully petitions, under 35 U.S.C. §311 and 37 C.F.R. §42.21, for *inter partes* review ("IPR") of claims 1-22 ("the Challenged Claims") of U.S. Patent No. 7,593,936 ("the '936 patent") (EX1001) on the grounds below.¹

The '936 patent claims a method of using a "snapshot" of a computer to detect anomalies that may be present on the computer. But antivirus programs and firewalls had been detecting anomalies on a computer for a decade before the priority date of the patent. The '936 patent, therefore, adds automatically creating a set of rules for identifying those anomalies using patterns in the snapshots. The '936 patent does not, however, purport to create some novel method of rule creation. And automatic rule creation based upon patterns in snapshots was clearly established in the prior art. None of the prior art in this Petition was before the examiner during prosecution, and the Challenged Claims are invalid over that art.

Accordingly, Petitioner respectfully requests that the Board cancel the Challenged Claims.

II. GROUNDS FOR STANDING (37 C.F.R. §42.104(a))

Petitioner certifies that the '936 patent is available for IPR and that Petitioner is not barred or estopped from requesting IPR of the Challenged Claims on the

¹ Unless otherwise noted, all emphases and annotations have been added.

grounds in this petition. The '936 patent issued more than 9 months ago, and Petitioner was served with the complaint alleging infringement of the '936 patent less than one year ago. EX1007.

III. THE PATENT AND STATE OF THE ART

The '936 patent was filed on August 11, 2004, and claims priority to a provisional application filed August 11, 2003. EX1001. For the purposes of this petition, Petitioner assumes a priority date of August 11, 2003. Because it was filed before March 2013, the '936 patent is subject to pre-AIA patent laws and cites to Title 35 of the U.S.C. will be to that version unless otherwise specified.

A. Computer Malware and Other Attacks

Computer malware is one type of cyberattack that refers to malicious software that is designed to harm, disrupt, or gain unauthorized access to a computer. Computer malware has been well-known in the field since at least the 1980s. For example, a computer virus is a type of malware that will often attach itself to existing files or programs and seek to replicate itself constantly, often causing disruption to computer operations and loss of data. EX1003, ¶48. Other well-known types of computer malware include worms, Trojan horses, keyloggers, adware, and ransomware. *Id.*

Computers may be infected by malware through data received over the internet. For example, if a user unknowingly visits a malicious website, such as by

clicking on a link in an email, the website may deliver data such that, when the user's web browser tries to render the content, a virus will get installed on the user's computer. *Id.*, ¶49. Viruses often take advantage of vulnerabilities in the user's computer. Vulnerabilities refer to weaknesses in the computer system that can be exploited by computer malware to, for example, install unwanted software. Often vulnerabilities are caused by mistakes or "bugs" in the underlying software, such as failing to check for valid inputs. *Id.*, ¶49.

Another example of a cyber-attack is a denial-of-service attack, where the attacker will try to disrupt the normal functioning of a computer, such as by flooding it with network traffic such that it cannot receive normal communications. *Id.*, ¶50. Denial-of-service attacks may come from a single source or may be distributed across multiple sources. *Id.* Denial-of-service attacks were well-known since at least the 1990s (e.g., Panix in 1996). *Id.*

Other types of cyberattacks include phishing, man-in-the-middle, and SQL injections. *Id.*, ¶51.

B. Computer Security

Computer security really took off as an industry in the 1990s with the popularization of the internet and well-publicized security issues (e.g., Y2K bug). EX1003, ¶¶52-53. For example, Norton Antivirus was first released in 1991 as a software program that was designed to identify and remove malware on a

computer by searching for known signatures. Signatures are sequences uniquely associated with known malware. *Id.*, ¶54.

As another example, computer firewalls became popular in the 1990s. Firewalls would intercept traffic between a user and the internet and block any packets that were malicious, such as by searching for a signature in the data packet. Firewalls could therefore prevent malicious data from even reaching a user's computer. *Id.*, ¶55.

Similarly, computer software companies, including Microsoft, started providing regular security patches in the 1990s, or software updates to eliminate (i.e., patch) known vulnerabilities in the software. *Id.*, ¶56. These security patches would prevent malware from taking advantage of these known vulnerabilities to gain access to the system and, for example, install a virus. *Id.*

C. Network and System Administration

By the time of the alleged inventions, large companies had hundreds or thousands of computers on their network that required management and oversight by administrators. EX1003, ¶57. These network administrators were responsible for ensuring the proper functioning and security of the computers on the network through the use of, for example, software upgrades, firewalls, and antivirus software. *Id.* These administrators were also responsible for applying security patches to the large number of computers on the network to ensure they were

protected from known security vulnerabilities. Installation of these patches often required certain actions to be taken by the users, such as by downloading and installing software and, sometimes, restarting the computer. *Id.*, ¶58.

Administrators also had means of tracking which computers on the network had taken the required actions to ensure compliance by all users. *Id.* For example, the following commercially available software was introduced for system and network administration by 2003: Microsoft's Systems Management Server (1994); Software Update Services (2002); Novell System's ZENworks (1990s); IBM's Tivoli Management Framework and Tivoli Software Distribution (1996); HP's OpenView (1995). *Id.*

D. The '936 Patent

The '936 patent generally relates to the idea of "automated computer support." EX1001, 1:15. According to the patent, most levels of computer support in the prior art require manual effort from the user and/or from trained professionals. The '936 patent purports to create a "a new support model in which machines will serve as the primary agents for making decisions and initiating actions." *Id.*, 3:36-38. Figure 1 depicts the alleged invention:

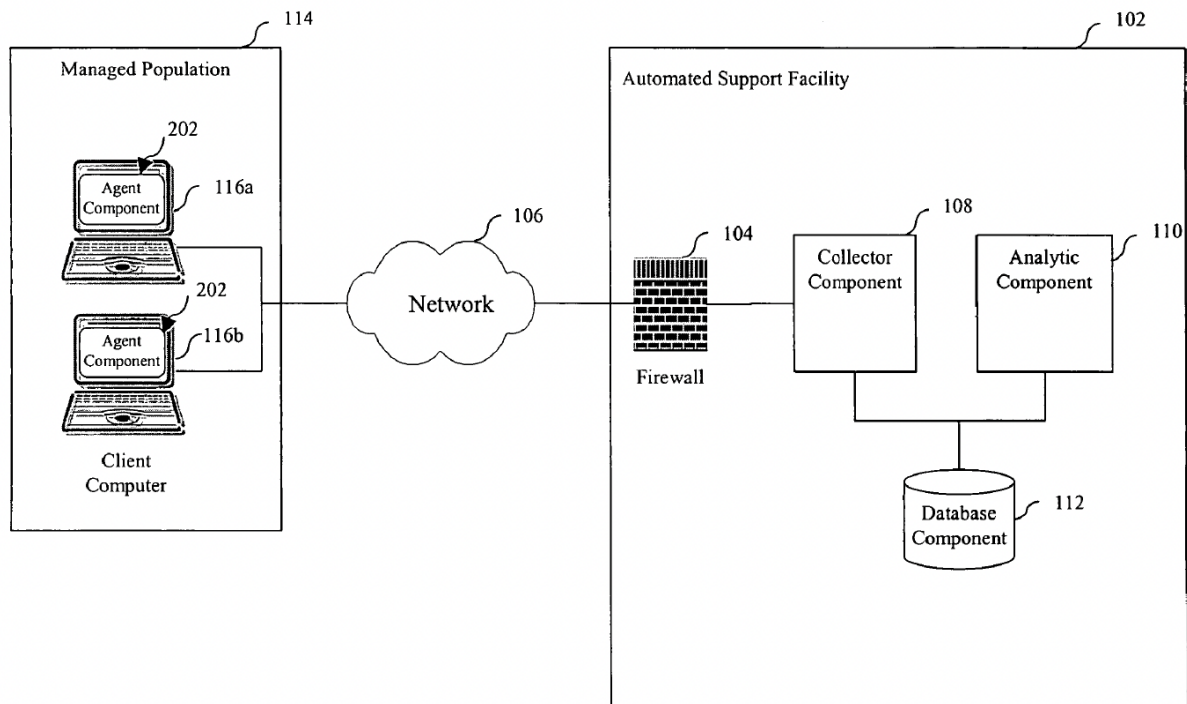


FIG. 1

The '936 patent discloses putting "Agent components **202**" on multiple monitored computers in a network. The Agent will take "a detailed snapshot of the state of the machine in which it resides. This snapshot includes a detailed examination of all system files, designated application files, the registry, performance counters, processes, services, communication ports, hardware configuration, and log files." *Id.*, 5:42-46. According to the patent, the Agent will compress the snapshot and send it to the Collector component **108**, which downloads, decompresses, and parses incoming snapshots for storage in a database **112**. *Id.*, 5:1-48. The Analytic component **110** can extract and analyze that data

from the database using an "adaptive reference model." *Id.*, 5:6-9. This adaptive reference model is supposed to be indicative of the "normal state" of the system and "can be used to distinguish between normal and abnormal system states." *Id.*, 7:17-18. According to the patent, the adaptive reference model "analyzes the snapshots from many computers and identifies statistically significant patterns using a generic data mining algorithm or a proprietary data mining algorithm designed specifically for this purpose." *Id.*, 7:24-28.

Upon receiving new snapshots from Agents, the adaptive reference model component can compare the snapshot against the adaptive reference model to determine if there is an anomaly present in the snapshot. *Id.*, 9:10-23.

E. The Challenged Claims

The '936 patent has 22 claims, of which 2 are independent. Claim 1 is reproduced below:

1. A method of detecting abnormal system states in computers, comprising:
 - receiving snapshots from a plurality of computers within a population of computers, wherein individual snapshots include data indicating a state of a respective computer;
 - automatically generating an adaptive reference model comprising a rule set customized to characteristics of the population of computers, the rule set being developed by identifying patterns among the snapshots from the plurality of computers such that the adaptive reference model is indicative of normal states in the computers within the population; and
 - comparing a snapshot from at least one of the computers to the adaptive reference model to determine whether an anomaly is

present in the state of the at least one of the computers.

Claim 12, the other independent claim, is the corresponding computer-readable medium claim for Claim 1. The dependent claims recite further action that may be taken by the system in response to detecting an anomaly (claims 2-6), the contents of the snapshot (claim 7), or the format or creation of the adaptive reference model (claims 8-11).

F. File History

The initially filed application that became the '936 patent had 50 claims. On March 23, 2005, the examiner issued a non-final rejection on all 50 claims, finding that the claims were obvious in light of Chen (US2003/0182312) over Bamford (US6,243,702). EX1002, 96-106. In response, Applicant argued that Chen over Bamford did not teach "automatically creating an adaptive reference model based at least in part on [a] plurality of snapshots." *Id.*, 123. Applicant claimed that the examiner agreed with that during an interview. *Id.* On December 7, 2005, however, the examiner issued final rejection, indicating that he never agreed with Applicant during the interview. The examiner thus rejected the claims again over the same art. *Id.*, 133-137.

To overcome the Chen/Bamford combination, Applicant amended the claims to recite that the snapshots were "detailed state snapshots," where "each of the plurality of detailed state snapshots having data associated with at least one of a

registry entry, a performance counter, a process, a communication port, a hardware configuration, or a log." *Id.*, 156 (underlined language added in amendment).

On April 24, 2006, the examiner entered a notice of allowance by entering an examiner's amendment that added that the detailed state snapshots had to come "from a plurality of computers." *Id.*, 186-190. But Applicant requested continued examination, cancelling claims 1-50 and replacing them with claims 51-78. The new claims included a plurality of computers limitation, but removed the "detailed snapshot" requirement, instead stating that "individual snapshots include data indicating a state of a respective computer." *Id.*, 211-212.

On May 1, 2007, the examiner issued a non-final rejection, again finding the claims obvious in light of Chen over Bamford. *Id.*, 243-248. In response, on August 1, 2007, Applicant amended the claims to add the limitation "in the event an anomaly is present, supplying information about the anomaly to a system configured to diagnose and correct a trouble condition in the at least one of the computers." *Id.*, 260. But Applicant removed that language 6 days later and instead traversed the rejection. *Id.*, 273. The examiner entered a final rejection and Applicant appealed. *Id.*, 285, 311.

On July 16, 2009, the Board reversed the examiner. In doing so, the Board found that:

- "normal states in the computers within the population" meant "a *desired state* of the computers within the population that is *absent*

undesirable changes or anomalies"; and

- "anomaly" meant "an unexpectedly present asset, an unexpectedly absent asset, or an asset that has an unknown value."

Id., 361.

According to the Board, Chen disclosed only a "basic set of rules" that were "not adaptable rule sets generated in the manner required by" the claim language. The Board also found that Chen does not identify patterns among snapshots but instead performs mirroring or replication on multiple computers. *Id.*, 363. The claims were therefore allowed.

G. POSITA

The relevant art for the '936 patent is the field of computer science generally, and specifically, computer security. A person of ordinary skill in that art ("POSITA"), as of August 2003, would have been an individual with either (1) at least a bachelor's degree in computer science or computer engineering or an equivalent field plus at least one year of experience working on computer security, or (2) at least 3 years of experience working on computer security, even without a formal degree. EX1003, ¶¶66-67.

IV. CLAIM CONSTRUCTION

A claim is construed "using the same claim construction standard that would be used to construe the claim in a civil action," 37 C.F.R. § 42.100(b), which is governed by *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir.

2005). Under the standard set out in *Philips*, this petition applies the plain and ordinary meaning to each term in the '936 Patent, which is "the meaning that the term would have to a [POSITA] in question at the time of the invention." *Id.*, 1313.

All Challenged Claims refer to "a snapshot" or "snapshots." The specification states that:

This snapshot includes a detailed examination of all system files, designated application files, the registry, performance counters, processes, services, communication ports, hardware configuration, and log files.

EX1001, 5:43-46. Petitioner submits that this description is consistent with the plain and ordinary meaning of "snapshot" at the time of the alleged invention.

This construction of a snapshot is further supported by the fact that the patent recognizes that the collection of these snapshots "involves collecting a massive amount of data and can take anywhere from a few minutes to hours to execute, depending on the configuration of the client." *Id.*, 8:67-9:2.

Patent Owner, however, appears to have taken a much broader interpretation of "snapshot" in its Complaint and Infringement Contentions. There, they appear to accuse the collection of "OS metadata" or "OS stack data" as the alleged snapshot. EX1008, 3-5; EX1009, 4-7. In other words, Petitioner seems to imply that collecting one type of information (e.g., OS metadata or OS stack data) from a computer can qualify as a snapshot, as opposed to all the data described as part of the snapshot in the patent. Petitioner disagrees with Patent Owner's apparent

construction of "snapshot" to capture only a single type of data.

The Board need not resolve this dispute for the purposes of this Petition, however, because the prior art invalidates the claims under either construction as detailed below.

Because the Board construed two limitations during prosecution, Petitioner will address those constructions below:

- "normal states in the computers within the population" meant "a *desired state* of the computers within the population that is *absent undesirable changes or anomalies*"; and
- "anomaly" meant "an unexpectedly present asset, an unexpectedly absent asset, or an asset that has an unknown value."

EX1002, 361.

Petitioner respectfully submits that no other constructions are required for the purposes of this petition.

V. SUMMARY OF CHALLENGE AND RELIEF REQUESTED

A. 37 C.F.R. §42.104(b)(1): The Challenged Claims

Petitioner requests IPR and cancellation of the Challenged Claims (claims 1-22) in view of the references discussed below.

B. 37 C.F.R. §42.104(b)(1): Prior Art Overview and Specific

Grounds of Rejection

1. Honig

Honig is a U.S. Patent that was filed January 27, 2003, and claims priority to a provisional application filed on January 25, 2002. EX1004. Honig is therefore prior art under at least §102(e).

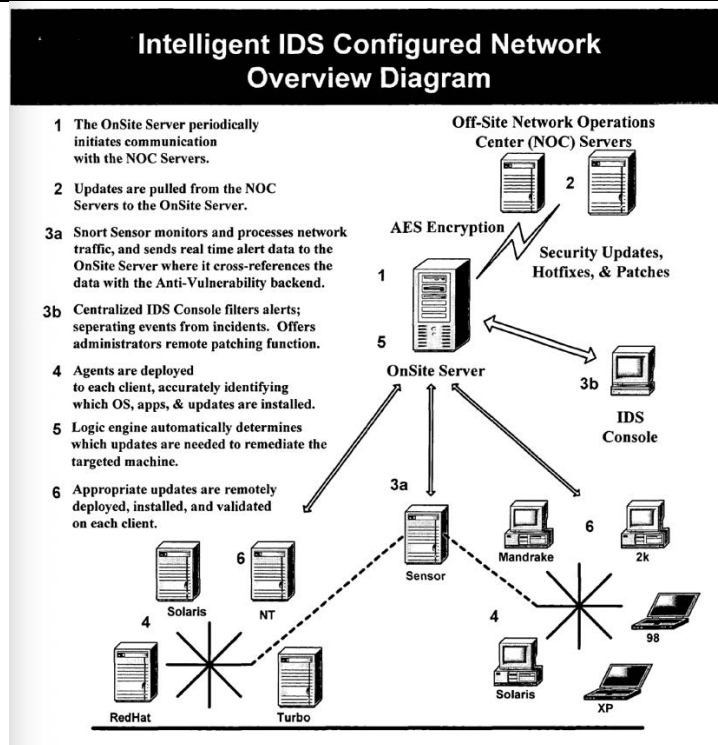
2. Oliphant

Oliphant is a published U.S. Patent that was filed July 1, 2004, and claims priority to a provisional application filed on July 1, 2003. EX1005. For Oliphant to be entitled to its provisional date, "the specification of the provisional must contain a written description of the invention and the manner and process of making and using it, in such full, clear, concise, and exact terms, 35 U.S.C. § 112 ¶ 1, to enable an ordinarily skilled artisan to practice the invention *claimed* in the *non-provisional* application." *Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015) (quoting *New Railhead Mfg., L.L.C. v. Vermeer Mfg. Co.*, 298 F.3d 1290, 1294 (Fed. Cir. 2002)) (quotation marks omitted). In sum, Oliphant is entitled to the priority date of the Oliphant Provisional (July 1, 2003) so long as the provisional application provides support to the claims in the non-provisional application. It does. As set forth in the chart below, the Oliphant Provisional provides Section 112 support for each limitation of claim 1 of the non-provisional application.

Claim 1 of Oliphant (EX1005)	Provisional Support (EX1010)
A system, comprising:	The Oliphant Provisional discloses a "SysUpdate Configured Network" system that includes an OnSite Server, connected to a number of different client devices. SysUpdate™ Configured Network Overview Diagram The diagram illustrates the SysUpdate™ Configured Network architecture. At the top, 'Network Operations Center (NOC) Servers' (labeled 2) are connected to an 'OnSite Server' (labeled 5) via a lightning bolt labeled 'Security Updates, Hotfixes, & Patches'. The OnSite Server is also connected to a 'Management Console(s)' (labeled 3). Below the OnSite Server, two groups of client devices are shown: 'Group One' and 'Group Two'. Arrows labeled 'AES Encryption' point from the OnSite Server to these groups. Group One includes Solaris, NT, Turbo, RedHat, and 2k. Group Two includes Solaris, 2k, XP, and 98. A central star-like symbol connects the OnSite Server to the client groups. A list of six numbered steps is provided on the left side of the diagram. 1 The OnSite Server periodically initiates communication with the off-site NOC Servers. 2 Updates are pulled from the NOC servers to the OnSite Server. 3 Centralized MMC Snap-In Console offers flexible security policy options & comprehensive reporting capabilities. 4 Agents accurately identifies OS, apps, & updates installed on each client by file versions and hashes. 5 Logic engine automatically determines which updates are needed on each client, taking into account OS, app, and update dependencies. 6 Appropriate updates are automatically deployed, installed, and validated on each client. EX1010, 36. The Oliphant Provisional further discloses an "Intelligent IDS Configured Network," or system, that includes an OnSite Server, connected to a number of different client devices.

**Claim 1 of Oliphant
 (EX1005)**

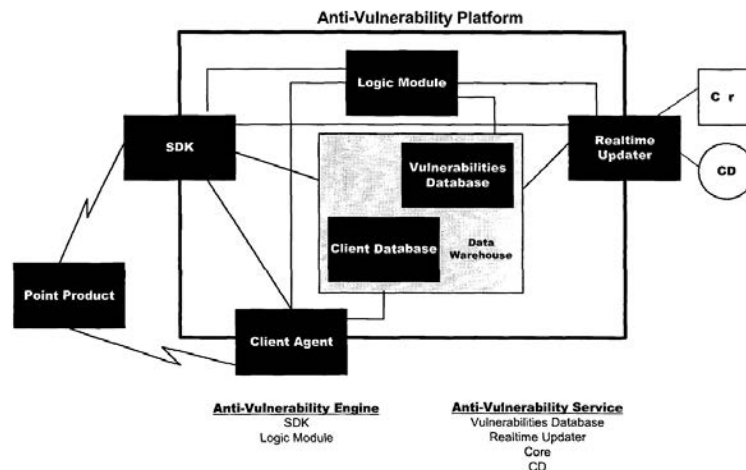
Provisional Support (EX1010)

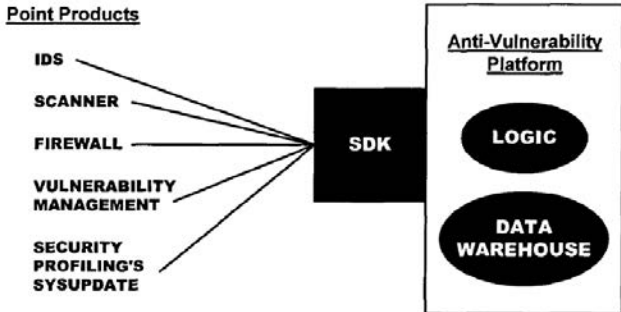


Id., 34; EX1003, ¶129.

See also EX1010, 11 (explaining the "system requirements" for anti-vulnerability technology);

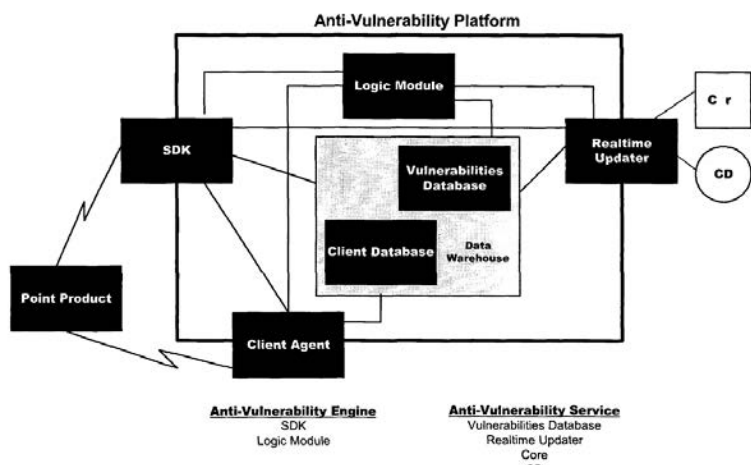
Figure 2 – SDK and Platform Component Diagram



Claim 1 of Oliphant (EX1005)	Provisional Support (EX1010)
	<i>Id.</i>, 16; Figure I – Anti-Vulnerability Technology  For more detail, see the Platform Component Diagram on page 8. EX1010, 12.
a plurality of computing devices, each comprising at least one processor and memory, wherein the memory is encoded with programming instructions executable by the processor;	The Oliphant Provisional further discloses that there are number of different "clients" connected to the OnSite Server and that "Agents are deployed to each client." EX1010, 34. The examples of clients, include laptops, desktops, and servers, all of which include at least one processor and memory. <i>Id.</i>; EX1003, ¶130. The Agents include instructions to perform the operations in the following limitations.

Claim 1 of Oliphant (EX1005)	Provisional Support (EX1010)
	SysUpdate™ Configured Network Overview Diagram The diagram illustrates the SysUpdate™ network architecture. At the top, 'Network Operations Center (NOC) Servers' (labeled 2) are connected to an 'OnSite Server' (labeled 1) via a lightning bolt labeled 'Security Updates, Hotfixes, & Patches'. The 'OnSite Server' is also connected to a 'Management Console(s)' (labeled 3). Below the 'OnSite Server', two groups of clients are shown: 'Group One' and 'Group Two'. 'Group One' includes clients labeled 'Solaris', 'NT', 'Turbo', and 'RedHat'. 'Group Two' includes clients labeled 'Solaris', '2k', '98', and 'XP'. Arrows labeled 'AES Encryption' point from the 'OnSite Server' to both 'Group One' and 'Group Two'. A list of six numbered steps is provided on the left side of the diagram. 1 The OnSite Server periodically initiates communication with the off-site NOC Servers. 2 Updates are pulled from the NOC servers to the OnSite Server. 3 Centralized MMC Snap-In Console offers flexible security policy options & comprehensive reporting capabilities. 4 Agents accurately identify OS, apps, & updates installed on each client by file versions and hashes. 5 Logic engine automatically determines which updates are needed on each client, taking into account OS, app, and update dependencies. 6 Appropriate updates are automatically deployed, installed, and validated on each client. EX1010, 36.

Claim 1 of Oliphant (EX1005)	Provisional Support (EX1010)
	Intelligent IDS Configured Network Overview Diagram The diagram illustrates the Intelligent IDS Configured Network architecture. It shows the following components and their interactions: Off-Site Network Operations Center (NOC) Servers: Represented by two server icons at the top right. OnSite Server: A central server icon in the middle. Sensor: A server icon connected to the OnSite Server and various client machines. Client Machines: Various operating systems including Solaris, NT, RedHat, Turbo, Mandrake, 2k, 98, and XP. IDS Console: A computer icon connected to the OnSite Server. The numbered steps describe the process: 1 The OnSite Server periodically initiates communication with the NOC Servers. 2 Updates are pulled from the NOC Servers to the OnSite Server. 3a Snort Sensor monitors and processes network traffic, and sends real time alert data to the OnSite Server where it cross-references the data with the Anti-Vulnerability backend. 3b Centralized IDS Console filters alerts; separating events from incidents. Offers administrators remote patching function. 4 Agents are deployed to each client, accurately identifying which OS, apps, & updates are installed. 5 Logic engine automatically determines which updates are needed to remediate the targeted machine. 6 Appropriate updates are remotely deployed, installed, and validated on each client. Additional labels in the diagram include "AES Encryption" and "Security Updates, Hotfixes, & Patches". EX1010, 34; EX1003, ¶130.
a database of device status information that characterizes the zero or more vulnerabilities to which each of the computing devices is subject, where the device status information is kept current in substantially real time; and	The Oliphant Provisional discloses a "vulnerabilities database" and "client database" as part of the data warehouse that is connected to the client agents. These databases are constant updated by the agent to keep the databse current in substantially real time as detailed in the subsequent limitations. EX1003, ¶131.

Claim 1 of Oliphant (EX1005)	Provisional Support (EX1010)
	Figure 2 – SDK and Platform Component Diagram  EX1010, 16.
an application that	The Oliphant Provisional further discloses that there are number of different "clients" connected to the OnSite Server and that "Agents are deployed to each client." The Agents are the claimed application.

Claim 1 of Oliphant (EX1005)	Provisional Support (EX1010)
	SysUpdate™ Configured Network Overview Diagram The diagram illustrates a network architecture for SysUpdate. At the top, 'Network Operations Center (NOC) Servers' (labeled 2) are connected to an 'OnSite Server' (labeled 1) via a lightning bolt labeled 'Security Updates, Hotfixes, & Patches'. The 'OnSite Server' is connected to a 'Management Console(s)' (labeled 3) via a double-headed arrow. Below the 'OnSite Server', two groups of clients are shown: 'Group One' and 'Group Two'. Arrows labeled 'AES Encryption' point from the 'OnSite Server' to these groups. 'Group One' contains clients labeled 'Solaris', 'NT', 'Turbo', and 'RedHat'. 'Group Two' contains clients labeled 'Solaris', '2k', '98', and 'XP'. A central star-like symbol is present in each group. A list of six numbered steps is on the left, and a horizontal line is at the bottom of the diagram area. 1 The OnSite Server periodically initiates communication with the off-site NOC Servers. 2 Updates are pulled from the NOC servers to the OnSite Server. 3 Centralized MMC Snap-In Console offers flexible security policy options & comprehensive reporting capabilities. 4 Agents accurately identify OS, apps, & updates installed on each client by file versions and hashes. 5 Logic engine automatically determines which updates are needed on each client, taking into account OS, app, and update dependencies. 6 Appropriate updates are automatically deployed, installed, and validated on each client. EX1010, 36.

Claim 1 of Oliphant (EX1005)	Provisional Support (EX1010)
	Intelligent IDS Configured Network Overview Diagram 1 The OnSite Server periodically initiates communication with the NOC Servers. 2 Updates are pulled from the NOC Servers to the OnSite Server. 3a Snort Sensor monitors and processes network traffic, and sends real time alert data to the OnSite Server where it cross-references the data with the Anti-Vulnerability backend. 3b Centralized IDS Console filters alerts; separating events from incidents. Offers administrators remote patching function. 4 Agents are deployed to each client, accurately identifying which OS, apps, & updates are installed. 5 Logic engine automatically determines which updates are needed to remediate the targeted machine. 6 Appropriate updates are remotely deployed, installed, and validated on each client. EX1010, 34; EX1003, ¶132.
transmits a query signal to the database;	The Oliphant Provisional discloses that, in the "SysUpdate" system, the Agents "accurately identify[] OS, apps, & updates installed on each client by file versions and hashes." Then Agent reports that information to the OnSite Server, which includes a "logic engine." The "system's logic engine automatically reports back each client's out-of-compliance parameters." EX1010, 36, 39. A POSITA would have understood that the logic engine must receive that information from the Agents to perform the required analysis—i.e., the Agents transmit a query signal to the database. EX1003, ¶133.

Claim 1 of Oliphant (EX1005)	Provisional Support (EX1010)
	SysUpdate™ Configured Network Overview Diagram 1 The OnSite Server periodically initiates communication with the off-site NOC Servers. 2 Updates are pulled from the NOC servers to the OnSite Server. 3 Centralized MMC Snap-In Console offers flexible security policy options & comprehensive reporting capabilities. 4 Agents accurately identifies OS, apps, & updates installed on each client by file versions and hashes. 5 Logic engine automatically determines which updates are needed on each client, taking into account OS, app, and update dependencies. 6 Appropriate updates are automatically deployed, installed, and validated on each client. EX1010, 36; EX1003, ¶133.
receives a result signal, responsive to the query signal, from the database;	The Oliphant Provisional discloses that the "[l]ogic engine automatically determines which updates are needed on each client, taking into account OS, app, and update dependencies." EX1010, 36-39 ("Based on each client's individual policy parameter settings and configuration profile, the system's logic engine automatically reports back each client's out-of-compliance parameters."). The logic engine thus sends (and the Agent receives) a results signal responsive to the query signal from the database. EX1003, ¶134.

Claim 1 of Oliphant (EX1005)	Provisional Support (EX1010)
makes a security-related determination based on the result signal.	The Oliphant Provisional discloses that "[p]olicy changes may be automatically enforced to each computer or device individually, or by group, consistently in accordance with the policy selected by the network administrator." EX1010, 39. The Agent thus must automatically enforce the security policy changes (make a security-related determination) received from the logic engine. The Agent may also be required to install needed updates make a security-related determination) based upon the message from the logic engine. EX1003, ¶135.

Oliphant is therefore prior art under at least §102(e). See *Dynamic Drinkware*, 800 F.3d at 1378; EX1003, ¶136. Consistent with *In re Riggs*, 131 F.4th 1377, 1384 (Fed. Cir. 2025), the Oliphant Provisional "also provide[s] written description support for the specific portions of the patent specification identified and relied on in the prior art rejection" and cites thereto will be included in the analysis below.

3. Vinberg

Vinberg is a published U.S. Patent that was filed March 5, 2001, and issued December 4, 2001. EX1006. Vinberg is therefore prior art under at least §102(b).

C. Grounds of Rejection

Petitioner asserts the following specific grounds:

Ground	Claims	Basis	Prior Art
1	1, 7-12, 17-19, 21-22	§102	Honig

Ground	Claims	Basis	Prior Art
2	1, 7-12, 17-19, 21-22	§103	Honig
3	2-6, 13-16, 20	§103	Honig in view of Oliphant
4	1, 7-12, 17-19, 21-22	§103	Vinberg
5	2-6, 13-16, 20	§103	Vinberg in view of Oliphant

VI. GROUNDS 1 AND 2 – CLAIMS 1, 7-12, 17-19, 21-22 ARE INVALID AS ANTICIPATED OR RENDERED OBVIOUS BY HONIG

A. Overview of Honig

Honig generally discloses an intrusion detection system that utilizes data from a number of different computers. Figure 1 provides an overview of this system.

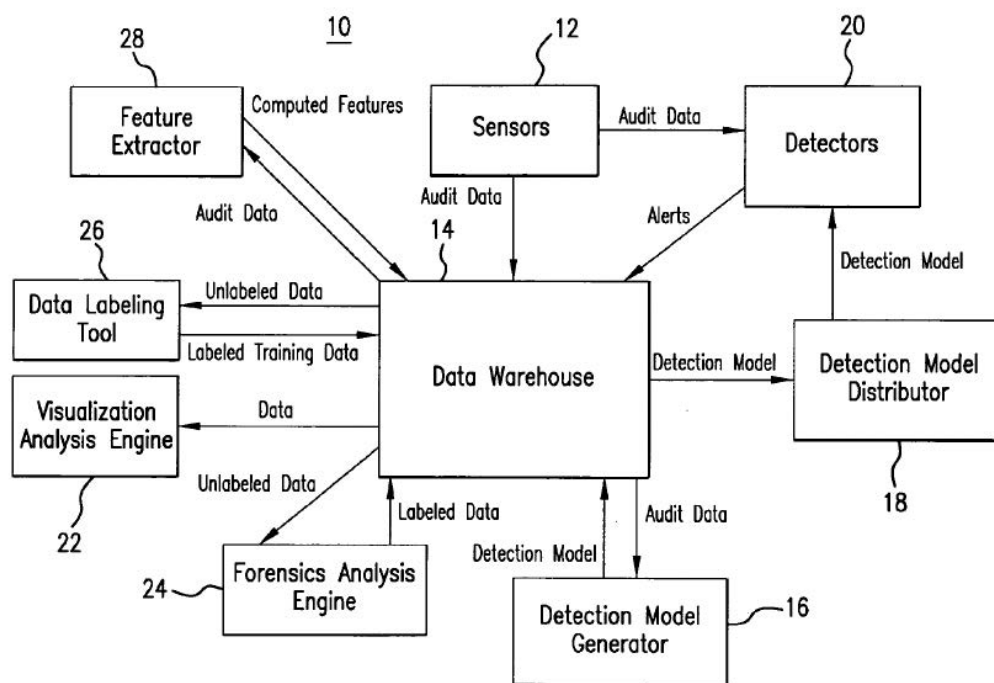


FIG. 1

Honig explains that the sensors 12 gather data from a host machine and then send that sensor data to the data warehouse 14. EX1004, 8:14-15. "Sensors 12

may be lightweight processes that gather information from a real-time audit stream.... The sensors **12** can gather information from any source." *Id.*, 9:17-22.

This includes sources of data on the host, such as event logs ("Windows event log"), the registry ("[r]egistry"), performance counters ("CPU and memory usage on a machine"), processes ("information about each running process"), services ("system calls"), and network usage ("network connections being made to and from the host"). *Id.*, 9:49-10:11. Honig makes clear that there can be "any number of host machines and sensors." *Id.*, 26:13-14.

Using the data accumulated from all of the different hosts in the data warehouse, the detection model generator **16** generates "detection models." *Id.*, 13:56-60. The system in Honig "is designed to work with any model generation algorithm." *Id.*, 13:61-62. Honig describes "three exemplary types of model generation algorithms," including "misuse detection," "supervised (traditional) anomaly detection," and "unsupervised anomaly detection." *Id.*, 20:33-38. Once the model is generated, it is "distributed to the detectors **20** by the model distributor **18**." *Id.*, 13:59-60.

The sensors **12** also send the gathered data to the detectors **20**. "Detectors **20** analyze audit stream data collected from a sensor and detect intrusions by using a detection model," which it received from the model distributor. *Id.*, 11:8-10.

And, if the analyzed "data corresponds to an attack, the detector **20** generates an alert." *Id.*, 8:56-57.

B. Claim 1

1. [1PRE]: "A method of detecting abnormal system states in computers, comprising:"

Honig discloses a system and method "for detecting intrusions in the operation of a computer system." EX1004, Abstract. Honig further discloses that it creates "anomaly detection models." *Id.*, 16:25-41. As discussed, a normal state is "a desired state of the computers within the population that is absent undesirable changes or anomalies," so an abnormal state would be a state with undesirable changes or anomalies. A POSITA would have understood that intrusions (which create undesirable changes) and anomalies in a computer system mean the computer is in an abnormal (anomalous) state. EX1003, ¶78.

Further, it would have been obvious to a POSITA to consider a computer with an intrusion or anomaly to be in an abnormal state. EX1003, ¶79.

2. [1A]: "receiving snapshots from a plurality of computers within a population of computers, wherein individual snapshots include data indicating a state of a respective computer;"

Honig discloses that the sensors **12** gather data from a host machine and then send that data to the data warehouse **14**. EX1004, 8:14-15. The data warehouse thus receives sensor data from a plurality of different computers. Honig makes clear that there can be "any number of host machines." *Id.*, 26:13-14.

The sensor data in Honig comprises a snapshot under both constructions set forth above, or any other reasonable construction, of a snapshot, where the snapshot includes data indicating a state of a computer. The '936 patent contemplates that a "snapshot includes a detailed examination of all system files, designated application files, the registry, performance counters, processes, services, communication ports, hardware configuration, and log files." EX1001, 5:43-46; EX1003, ¶¶80-82.

Honig explains that the "[s]ensors **12** may be lightweight processes that gather information from a real-time audit stream.... The sensors **12** *can gather information from any source.*" EX1004., 9:17-22. A POSITA would have understood that information gathered by the sensors from the machine constitutes a snapshot as contemplated by the '936 patent. Indeed, Honig makes clear that there are "multiple sensors **12** for each host." *Id.*, 9:49-50; EX1003, ¶83. For example, Honig discloses collecting the following information by sensors:

'936 patent	Honig
"all system files"	"registry sensors" (EX1004, 9:56). A POSITA would have understood that the system files include the crucial files necessary for the OS to function properly and therefore would have been motivated to track those files a sensor as disclosed in Honig. Indeed, Honig already discloses tracking the registry, which is generally considered to be part of the system files. EX1003,

'936 patent	Honig
	¶83. It would have been obvious to expand that sensor to track all of the other system files as well. <i>Id.</i>
"designated application files"	"Windows™ event log sensors which take information from the application , security, and event logs." (EX1004, 9:56-58). A POSITA would have understood that application logs would necessarily include certain application files. EX1003, ¶83.
"the registry"	"registry sensors" (EX1004, 9:56)
"performance counters"	"Resource sensors gather information about CPU and memory usage on a machine." (EX1004, 10:3-4)
"processes"	"Process sensors use the /proc virtual file-system to gather information about each running process." (EX1004, 9:66-10:1)
"services"	"Software wrappers are sensors that gather information about system calls." (EX1004, 9:59-60)
"communication ports"	"Netatat sensors use the netatat tool that gathers information about network connections on the host." (EX1004, 9:60-62)
"hardware configuration"	"Resource sensors gather information about CPU and memory usage on a machine." (EX1004, 10:3-4). A POSITA would have understood that, by tracking CPU and memory usage, the resource sensors would have necessarily tracked the hardware configuration of the host, including at least the CPU and memory configurations. EX1003, ¶83.

'936 patent	Honig
"log files"	"Windows event log sensors" (EX1004, 9:55)

A POSITA would have understood that all of this data disclosed in Honig as part of the snapshot is indicative of the state of the computer from which it was collected. That is why Honig discloses collecting that data (e.g., event log sensors, registry, performance counters). EX1003, ¶83. Furthermore, Honig discloses that the data from all of the sensors on a host can be sent "through a single source" so that data from all of the sensors can be sent together to the data warehouse. EX1004, 25:31-36.

As such, under any reasonable construction of "snapshot," Honig discloses and receiving a snapshot at the data warehouse from the plurality of sensors on the host device because it explicitly discloses or at least renders obvious all nine enumerated data types disclosed by the '936 patent as part of the snapshot.

Honig further renders obvious this limitation. As one (non-limiting) example, and as discussed in further detail below, given the open-ended nature of Honig's disclosures with respect to what information might be disclosed in the "snapshot," a POSITA would understand that a wide variety of information could be included in such a report. EX1003, ¶¶85-88.

3. **[1B]: "automatically generating an adaptive reference model comprising a rule set customized to characteristics of the population of computers, the rule set being developed by identifying patterns among the snapshots from the plurality of computers such that the adaptive reference model is indicative of normal states in the computers within the population; and"**

Honig discloses a detection model generator **16** that generates "detection models" (i.e., the claimed "*adaptive reference model*") based upon all of the data generated from the sensors across different devices (the population of computers). EX1004, 13:56-60. For example, Honig explains that "[t]he distributed architecture of system **10** allows the system to gather data from heterogeneous systems.... The ability to accommodate these different sources of information in a consistent way is a large advantage of the adaptive model generation system." *Id.*, 19:32-41, 12:38-43 ("A sensor **12** is installed on each of the hosts. This sensor collects the data and stores it in the data warehouse **14**. If each host is typically used in the same way, *a large data set may be created containing the combined event logs from each of the hosts.*"); EX1003, ¶89.

Honig further discloses that the detection models are automatically generated. For example, Honig discloses that its system uses "Adaptive Model Generation" that "provides and automates many of the critical processes in the deployment and operation of real time data mining-based intrusion detection systems," including "detection model generation." EX1004, 6:50-59. And Honig

discloses both "supervised" and "unsupervised anomaly detection which trains on unlabeled data." *Id.*, 20:36-38. Unsupervised anomaly detection, as disclosed in Honig, means that these updated models are created automatically, or unsupervised. EX1003, ¶¶90-91. Further, a POSITA would have understood that, to provide real-time updates, the process must be done automatically, or by the system without manual intervention. *Id.* It would further have been obvious to a POSITA that the system in Honig could be implemented such that real-time updates are generated automatically. *Id.*

The system in Honig "is designed to work with any model generation algorithm." EX1004, 13:61-62. Honig describes "three exemplary types of model generation algorithms," including "misuse detection," "supervised (traditional) anomaly detection," and "unsupervised anomaly detection." *Id.*, 20:33-38. Each of these model generation algorithms look for patterns in the data from the different distributed hosts across the system to create a "detection model" that is updated (adapted) in real-time as discussed above (i.e., the claimed "***adaptive reference model***"). *Id.*, 20:39-27:57. Further, a POSITA would have found it obvious that these model generation algorithms look for patterns in the data. EX1003, ¶92.

Honig explains that there are a number of different detection models that could be developed and updated by the model distributor, including "signature-

based," "decision tree," "probabilistic," and "support vector machine model." *Id.*, 11:8-55. Each of these detection models is a set of rules developed by the model generation algorithm (which Honig discloses could be unsupervised) that will distinguish between normal and abnormal states of the computer. EX1003, ¶¶93-94. By distinguishing between normal and abnormal states of a computer, the models are indicative of a desired state of the computers within the population that is absent an undesirable change or other anomaly. *Id.* It further would have been obvious to a POSITA that, because the models identify anomalies, they are necessarily indicative of a normal state of the computers in the population or else the model would not be able to identify any anomalies. *Id.*

For example, a signature-based detection model will look for signatures of known attacks and, if any signatures match, the system generates an alert that the system is in an abnormal state. EX1004, 11:20-24. As another example, a decision-tree model will consist of a set of rules set forth in the tree's branches. EX1003, ¶¶93-94. A probabilistic model will also include a set of rules based upon the different probabilities determined. And a support vector model likewise has a set of rules based upon the distance between a data point and a hyperplane. *Id.*

It would further have been obvious to a POSITA to implement a signature-based, decision tree, probabilistic, and support vector model with a set of rules to

determine whether or not an anomaly exists. For example, the rules in these models could be implemented to check the model to see if there is an anomaly. Then, if there is an anomaly, the system will generate an alert, which is an example of a rule. *Id.*

4. **[1C]: "comparing a snapshot from at least one of the computers to the adaptive reference model to determine whether an anomaly is present in the state of the at least one of the computers."**

Honig discloses that the detection models are distributed to the detectors **20**. The detectors **20** use that model to compare it against real-time data from the sensors: "Detectors **20** analyze audit stream data collected from a sensor and detect intrusions by using a detection model." EX1004, 11:8-10. "Detectors **20** receive detection models from model distributors **18**, which distribute models stored in the data warehouse originally created by model generators. The detectors **20** receive real time updates from the model distributors **18**." *Id.*, 11:56-60.

The analysis by the detectors involves comparing the snapshot data received from the sensors to the detection model received from the model distributors. For example, if a signature-based model is used, the system will compare the signatures in the detection model to those in the sensor data. If a signature is detected (unexpectedly present value), then the system will identify an anomaly. EX1003, ¶¶95-96. As another example, if a decision tree model is used, the snapshot will be compared against each branch of the tree. The system will

identify an anomaly if there is a branch of the tree that indicates an unexpected value, or an unexpectedly missing value. *Id.* If a probabilistic model is used, the detection will involve a comparison against probabilities. The system will identify an anomaly if the result of the analysis indicates an unexpected or unknown value or asset. And, if a support vector model is used, there will be a comparison of distances and identify an anomaly if there is an unknown or unexpected distance value. *Id.* In sum, a POSITA would have understood that, for each of these models, the analysis identifies anomalies by checking for values that are unknown, unexpectedly present, or unexpectedly absent. *Id.*

Then, "[i]f the data corresponds to an attack, the detector **20** generates an alert." EX1004, 8:56-57; 11:23-24 ("If any of the signatures match, the detector generates an alert."); 11:47-50.

In sum, Honig discloses each and every limitation of claim 1 because it discloses a detection model generator that receives snapshots from agents from different computers around the network and automatically generates a detection model (adaptive reference model) that it will compare against future snapshots to determine anomalies. EX1003, ¶¶99-100. To the extent Honig does not explicitly disclose certain required aspects of the "snapshot," Honig at least renders obvious receiving a snapshot for the reasons set forth above. *Id.*

- C. Claim 7: "The method of claim 1, wherein individual snapshots include data associated with at least one of: system files, application files, a registry entry, a performance counter, a process, a communication port, a hardware configuration, a log file, a running task, services, and network connections."**

As in [1A], Honig discloses and/or renders obvious that the snapshot contains system files, application files, a registry entry, a performance counter, a process, communication ports, hardware configuration, log file, and services. EX1004, 9:49-10:11; EX1003, ¶¶101-103. Honig further discloses including information about a running task in the snapshot. EX1004, 9:67-10:1 ("gather information about each running process"). Honig also discloses including information about network connections in the snapshot. *Id.*, 9:60-62 ("Netatat sensors use the netatat tool that gathers information about network connections on the host."). In other words, Honig discloses or renders obvious each and every one of types of data in claim 7, any one of which would be sufficient to invalidate this claim. EX1003, ¶¶101-103.

- D. Claim 8: "The method of claim 1, further comprising: inserting rules into the rule set of the adaptive reference model to augment or override rules of the rule set automatically generated from the snapshots."**

As in [1B], Honig discloses that the "detectors **20** receive real time updates from the model distributors **18**. This process keeps the detection models updated as soon as new models are available." EX1004, 11:59-61. Honig therefore discloses automatically updating or augmenting the adaptive reference model by

providing new rules. A POSITA would have understood that these updates would be in the form of inserting new rules or modifying (overriding) existing rules. For example, if a signature-based model is used, a POSITA would have understood that new signatures (rules) would be added to the existing signatures (rules) or existing signatures would be modified or removed, augmenting those existing rules in the detection model. EX1003, ¶¶104-106. Similarly, a POSITA would have understood that for a decision tree model, new branches (rules) could simply be added to the tree to augment and override the existing rules. *Id.* For a probabilistic model, probabilities would be updated, overriding the existing rules. *Id.* And, for the support vector model, new rows (rules) would be added to update or override the existing rules. *Id.*

Furthermore, a POSITA would have found it obvious that updates to the detection models disclosed in Honig would be sent via rules that would add to or update the existing detection model. A POSITA would have understood that it would not be efficient to resend the entire detection model for every update and, therefore, would have known to send only updated rules. *Id.*, ¶107.

E. Claim 9: "The method of claim 1, wherein the adaptive reference model is generated to include a value layer that determines whether an asset value contained in a snapshot is anomalous."

Honig discloses that one type of detection model that may be used is a "probabilistic model." EX1004, 11:37. In the probabilistic model, the detector

computes a "probability associated with each record." *Id.*, 11:38-40. In other words, the detector computes an asset value (probability) for each record to determine if it is anomalous. EX1003, ¶108-109.

Further supporting that Honig discloses claim 9, the '936 patent discloses that the "value layer 406 tracks the values of asset/value pairs provided by the Agent component." EX1001, 10:37-38. Honig discloses precisely that. Honig discloses probabilities (values) associated with each record (asset). EX1004, 11:37; EX1003, ¶110.

It would further have been obvious to a POSITA that the detection model described in Honig would track the values of the records (assets) paired with different values to check for anomalies associated with those records. EX1003, ¶¶111-112.

F. Claim 10: "The method of claim 1, wherein the adaptive reference model is generated to include a cluster layer that tracks relationships between assets and identifies an anomaly in response to an asset being unexpectedly absent from or present in a set of assets in a snapshot."

Honig discloses that one type of detection model that may be used is a "decision tree" model. EX1004, 11:27. In the decision tree model, the detector "would take this model and evaluate the detection model on a record by following the relevant branches of the tree." *Id.*, 11:30-32; EX1003, ¶¶113-114.

Further supporting that Honig discloses claim 10, the '936 patent discloses that the "cluster layer **408** tracks relationships between asset names." EX1001, 11:58-59. A POSITA would have understood that the branches of the tree track relationships of different assets and that various relationships (branches) would lead to identifying whether there is an anomaly or not. EX1003, ¶115.

It would further have been obvious to a POSITA that the detection model described in Honig would track relationships between asset names to determine when anomalies exist between different assets. EX1003, ¶¶116-117.

G. Claim 11: "The method of claim 1, wherein the adaptive reference model is generated to include a profile layer that identifies anomalies in response to violation of relationships of clusters of assets in a snapshot."

As in claim 10, Honig discloses a decision tree model with multiple branches that set forth the relationship of different assets. EX1004, 11:27-32. A POSITA would have understood that the branches would identify anomalies based upon the branches and that nearby branches would represent similar records, or a cluster of assets. EX1003, ¶¶118-119.

Further supporting that Honig discloses claim 11, the '936 patent discloses that the "profile layer **410** in the embodiment shown detects anomalies based upon violations of cluster relationships." EX1001, 12:8-10. A POSITA would have understood that the decision tree model detects anomalies based upon violations of branches and nearby branches having anomalous values. EX1003, ¶120.

It further would have been obvious to a POSITA that the detection model described in Honig would detect anomalies based upon relationships between asset names having anomalous values. EX1003, ¶¶121-122.

H. Claim 12

Claim 12 is substantially similar to claim 1 except it recites a computer-readable medium instead of a method. It is invalid over Honig for the same reasons detailed in claim 1 because a POSITA would have known that the instructions for executing that method would be stored on a computer-readable medium, so it could be executed by a computer based upon the following mapping. EX1004, 7:26-51; EX1003, ¶123.

Claim	Reasons for Invalidity
[12PRE] A computer readable medium storing instructions, that when executed by a computer, cause the computer to perform functions of:	<i>See</i> [1PRE]. EX1003, ¶123.
[12A] receiving snapshots from a plurality of computers within a population of computers, wherein individual snapshots include data indicating a state of a respective computer;	<i>See</i> [1A]. <i>Id.</i>
[12B] automatically generating an adaptive reference model comprising a rule set customized to characteristics of the population of computers, the rule set being developed by identifying patterns among the snapshots from the plurality of computers such that the adaptive reference model is	<i>See</i> [1B]. <i>Id.</i>

indicative of normal states in the computers within the population; and	
[12C] comparing a snapshot from at least one of the computers to the adaptive reference model to determine whether an anomaly is present in the state of the at least one of the computers.	See [1C]. <i>Id.</i>

I. Claims 17-19, 21-22

Claims 17-19, 21-22 are substantially the same as claims 7-11 and are invalid for the same reasons as set forth below:

Claim	Reasons for Invalidity
17	See claim 9. EX1003, ¶124.
18	See claim 10. <i>Id.</i>
19	See claim 11. <i>Id.</i>
21	See claim 7. <i>Id.</i>
22	See claim 8. <i>Id.</i>

VII. GROUND 3 – CLAIMS 2-6, 13-16, 20 ARE INVALID AS RENDERED OBVIOUS BY HONIG OVER OLIPHANT

A. Overview of Oliphant

Oliphant disclose a real-time vulnerability monitoring system that automatically applies different remediations to fix a problem. EX1005, Abstract. Oliphant teaches a database that stores "data indicating the installed operating system(s), installed software, patches that have been applied, system policies that

are in place, and configuration information" of different devices in a network. *Id.*,

¶5. Oliphant also discloses a "vulnerability and remediation database" that maps remediation techniques that can be used to resolve different vulnerabilities. *Id.*,

¶¶10-11.

According to Oliphant, when a "security server" determines that there was an attempted attack, it may select "one or more remediation techniques" from the database to "remediate that particular vulnerability." *Id.*, ¶22. Oliphant explains that "remediation techniques include the closing of open ports on the device; installation of a patch that is known to correct the vulnerability; changing the device's configuration; stopping, disabling, or removing services; setting or modifying policies; and the like." *Id.*, ¶23.

B. Motivation to combine Honig and Oliphant

It would have been obvious to a POSITA to combine the teachings of Honig and Oliphant. Both references are in the same general field of real-time malware and vulnerability detection. EX1004, 6:65-67 ("Real time components such as sensors and detectors collect information from the monitored system and detect intrusions in real time."); EX1005, Title ("Real-Time Vulnerability Monitoring"); EX1010, 9 (describing "Anti-Vulnerability Technology"). Honig discloses a specific method for detecting malware in real-time by comparing a snapshot of a computer against a "detection model." EX1004, 11:8-9. Honig explains that, if the

analyzed "data corresponds to an attack, the detector **20** generates an alert." *Id.*,

8:56-57. A POSITA would have understood (or at minimum would have found obvious) that the "alert" in Honig would trigger the expectation that some action be taken in response to the alert, or else there would be no point to the alert. EX1003, ¶138.

For example, a POSITA would have been motivated to take further remediation steps in response to the alert to prevent any further damage to the network and prevent propagation of the malware to other computers. EX1003, ¶139. A POSITA therefore would have been motivated to look for other real-time detection systems to see what remediation steps could be taken. A POSITA would have understood that a network administrator could take multiple different actions in response to the alert, including, for example, limit network access of the device causing the alert or install security patches to address the problem. Oliphant discloses multiple examples of those potential responses. *Id.* ¶141. According to Oliphant, when a "security server" determines that there was an attempted attack, it may select "one or more remediation techniques" from the database to "remediate that particular vulnerability." EX1005, ¶22; EX1010, 9-10 (explaining that the system "[p]roactively remediates the vulnerability" automatically). The detection of the attempted attacks in Oliphant are the same kind of alerts generated in Honig. EX1003, ¶141. Oliphant explains that "remediation techniques include the closing

of open ports on the device; installation of a patch that is known to correct the vulnerability; changing the device's configuration; stopping, disabling, or removing services; setting or modifying policies; and the like." EX1005, ¶23; EX1010, 9-10 ("a remediation function is provided that may remotely and automatically deploy the appropriate update to each machine or device, install it, and document it with change tracking and audit trail reports").

And many of the remediation techniques disclosed by Oliphant would have been obviously beneficial and obvious to implement in Honig's system. For example, a POSITA would have understood that "stopping, disabling, or removing services" as disclosed in Oliphant would have prevented harm to the computer and prevented propagation to other devices on the network when certain malware is detected. EX1005, ¶23; EX1010, 9-10; EX1003, ¶142. As another example, closing certain ports on a device, as disclosed in Oliphant, would have the obvious benefit of preventing attacks that may target those ports or use those ports to propagate to other machines. EX1005, ¶23; EX1010, 9-10; EX1003, ¶142.

A POSITA would have recognized that it would have been beneficial to combine the remediation techniques disclosed in Oliphant with the detection system disclosed in Honig to prevent any further damage to the network and prevent propagation of the malware to other computers. EX1003, ¶142. In other words, Honig in combination with Oliphant discloses a system that generates alerts

in response to detected anomalies and a remediation system that responds to those alerts by taking appropriate action.

And a POSITA would have had a reasonable expectation of success in doing so. A POSITA would have understood that, once an attack is detected by the system in Honig, the attack could be looked up in the remediation database disclosed in Oliphant to determine what remediation techniques are available for that particular attack. EX1003, ¶143. It would have been obvious to a POSITA that the attacks detected by Honig are often linked to vulnerabilities as well. *Id.* It would further have been obvious to a POSITA to try to determine what vulnerability is targeted by a detected attack. *Id.*

Furthermore, it would have been obvious to a POSITA that the alerts in Honig contain sufficient information for the remediation system in Oliphant to determine what action should be taken. For example, Honig teaches that alerts will include what signature matched to generate the alert. EX1004, 11:15-24. A POSITA would have understood that the signature matching could be used to look up that signature in the remediation database of Oliphant. EX1003, ¶¶144-145.

As such, a POSITA would have been motivated to combine the system in Honig with that teaching in Oliphant and would have had a reasonable expectation of success for the foregoing reasons. *Id.*

- C. **Claim 2: "The method of claim 1, further comprising: comparing the anomaly to a recognition filter to diagnose a trouble condition on the at least one of the computers; and in the event of a trouble condition, generating an automated response to the trouble condition."**

The combination of Honig and Oliphant, as described above, renders obvious this limitation. For example, after an attack/anomaly is detected by the detector in Honig, the Honig-Oliphant combination would compare the detected attack against the attacks in the remediation database disclosed in Oliphant (recognition filter) to diagnose the issue. And, if there is a match in the remediation database (diagnose a trouble condition), Oliphant teaches taking different remediation measures (automated responses), such as "closing of open ports on the device; installation of a patch that is known to correct the vulnerability; changing the device's configuration; stopping, disabling, or removing services; setting or modifying policies; and the like." EX1005, ¶¶22-23; EX1010, 9-10; EX1003, ¶147.

And, it would have been obvious to a POSITA that the security server itself in Oliphant, not some administrator, would automatically take the appropriate remediation action, and keep the remediation database up to date. EX1005, ¶11; EX1010, 9-10 ("The platform will independently identify which network machines and devices have vulnerabilities by querying a client agent, or a device's OS or firmware directly."); EX1003, ¶148.

- D. Claim 3: "The method of claim 2, wherein the automated response is a generic response not specific to a particular asset of the at least one computer, and wherein the method further comprises: sending the generic response and a set of anomalies found in the snapshot to the at least one computer, the set of anomalies indicating assets of the at least one computer whose states are anomalous; and applying the generic response to the anomalous assets to correct the trouble condition."**

Honig discloses sending an alert in addition to the remediation measures of Oliphant. It would have been obvious to a POSITA to include in the alert the issues that were detected (set of anomalies found in the snapshot) to the computer so that the user or her computer could take further action to address the situation, such as by deleting the anomalous program or file. EX1003, ¶149.

Furthermore, the remediation measures (automated responses) disclosed in Oliphant include "closing of open ports on the device," "changing the device's configuration," "installation of a patch," "stopping, disabling, or removing services," and "setting or modifying policies." EX1005, ¶23; EX1010, 9-10; EX1003, ¶150. Oliphant discloses that these responses are applied on the computer to address the problem on the computer. EX1003, ¶150. And, although the term "generic response" is unclear, at least some of these generic responses are enumerated in claim 4, which depends from claim 3 and therefore must be within the scope of claim 3. *See* claim 4, *infra*; EX1003, ¶150.

- E. Claim 4: "The method of claim 3, wherein the generic response includes at least one of: installing a missing software component, removing an undesirable software component, and restoring an incorrect registry setting."**

Oliphant discloses that one of the remediation techniques is "installation of a patch that is known to correct the vulnerability" (installing a missing software component). EX1005, ¶23; EX1010, 9-10. Oliphant also discloses that another remediation technique is "removing services" (removing an undesirable software component). *Id.* As such, Honig over Oliphant renders obvious at least two of the generic responses in claim 4, and it would have been obvious to combine those references for the reasons discussed above. EX1003, ¶151.

- F. Claim 5: "The method of claim 2, wherein the recognition filter comprises a particular pattern of anomalies that indicates the presence of a particular root cause condition or a generic class of conditions."**

Based upon the disclosures in Honig, a POSITA would have understood that it detects a pattern of anomalies. EX1004, 6:17-20, 7:42-48.

Based upon the disclosures in Oliphant, a POSITA would have understood that these coordinated distributed attacks comprise a pattern of anomalies that indicate the presence of a distributed attack (particular root cause). As such, Oliphant discloses checking the remediation database for those attacks, which indicates a pattern of anomalies that indicate the presence of a distributed attack (root cause). EX1005, ¶¶22-23; EX1010, 9-10; EX1003, ¶¶152-153.

- G. Claim 6: "The method of claim 1, further comprising: comparing a plurality of anomalies associated with a particular snapshot with a recognition filter to diagnose a trouble condition; and diagnosing a trouble condition on the at least one of the computers in response to at least a subset of the plurality of anomalies matching information in the recognition filter."**

Honig discloses that "there are multiple sensors **12** for each host." EX1004, 9:49-50. Honig explains that, because there are multiple sensors, there may be multiple attacks (plurality of anomalies) detected for each snapshot from the host. EX1003, ¶154.

Oliphant discloses searching for one or more of these anomalies from the snapshot in the remediation database because, in many, if not most or all circumstances, there would be multiple problems that need to be addressed. EX1005, ¶¶22-23; EX1010, 9-10; EX1003, ¶155.

It would further have been obvious to a POSITA that multiple anomalies could have been caused by the same underlying problem and could be addressed by a single remediation measure.

It would therefore have been obvious to a POSITA that multiple anomalies would be mapped to the same underlying trouble condition (vulnerability) and would match the same information in the remediation database. EX1003, ¶¶155-156.

H. Claims 13-16, 20

Claims 13-16 and 20 are substantially the same as claims 2-6 and are invalid for the same reasons as set forth below:

Claim	Reasons for Invalidity
13	<i>See</i> claim 2. EX1003, ¶157.
14	<i>See</i> claim 3. <i>Id.</i>
15	<i>See</i> claim 4. <i>Id.</i>
16	<i>See</i> claim 5. <i>Id.</i>
20	<i>See</i> claim 6. <i>Id.</i>

VIII. GROUND 4 – CLAIMS 1, 7-12, 17-19, 21-22 ARE INVALID AS RENDERED OBVIOUS BY VINBERG

A. Overview of Vinberg

Vinberg generally discloses a system for monitoring different computers with pattern recognition and neural networks to determine whether they are in a good or bad state. EX1006, Abstract. Figure 1 depicts the overall architecture of the system.

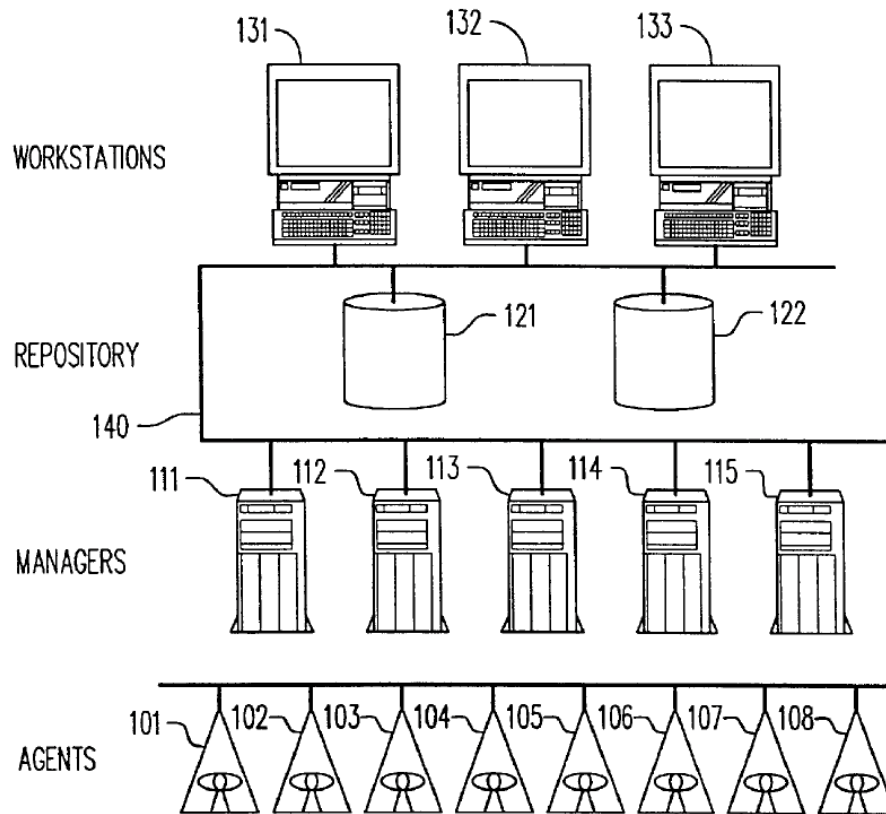


FIG. 1

Vinberg explains that the system will first go through a "learning phase" where agents **101-108** collect data from different computers in a network. This data is collected by the managers **111-115** and stored in the repository **121-122**. *Id.*, 5:42-61, 8:1-5. "When sufficient data has been collected," the managers "analyze[] this data with pattern recognition techniques to find 'common modes' of the system operation." *Id.*, 8:6-10. Then, in the "monitoring phase," the managers "receive[] state information from the agents **101-108** on a continual basis ... and compare[] these state vectors with the common modes that make up the model

stored" by the managers. *Id.*, 8:16-21. Vinberg also discloses using data in the monitoring phase to automatically supplement the learning phase. *Id.*, 8:16-21.

When the state vector of a monitored machine is outside of the "common modes" model, the system "raises an alarm." *Id.*, 8:24-27.

B. Claim 1

1. [1PRE]: "A method of detecting abnormal system states in computers, comprising:"

To the extent the preamble is limiting, Vinberg discloses a system that detects when a computer's state is in "an uncommon or unprecedented situation and raises an alarm." EX1006, 8:26-27. A POSITA would have understood that the uncommon and unprecedented situations are abnormal system states in a computer. EX1003, ¶161.

2. [1A]: "receiving snapshots from a plurality of computers within a population of computers, wherein individual snapshots include data indicating a state of a respective computer;"

In the "learning phase," Vinberg discloses receiving data defining the current state of the system from a plurality of agents **101-108** about the different computers in the managed system. EX1006, 8:1-5. It would have been obvious to a POSITA that, in one implementation of Vinberg, each agent would be installed on a different computer in the managed system. EX1003, ¶¶162-163. Vinberg further discloses that the agents collect "state information" in the form of "state vectors." EX1006, 8:16-21. It would have been obvious to a POSITA based upon

this disclosure that the data collected from the plurality of agents during the "learning phase" is also in the form of a state vector. EX1003, ¶¶162-163. Indeed, Vinberg also discloses using data collected in later phases to automatically supplement the learning phase, meaning that the data collected in the learning phase is the same as the data collected later. EX1006, 8:16-21.

It further would have been obvious to a POSITA that the state vectors are a snapshot that indicates a "state" of the computer from which it was collected. EX1003, ¶164. For example, Vinberg discloses that the state vector is made up of "n metrics." EX1006, 2:24. Vinberg further explains that this information includes "information from the managed network, hardware and software systems." *Id.*, 6:26-31. A POSITA, therefore, would have understood that the state vector included many different metrics, not just one. *See* claim 7, *infra*.

3. **[1B]: "automatically generating an adaptive reference model comprising a rule set customized to characteristics of the population of computers, the rule set being developed by identifying patterns among the snapshots from the plurality of computers such that the adaptive reference model is indicative of normal states in the computers within the population; and"**

As another part of the learning phase, Vinberg explains that "[w]hen sufficient data has been collected," the system managers will "analyze[] this data with pattern recognition techniques to find 'common modes' of system operation." EX1006, 8:6-10. "A common mode is a region in the n-dimensional space in

which the state vectors appear that represents a cluster of commonly occurring state vectors." *Id.*, 8:10-12. In sum, a POSITA would have recognized that the "common modes" is a form of an adaptive reference model, or rule set that is customized to the different computers from which data was collected in the first part of the learning phase. *Id.*, 8:1-5; EX1003, ¶¶165-166. A POSITA further would have understood that the common modes are developed by identifying clusters (patterns) from the snapshots from the plurality of computers. EX1003, ¶¶165-166. And, a POSITA would have recognized that the common modes equate to the normal states in the computers in that population, including based upon the name "*common*." *Id.*

4. [1C]: "comparing a snapshot from at least one of the computers to the adaptive reference model to determine whether an anomaly is present in the state of the at least one of the computers."

Vinberg explains that, in the "monitoring phase," the managers will receive "state information from the agents **101-108** on a continual basis, collected through the services of the data collection manager **113**, and *compares these state vectors with the common modes*." EX1006, 8:17-31. In other words, Vinberg discloses comparing a state vector from a computer (snapshot) to the common modes (adaptive reference model) to determine whether the computer is in a recognized state, or an unrecognized state. EX1003, ¶167. It would have been obvious to a

POSITA that, when a computer is in an unrecognized state, there would be some sort of anomaly present in the state vector that led to that classification. *Id.*, ¶168.

C. Claim 7: "The method of claim 1, wherein individual snapshots include data associated with at least one of: system files, application files, a registry entry, a performance counter, a process, a communication port, a hardware configuration, a log file, a running task, services, and network connections."

Although Vinberg does not explicitly state the data included within the collected state vectors, it discloses that it can include "n metrics." EX1006, 2:24. Vinberg further explains that this information includes "information from the managed network, hardware and software systems." *Id.*, 6:26-31. Given these disclosures, a POSITA would have understood or it would have been obvious to a POSITA that at least one of the enumerated metrics are included in the Vinberg state vector. For example, performance counters and registry entries are commonly taken metrics from a computer, and a POSITA would have known that those metrics could be easily derived and would likely be indicative of whether a computer was in a normal state. EX1003, ¶169. A POSITA, therefore, would have been motivated to include them in the state vector. *Id.*

D. Claim 8: "The method of claim 1, further comprising: inserting rules into the rule set of the adaptive reference model to augment or override rules of the rule set automatically generated from the snapshots."

Vinberg discloses that its system "continuously updates its degree of learning during its monitoring phase." EX1006, 2:39-40. Vinberg explains that an

"automated management tool studies the alarm message, and if necessary inspects the managed system in more detail, and then determines what should be done. If it is determined that the alarm message referred to a common situation that should be included among the common modes for future monitoring, the invention can add this data point to the set of data used for tutoring. If certain techniques such as neural networks are used for pattern recognition, the inventive system is able incrementally to learn a new state and add the new state to the set of common modes." *Id.*, 2:41-50. In other words, Vinberg discloses inserting new adding new data points (rules) into the common modes (rule set) to augment the rules for future monitoring. EX1003, ¶170. And Vinberg discloses automatically doing so through, for example, the use of neural networks. *Id.*

E. Claim 9: "The method of claim 1, wherein the adaptive reference model is generated to include a value layer that determines whether an asset value contained in a snapshot is anomalous."

Vinberg discloses that it will consider situations where there are "a few metrics with unprecedented values" or "many metrics with unprecedented values." EX1006, 8:40-44. In other words, Vinberg discloses a layer that determines whether particular metrics (asset values) are anomalous. EX1003, ¶¶171-173.

And further supporting that Vinberg discloses claim 9, the '936 patent discloses that the "value layer 406 tracks the values of asset/value pairs provided

by the Agent component." EX1001, 10:37-38. Vinberg discloses precisely that.

EX1006, 8:40-44, 2:62-3:6.

It would further have been obvious to a POSITA that the detection model described in Vinberg would track the values of the metrics to check for anomalies associated with those metrics. EX1003, ¶¶174-175.

F. Claim 10: "The method of claim 1, wherein the adaptive reference model is generated to include a cluster layer that tracks relationships between assets and identifies an anomaly in response to an asset being unexpectedly absent from or present in a set of assets in a snapshot."

Vinberg further discloses that the system will identify "strongly correlated" and "combinations of metrics." EX1006, 8:57-58. In other words, Vinberg discloses tracking relationships (correlations and combinations) between the different metrics (assets), and uses those relationships as part of its analysis. *Id.*, 8:56-60; EX1003, ¶¶176-177.

Vinberg further discloses considering different predefined cases, including:

- the state contains a few metrics with ***unprecedented values***, but is within the overall bounds of the historical data;
- the state contains many metrics with ***unprecedented values***, but is within the overall bounds of the historical data;
- all metrics individually have familiar values but the ***combination of values is unprecedented***, although still within the overall bounds of the historical data.

EX1006, 8:40-49. It would have been obvious to a POSITA that the

"unprecedented values" would have included situations where a metric is

unexpectedly absent (e.g., null). EX1003, ¶¶178-179. And, it further would have been obvious that, by determining that a **combination** of values is unprecedented, the Vinberg system necessarily was tracking the relationships between the values as part of the common modes. EX1003, ¶¶178-179.

And further supporting that Vinberg discloses or renders obvious claim 10, the '936 patent discloses that the "cluster layer **408** tracks relationships between asset names." EX1001, 11:58-59. Vinberg discloses checking for unprecedented combinations of values, which necessarily means it is tracking the relationships between different metrics. EX1003, ¶¶178-179.

It further would have been obvious to a POSITA that checking for unprecedented combinations of values necessarily means it is tracking the relationships between different metrics. EX1003, ¶¶179-181.

G. Claim 11: "The method of claim 1, wherein the adaptive reference model is generated to include a profile layer that identifies anomalies in response to violation of relationships of clusters of assets in a snapshot."

As in claim 10, Vinberg discloses identifying "strongly correlated" and "combinations of metrics." EX1006, 8:57-58. And Vinberg discloses identifying cases (anomalies) where "all metrics individually have familiar values but the combination of values is unprecedented, although still within the overall bounds of the historical data." *Id.*, 8:46-49. In sum, a POSITA would have understood this to mean that the Vinberg system identifies anomalies based upon a violation of the

relationship between the different metrics (assets), not just the values of the metrics themselves. EX1003, ¶¶182-183.

And further supporting that Vinberg discloses claim 11, the '936 patent discloses that the "profile layer **410** in the embodiment shown detects anomalies based upon violations of cluster relationships." EX1001, 12:8-10. Vinberg discloses identifying anomalies based upon a violation of the relationship between the different metrics (assets), not just the values of the metrics themselves. EX1003, ¶184.

It would further have been obvious to a POSITA that Vinberg detects anomalies based upon relationships between asset names having anomalous values. EX1003, ¶¶183-186.

H. Claim 12

Claim 12 is substantially similar to claim 1 except it recites a computer-readable medium instead of a method. It is invalid over Vinberg for the same reasons detailed in claim 1 because a POSITA would have known that the instructions for executing that method would be stored on a computer-readable medium, so it could be executed by a computer based upon the following mapping. EX1006, 6:32-57; EX1003, ¶187.

Claim	Reasons for Invalidity
[12PRE] A computer readable medium storing instructions, that when executed by a	See [1PRE]. EX1003, ¶187.

Claim	Reasons for Invalidity
computer, cause the computer to perform functions of:	
[12A] receiving snapshots from a plurality of computers within a population of computers, wherein individual snapshots include data indicating a state of a respective computer;	<i>See</i> [1A]. <i>Id.</i>
[12B] automatically generating an adaptive reference model comprising a rule set customized to characteristics of the population of computers, the rule set being developed by identifying patterns among the snapshots from the plurality of computers such that the adaptive reference model is indicative of normal states in the computers within the population; and	<i>See</i> [1B]. <i>Id.</i>
[12C] comparing a snapshot from at least one of the computers to the adaptive reference model to determine whether an anomaly is present in the state of the at least one of the computers.	<i>See</i> [1C]. <i>Id.</i>

I. Claims 17-19, 21-22

Claims 17-19, 21-22 are substantially the same as claims 7-11 and are invalid for the same reasons as set forth below:

Claim	Reasons for Invalidity
17	<i>See</i> claim 9. EX1003, ¶188.
18	<i>See</i> claim 10. <i>Id.</i>
19	<i>See</i> claim 11. <i>Id.</i>

Claim	Reasons for Invalidity
21	<i>See claim 7. Id.</i>
22	<i>See claim 8. Id.</i>

IX. GROUND 5 – CLAIMS 2-6, 13-16, 20 ARE INVALID AS RENDERED OBVIOUS BY VINBERG OVER OLIPHANT

A. Motivation to combine Vinberg and Oliphant

It would have been obvious to a POSITA to combine the teachings of Vinberg and Oliphant. Both references are in the same general field of malware and vulnerability detection. EX1006, 9:7-15 (explaining that it will determine whether a computer is in a "good" or "bad" state); EX1005, Title ("Real-Time Vulnerability Monitoring"). Vinberg discloses a specific method for detecting anomalies in a computer by comparing the "state vector" of a computer against a "common mode" rule set. EX1006, 8:1-21. Vinberg explains that, if the state vector of a computer does not relate to a known common mode, it will "raise[] an alarm." *Id.*, 8:27. A POSITA would have understood that the "alarm" in Vinberg would trigger the expectation that some action be taken in response to the alarm, or else there would be no point to the alarm. EX1003, ¶189.

For example, a POSITA would have been motivated to take further remediation steps in response to the alert to prevent any further damage to the network and prevent propagation of the malware to other computers. EX1003,

¶190. A POSITA, therefore, would have been motivated to look for other real-time detection systems to see what remediation steps could be taken. A POSITA would have understood that a network administrator could take multiple different actions in response to the alert, including, for example, limit network access of the device causing the alert or install security patches to address the problem.

Oliphant discloses multiple examples of those potential responses. *Id.* According to Oliphant, when a "security server" determines that there was an attempted attack, it may select "one or more remediation techniques" from the database to "remediate that particular vulnerability." EX1005, ¶22; EX1010, 9-10. The detection of the attempted attacks in Oliphant is the same kind of alarms generated in Vinberg. EX1003, ¶¶191-192. Oliphant explains that "remediation techniques include the closing of open ports on the device; installation of a patch that is known to correct the vulnerability; changing the device's configuration; stopping, disabling, or removing services; setting or modifying policies; and the like." EX1005, ¶23; EX1010, 9-10.

And many of the remediation techniques disclosed by Oliphant would have been obviously beneficial and obvious to implement in Vinberg's system. For example, a POSITA would have understood that "stopping, disabling, or removing services" as disclosed in Oliphant would have prevented harm to the computer and prevented propagation to other devices on the network when certain malware is

detected. EX1005, ¶23; EX1010, 9-10; EX1003, ¶193. As another example, closing certain ports on a device, as disclosed in Oliphant, would have the obvious benefit of preventing attacks that may target those ports or use those ports to propagate to other machines. EX1005, ¶23; EX1010, 9-10; EX1003, ¶193.

A POSITA would have recognized that it would have been beneficial to combine the remediation techniques disclosed in Oliphant with the detection system disclosed in Vinberg to prevent any further damage to the network and prevent propagation of the malware to other computers. EX1003, ¶193. In other words, Vinberg in combination with Oliphant discloses a system that generates alarms in response to detected anomalies and a remediation system that responds to those alerts by taking appropriate action.

And, a POSITA would have had a reasonable expectation of success in doing so. A POSITA would have understood that, once an attack is detected by the system in Vinberg, the attack could be looked up in the remediation database disclosed in Oliphant to determine what remediation techniques are available for that particular attack. EX1003, ¶194. It would have been obvious to a POSITA that the attacks detected by Vinberg are often linked to vulnerabilities as well. *Id.* It would further have been obvious to a POSITA to try to determine what vulnerability is targeted by a detected attack. *Id.*

Furthermore, it would have been obvious to a POSITA that the alarms in Vinberg contain sufficient information for the remediation system in Oliphant to determine what action should be taken. For example, Vinberg teaches that alarms report the monitored system that it has entered an unusual state and includes enough detail to "what should be done." EX1006, 2:35-43. A POSITA would have understood that the signature matching could be used to look up that signature in the remediation database of Oliphant. EX1003, ¶¶195-196.

As such, a POSITA would have been motivated to combine the system in Vinberg with that teaching in Oliphant and would have had a reasonable expectation of success for the foregoing reasons. *Id.*, ¶¶195-197.

B. Claim 2: "The method of claim 1, further comprising: comparing the anomaly to a recognition filter to diagnose a trouble condition on the at least one of the computers; and in the event of a trouble condition, generating an automated response to the trouble condition."

The combination of Vinberg and Oliphant, as described above, renders obvious this limitation. For example, after an attack/anomaly is detected by the common modes in Vinberg, the Vinberg-Oliphant combination would compare the detected attack against the attacks in the remediation database disclosed in Oliphant (recognition filter) to diagnose the issue. And, if there is a match in the remediation database (diagnose a trouble condition), Oliphant teaches taking different remediation measures (automated responses), such as "closing of open

ports on the device; installation of a patch that is known to correct the vulnerability; changing the device's configuration; stopping, disabling, or removing services; setting or modifying policies; and the like." EX1005, ¶¶22-23; EX1010, 9-10; EX1003, ¶198. And, it would have been obvious to a POSITA that the security server itself in Oliphant, not some administrator, would automatically take the appropriate remediation action, and keep the remediation database up to date. EX1005, ¶11; EX1010, 9-10; EX1003, ¶199.

C. Claim 3: "The method of claim 2, wherein the automated response is a generic response not specific to a particular asset of the at least one computer, and wherein the method further comprises: sending the generic response and a set of anomalies found in the snapshot to the at least one computer, the set of anomalies indicating assets of the at least one computer whose states are anomalous; and applying the generic response to the anomalous assets to correct the trouble condition."

Vinberg discloses raising an alarm in addition to the remediation measures of Oliphant. It would have been obvious to a POSITA to include the issues that were detected (set of anomalies found in the snapshot) in the alarm to the computer so that the user or her computer could take further action to address the situation, such as by deleting the anomalous program or file. EX1003, ¶200.

Furthermore, the remediation measures (automated responses) disclosed in Oliphant include "closing of open ports on the device," "changing the device's configuration," "installation of a patch," "stopping, disabling, or removing services," and "setting or modifying policies." EX1005, ¶23; EX1010, 9-10;

EX1003, ¶201. Oliphant discloses that these responses are applied on the computer to address the problem on the computer. EX1003, ¶201. And, although the term "generic response" is unclear, at least some of these generic responses are enumerated in claim 4, which depends from claim 3 and therefore must be within the scope of claim 3. *See* claim 4, *infra*; EX1003, ¶201.

D. Claim 4: "The method of claim 3, wherein the generic response includes at least one of: installing a missing software component, removing an undesirable software component, and restoring an incorrect registry setting."

Oliphant discloses that one of the remediation techniques is "installation of a patch that is known to correct the vulnerability" (installing a missing software component). EX1005, ¶23; EX1010, 9-10. Oliphant also discloses that another remediation technique is "removing services" (removing an undesirable software component). *Id.* As such, Vinberg over Oliphant renders obvious at least two of the generic responses in claim 4, and it would have been obvious to combine those references for the reasons discussed above. EX1003, ¶202.

E. Claim 5: "The method of claim 2, wherein the recognition filter comprises a particular pattern of anomalies that indicates the presence of a particular root cause condition or a generic class of conditions."

Based upon the disclosure in Vinberg, a POSITA would have understood that it detects a pattern of anomalies. EX1006, 8:40-55.

Based upon the disclosures in Oliphant, a POSITA would have understood that this historical data analysis constitutes a pattern of anomalies that indicate the presence of a potential attack (particular root cause). As such, Oliphant discloses checking the remediation database for those attacks, which indicates a pattern of anomalies that indicate the presence of patterns in the historical data (root cause). EX1005, ¶¶22-23; EX1010, 9-10; EX1003, ¶¶203-204.

F. Claim 6: "The method of claim 1, further comprising: comparing a plurality of anomalies associated with a particular snapshot with a recognition filter to diagnose a trouble condition; and diagnosing a trouble condition on the at least one of the computers in response to at least a subset of the plurality of anomalies matching information in the recognition filter."

In comparing the snapshot to the common modes, Vinberg discloses determining between different cases, including determining when "the state is outside the bounds of the historical data and far from any common mode."

EX1006, 8:54-55. A POSITA would have understood that there are multiple anomalies associated with this case because it is both outside the bounds of historical data (anomaly 1) and far from any common mode (anomaly 2). EX1003, ¶205.

Oliphant discloses searching for one or more of these anomalies from the snapshot in the remediation database because in many, if not most or all circumstances, there would be multiple problems that need to be addressed.

EX1005, ¶¶22-23; EX1010, 9-10; EX1003, ¶206.

It would further have been obvious to a POSITA that multiple anomalies could have been caused by the same underlying problem and could be addressed by a single remediation measure. EX1003, ¶207.

It would therefore have been obvious to a POSITA that multiple anomalies would be mapped to the same underlying trouble condition (vulnerability) and would match the same information in the remediation database. *Id.*

G. Claims 13-16, 20

Claims 13-16 and 20 are substantially the same as claims 2-6 and are invalid for the same reasons as set forth below:

Claim	Reasons for Invalidity
13	<i>See</i> claim 2. EX1003, ¶208.
14	<i>See</i> claim 3. <i>Id.</i>
15	<i>See</i> claim 4. <i>Id.</i>
16	<i>See</i> claim 5. <i>Id.</i>
20	<i>See</i> claim 6. <i>Id.</i>

X. MANDATORY NOTICES UNDER 37 C.F.R. §42.8(a)(1)

A. 37 C.F.R. §42.8(b)(1): Real Parties-In-Interest

Petitioner is the real party-in-interest.

B. 37 C.F.R. §42.8(b)(2): Related Matters

The '936 patent is asserted against Petitioner in *Netskope, Inc. v. Fortinet*,

Inc., No. 3:25-cv-2360 (N.D. Cal.).

C. 37 C.F.R. §42.8(b)(3)-(4): Lead And Back-Up Counsel And Service Information

Designated Counsel for Petitioner and service information is below:

Lead Counsel	Back-Up Counsel
Andrew D. Gish (Reg. # 67,562) GISH PLLC 41 Madison Avenue New York, New York 10010 Telephone: (212) 518-7380 andrew@gishpllc.com	Ryan Iwahashi (Reg. # 63,378) GISH PLLC 50 California Street, Suite 1500 San Francisco, CA 94111 Telephone: (415) 630-8960 ryan@gishpllc.com

Petitioner consents to service by email at the addresses above.

XI. FEES UNDER 37 C.F.R. §42.103

Petition and Post-Institution fees totaling \$54,695 have been paid by via the online payment portal.

XII. CONCLUSION

Petitioner requests the Board institute IPR of the Challenged Claims and cancel them.

Respectfully Submitted,

Dated: October 7, 2025

/s/Andrew Gish

Andrew Gish (Reg. #67,562)

ATTORNEY FOR PETITIONER

CERTIFICATE OF WORD COUNT UNDER 37 C.F.R. §42.24(a)

I, the undersigned, do hereby certify that the attached petition contains 12,781 words, as measured by the Word Count function of Microsoft Word. This is less than the limit of 14,000 words as specified by 37 C.F.R. §42.24(a)(i).

Dated: October 7, 2025

/s/ Andrew Gish

Andrew Gish (Reg. # 67,562)

ATTORNEY FOR PETITIONER

CERTIFICATION OF SERVICE

The undersigned certifies service pursuant to 37 C.F.R. §§42.6(e) and 42.105 on the Patent Owner of a copy of this Petition for *Inter Partes* Review and supporting materials via FedEx at the following correspondence address of record:

EDELL, SHAPIRO & FINNAN, LLC
9801 Washingtonian Blvd.
Suite 750
Gaithersburg, MD

Dated: October 7, 2024

/s/ Andrew Gish

Andrew Gish (Reg. # 67,562)

ATTORNEY FOR PETITIONER