

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETSKOPE, INC.,

Petitioner,

v.

FORTINET, INC.,

Patent Owner.

PTAB Case No. IPR 2023-00457

Patent No. 9,231,968 B2

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 9,231,968 B2**

TABLE OF CONTENTS

Page

EXHIBIT LIST	III
I. INTRODUCTION	1
II. MANDATORY NOTICES	1
A. Real party in interest.....	1
B. Related matters	1
C. Counsel and service information.....	2
III. IPR REQUIREMENTS	2
A. Standing.....	2
B. Challenged claims and relief requested.....	2
1. Prior art	3
2. Grounds	3
IV. TECHNOLOGY DESCRIPTION	4
A. Malware Detection	4
B. Updating Malware Detection Signatures	6
C. Distributed Malware Signature Updates	7
D. '968 Patent.....	10
1. Alleged Invention.....	10
2. Prosecution History.....	13
3. Priority	13
V. SECTION 325(D).....	19
VI. LEVEL OF ORDINARY SKILL	20
VII. CLAIM CONSTRUCTION	21
VIII. GROUNDS	21
A. Ground 1: Combining Jordan and FortiGate renders claims 1-9 obvious.....	21
1. Motivation to combine	25

TABLE OF CONTENTS
(continued)

	Page
2. Claims 1/4/7	26
3. Claims 2/5/8: “...wherein the analysis process, to which the network traffic content is forwarded, executes on a computing device distinct from the network gateway device[/data network device].”	61
4. Claims 3/6/9: “...wherein the analysis process facilitates an analysis that includes receiving administrator input.”	62
5. Jordan Is Prior Art.....	63
B. Ground 2: Combining Lee and Magdych renders claims 1-9 obvious.	68
1. Motivation to combine.....	71
2. Claims 1/4/7	72
3. Claims 2/5/8:	91
4. Claims 3/6/9:	92
C. Ground 3: Combining Lee, Magdych, and Judge renders claims 1-9 obvious.....	93
D. Ground 4: Combining Fang and FortiGate renders claims 1-9 obvious.	96
1. Motivation to combine.....	96
2. Claims 1/4/7:	97
3. Claims 2/5/8:	111
4. Claims 3/6/9:	112
IX. CONCLUSION.....	112

EXHIBIT LIST

Ex. 1001	U.S. Patent No. 9,231,968 B2 to Fang et al. (“the ’968 Patent”)
Ex. 1002	Supporting Declaration of Dr. Wenke Lee (“Wenke-Lee”)
Ex. 1003	Prosecution History of U.S. Patent No. 9,231,968 (the ’968 Prosecution History”)
Ex. 1004	Prosecution History of U.S. Patent App. No. 13/795,407 (the ’407 Prosecution History”)
Ex. 1005	U.S. Patent No. 8,839,417 B1 to Jordan (“Jordan”)
Ex. 1006	U.S. Provisional App. No. 60/552,457 to Fang et al. (“the ’457 provisional”)
Ex. 1007	U.S. Provisional App. No. 60/520,577 to Jordan (“the ’577 provisional”)
Ex. 1008	FortiGate – 3600 User Manual (“FortiGate”)
Ex. 1009	U.S. Patent App. Pub. No. 2004/0015719 A1 to Lee et al. (“Lee”)
Ex. 1010	U.S. Patent No. 6,513,122 B1 to Magdych et al. (“Magdych”)
Ex. 1011	U.S. Patent App. Pub. No. 2003/0172292 A1 to Judge (“Judge”)
Ex. 1012	U.S. Patent App. Pub. No. 2012/0278896 A1 to Fang et al. (“Fang”)
Ex. 1013	Heady et al., The Architecture of a Network Level Intrusion Detection System, UNIVERSITY OF NEW MEXICO (1990)
Ex. 1014	Paxson, <i>Bro a System for Detecting Network Intruders in Real-Time</i> , LAWRENCE BERKELEY NATIONAL LABORATORY (1998)
Ex. 1015	U.S. Pat. No. 5,440,723 to Arnold
Ex. 1016	Lee, <i>A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems</i> , COLUMBIA UNIVERSITY
Ex. 1017	U.S. Pat. App. Pub. 2003/0204632A1 to Willebeek-LeMair et al.
Ex. 1018	Kephart et al., <i>Fighting Computer Viruses</i> , SCIENTIFIC AMERICAN (1997)

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

I. INTRODUCTION

The notion of making and updating lists has been around for centuries. In particular, lists of threats—whether it is the FBI’s “10 Most Wanted” or a list of known computer viruses and malware—need to be updated constantly to ensure the most comprehensive protection. Despite that long and obvious history, Fortinet took this notion and made it the subject of U.S. Pat. No. 9,231,968. Indeed, the ’968 patent merely claims (1) analyzing network traffic for a threat (e.g., a virus), (2) checking if the threat is on a list of known threats, and (3) if not, updating the list to include the new threat. As numerous prior art references show, these common-sense steps were routine and well-understood before the ’968 patent, and therefore nothing it claims is novel. Accordingly, Netskope, Inc. (“Petitioner”) requests IPR and cancellation of all the ’968 claims.

II. MANDATORY NOTICES

A. Real party in interest

Netskope, Inc. is the sole real party-in-interest.

B. Related matters

Petitioner has filed a civil action seeking a declaratory judgment that it does not infringe the ’968 Patent in the United States District Court for the Northern District of California, Case No. 3:22-CV-01852-TLT. That case is currently stayed. Petitioner has challenged the validity of other patents allegedly owned by the Patent Owner in co-pending *inter partes* review proceedings. (See IPR 2023-00456,

IPR2023-00458, IPR2023-00459, IPR2022-01587, IPR2023-00175, IPR2023-00030.)

C. Counsel and service information

Lead counsel: Thomas N. Millikan (Reg. No. 72,316).

Back-up counsel: Babak Tehranchi (Reg. No. 55,937), Andrew N. Klein (*pro hac vice* forthcoming), Cory N. Owan (Reg. No. 79,841).

These attorneys can be reached by mail at Perkins Coie LLP, 11452 El Camino Real, Suite 300, San Diego, California, 92130, by phone at (858) 720-5700 and by fax at (858) 720-5799.

Petitioner consents to electronic service. All services and communications to the attorneys listed above may be sent to:

Netskope-FortinetIPRTeam@perkinscoie.com.

A Power of Attorney for Petitioner is being filed concurrently.

III. IPR REQUIREMENTS

A. Standing

The '968 Patent is available for IPR and Petitioner is not barred or estopped from challenging the claims on the grounds herein.

B. Challenged claims and relief requested

Petitioner respectfully requests cancellation of the challenged claims under §103.

1. Prior art

Petitioner relies on the references listed in the Table of Exhibits, including:

- U.S. Patent 8,839,417 A1 (“Jordan”)
- Fortinet’s FortiGate – 3600 User Manual (“FortiGate”)
- U.S. Patent Publication 2004/0015719 A1 (“Lee”)
- U.S. Patent 6,513,122 B1 (“Magdych”)
- U.S. Patent Publication 2003/0172292 A1 (“Judge”)
- U.S. Patent Publication 2012/0278896 A1 (“Fang”)

Due to written description issues (*see, infra*, Section IV.D.3), the challenged claims are entitled to a priority date of no earlier than November 5, 2013. Under that priority date, all references are prior art under AIA §102(a). But even if the Board finds adequate written description support, all references are prior art under pre-AIA §§102(a), 102(b), and/or 102(e), except Fang.

Petitioner also relies on the declaration of Dr. Lee (hereinafter “Wenke-Lee”).

2. Grounds

Ground	Basis	Reference(s)	Challenged Claims
1	§103	Jordan, FortiGate	1-9
2	§103	Lee, Magdych	1-9
3	§103	Lee, Magdych, Judge	1-9
4	§103	Fang, FortiGate	1-9

IV. TECHNOLOGY DESCRIPTION

A. Malware Detection

Preventing malicious traffic from entering a network has been a top concern for both the public and private sectors since the 1980s. That decade saw the first government-funded research into detecting and preventing cyberattacks as well as the first firewalls. Over time, various techniques were developed for detecting and remediating malicious traffic. (Wenke-Lee, ¶¶36-38.)

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

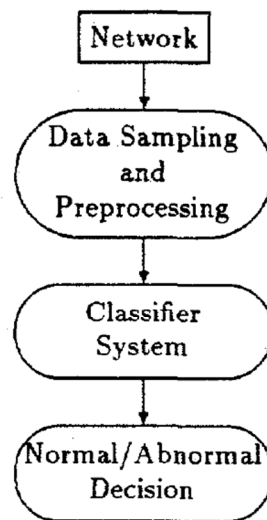


Figure 1: Basic architecture of intrusion detection system.

This figure illustrated the general process for an IDS system: a network device ran

monitoring software and captured potentially-malicious traffic. After analyzing the traffic, the software determined whether to allow or block the traffic. (Ex. 1013, 4-5; Wenke-Lee, ¶41.)

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

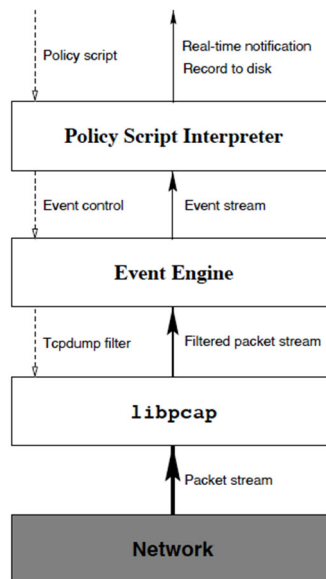


Figure 1: Structure of the Bro system

B. Updating Malware Detection Signatures

Malware detection signatures date back to at least 1993. For example, below is a figure from U.S. Patent No. 5,440,723 that depicted looking for a certain pattern in traffic to identify malicious data. (Wenke-Lee, ¶¶43-44; Ex. 1015.)

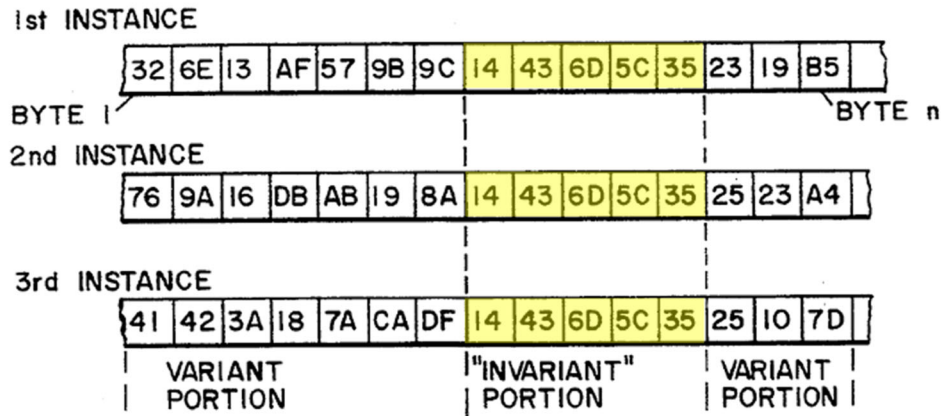


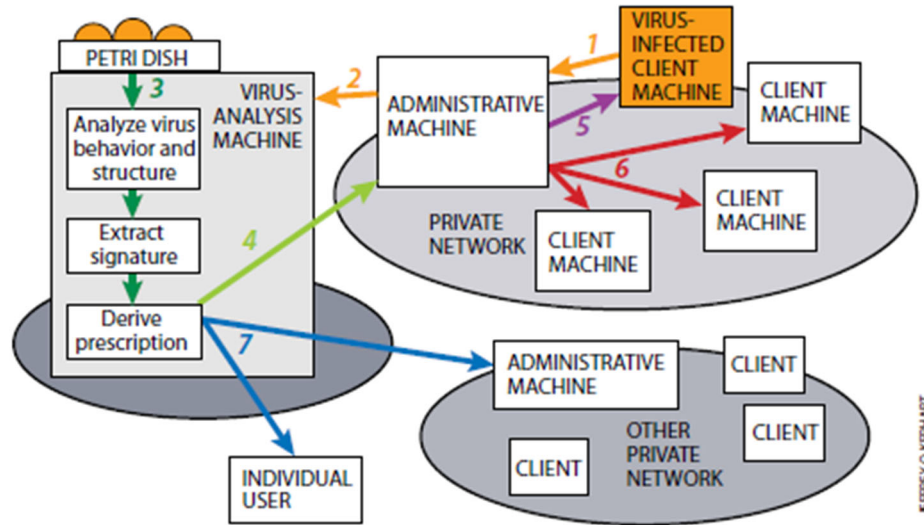
FIG. 6A (Annotated)

Given the importance of detection signatures, security systems needed both to generate new signatures in response to new threats and to update existing detection signatures in response to new information. (Wenke-Lee, ¶45.) This process could be performed manually or automated using machine learning algorithms, as illustrated by Dr. Lee's 1999 Ph.D. thesis ("A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems"). (Wenke-Lee, ¶45; Ex. 1016, 1.)

C. Distributed Malware Signature Updates

For decades, updated detection signatures have been distributed to multiple monitoring devices on a given network. (Wenke-Lee, ¶46.) Detection signature updates are usually carried out by a “back-end” analysis on a separate “update server” that runs the network security monitor. (*Id.*) The server can be operated by human analysts to manually produce the new and updated signatures or can run machine learning algorithms to automate these tasks. (*Id.*)

For example, around the late 1990’s, companies such as IBM, Symantec Corporation, and McAfee Associates were developing technology that automatically detected new malware and responded to it. (Wenke-Lee, ¶48; Ex. 1017.) Below is a figure describing a security system IBM was developing in the late 1990’s wherein a central computer analyzed viruses. (Wenke-Lee, ¶48; Ex. 1017.) Notably, the caption described a complete system for detecting and updating detection information. (Wenke-Lee, ¶48.)



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

Updates to the active set of detection signatures were often distributed according to specific policies. (Wenke-Lee, ¶49.) For example, a traffic monitor for the engineering department might get a different signature than the traffic monitor for the finance department subnet because the two departments had different security policies and hence different signatures and rules. (*Id.*) Also, to manage the potentially large workload and customize updates according to the specific needs of different (sub)networks, there could be multiple update servers to accomplish “distributed updates”—for example, different update servers for engineering and finance. (*Id.*)

Detection signatures were created using information from multiple networks. For example, U.S. Pat. App. Pub. 2003/020632 disclosed a security system that “collected threats from about the world.” (Wenke-Lee, ¶47; Ex. 1017., ¶57.) As illustrated in the Figure below, “[t]he collected information 130 [was] then analyzed and utilized by the network administrator 142 to design the detection signatures 132 (comprising, for example, security rules, policies and algorithms) that can be used by the system 10....” (Ex. 1017, ¶57.)

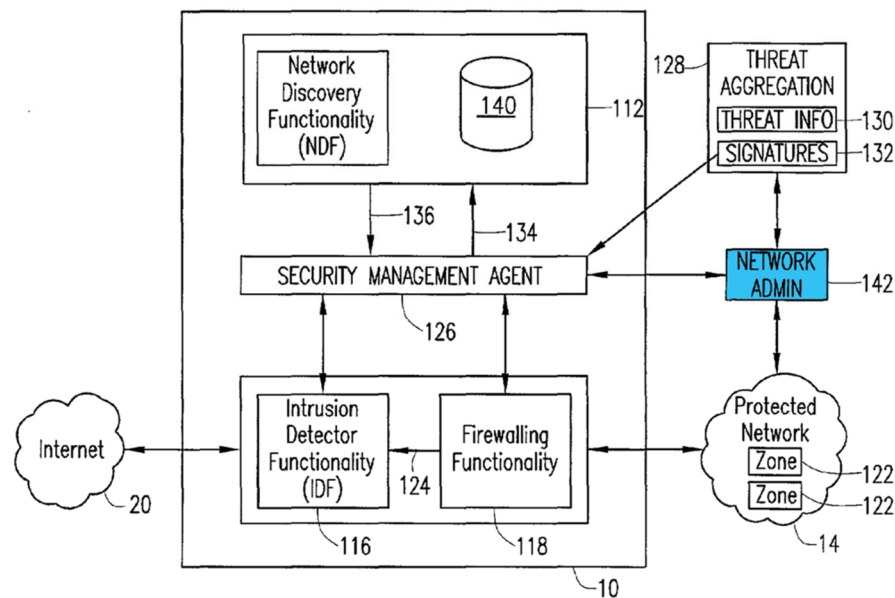


FIG. 2 (Annotated)

D. '968 Patent

1. Alleged Invention

The '968 patent concerns “[s]ystems, methods, and software for processing received network traffic content in view of content detection data and configuration data to either block, permit, or to further evaluate network traffic content when entering a network.” ('968 patent, Abstract; Wenke-Lee, ¶50.)

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

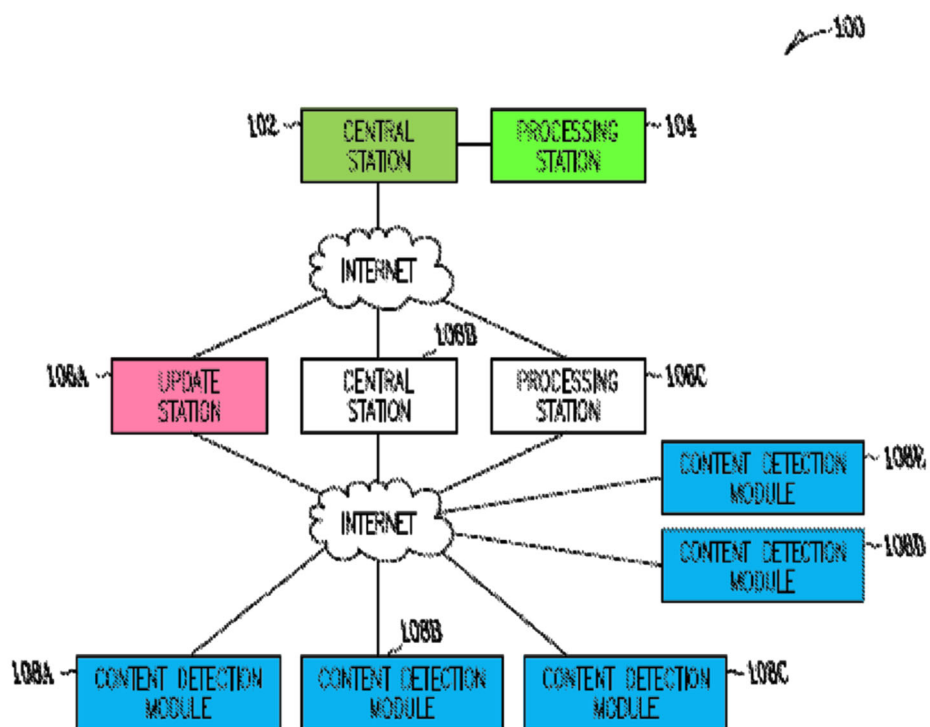


Fig. 1 (Annotated)

Petitioner has redrawn Figure 1 (below) to make it more legible.¹

¹ Elements 108b-c should read “update station.” (Wenke-Lee, ¶51, n.1.)

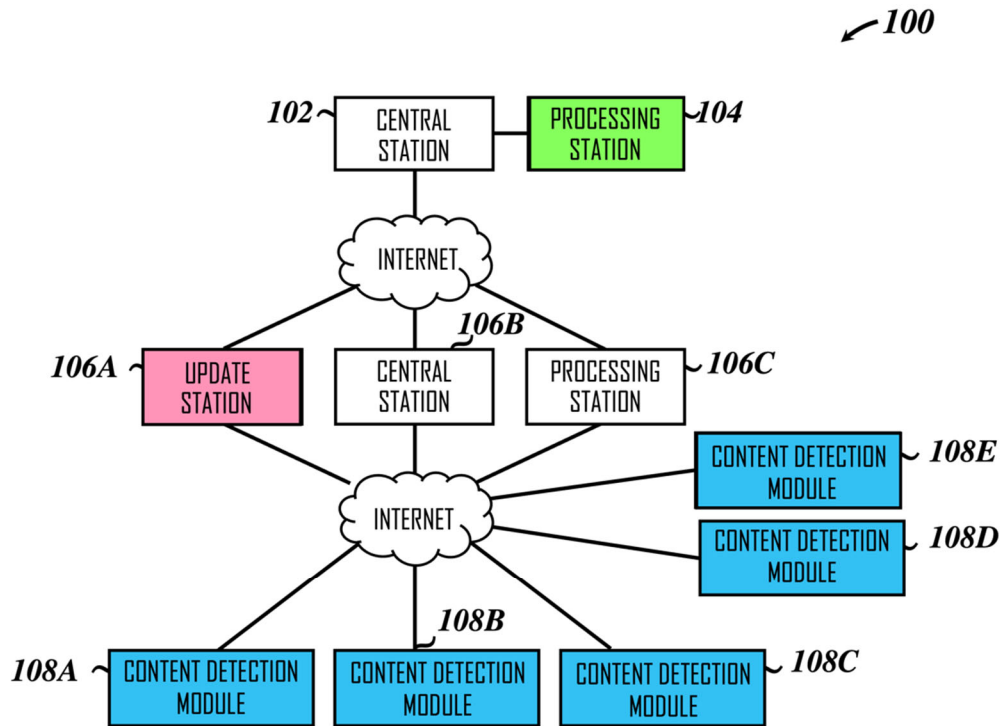


Fig. 1 (Redrawn and Annotated)

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

the like, e.g., any information that can be used in a content detection or screening process.” (*Id.*, 6:33-38; Wenke-Lee, ¶53.)

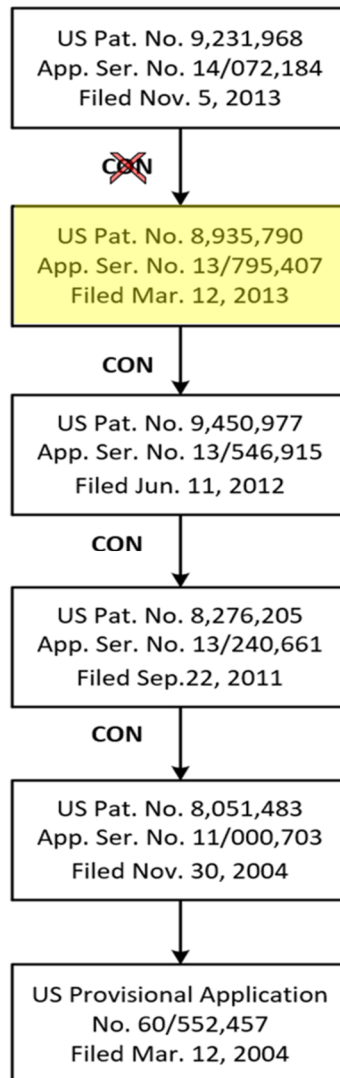
Once **processing station (104)** generates the content detection data, the **processing station (104)** sends the data to an **update station (106)**. (*Id.*, 6:47-50.) **Update station (106)** then sends the content detection data to the **CDM (108)**. (*See id.*, 6:55-58, 8:35-38.) The **CDM (108)**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

2. Prosecution History

As the sixth application in this family, the ’968 patent breezed through prosecution with only a single office action. The examiner rejected the applied-for claims as anticipated or obvious, except claims 4, 10, and 16. Applicant amended the independent claims to incorporate allowable dependent claims and any intervening dependent claims to obtain allowance. (*Id.*, 261-62, 220-24, 14-20.)

3. Priority

The ’968 patent issued from U.S. Patent Application No. 14/072,184, filed on November 5, 2013. The ’968 Patent claims priority to U.S. Patent Application No. 13/795,407, filed on March 12, 2013, as shown below. (Wenke-Lee, ¶¶55-56.)



But the challenged claims lack written description support in the ‘407 application because, *inter alia*, it does not disclose (1) “configuration data” and operations that implicate configuration data, or (2) “user identification” and operations that implicate user identification. Therefore, the challenged claims are entitled to a priority date of no earlier than November 5, 2013, the date the ’968 patent was filed. (Wenke-Lee, ¶¶57-58.)

“Configuration Data” (Limitations 1[d]/1[e][ii]/1[e][iii]; 4[b][i]/4[b][ii];

7[b][i]/7[b][ii]): The issued ’968 independent claims require two distinct types of data that define policies and are used for processing received network traffic to enforce those policies—“content detection” data and “configuration data.” While the claim term “content detection” data is described in several locations in the ’968 Patent (*see, e.g.*, ’968 patent, 6:33-41), “configuration data” and their associated operations are not disclosed anywhere in the ’968 Patent, the ’407 application, or in any other application in the ’968’s priority chain. This fact alone indicates that “configuration data” lacks written description support. (Wenke-Lee, ¶59.)

Searching the written description for something that might resemble “configuration data” fares no better. As noted, the claims require both “content detection data” and “configuration data” that define policies and are used for processing received network traffic by a network gateway device/content detection module. Yet the ’407 application discloses neither for configuration data. (Wenke-Lee, ¶60.)

Notably, the ’407 application’s only mention of “configuration” concerns the central station selecting update stations in a “round robin configuration”—an entirely different device than the claimed CDM that performs an entirely different operation. (*See* ’407 Application, 180, ¶37.) Indeed, every mention of the CDM processing network traffic describes processing the network traffic using *only*

“content detection data,” not “configuration data.” (*See id.*, 175-76, ¶¶29; 177, ¶¶32; 181-82, ¶¶39; *see also* 275, ¶¶28; 178, ¶¶33; 180, ¶¶36; 12, ¶¶37; 181, ¶¶38; 182, ¶¶40; 182, ¶¶41.) The written description never describes “configuration data”—either expressly or impliedly—as recited in the claims. (Wenke-Lee, ¶¶61.)

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

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

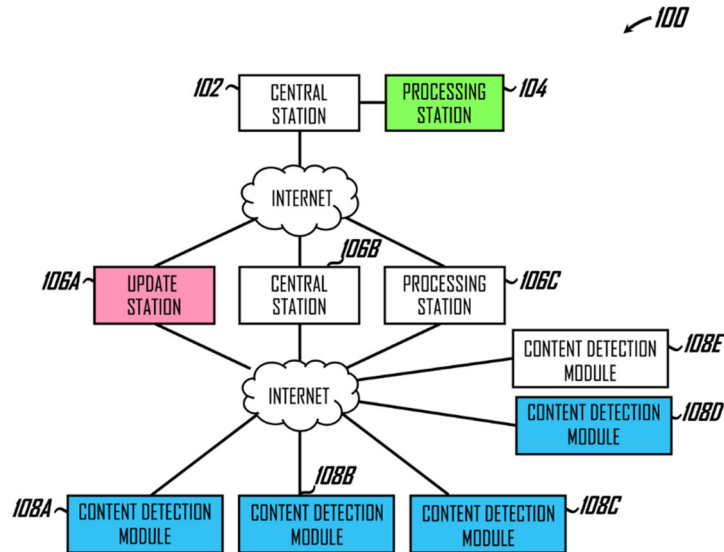


Fig.1 (Redrawn and Annotated)

Neither instance disclosed providing “user identification” to a content detection module/network gateway device, let alone that “user identification” is used in policy enforcement. (Wenke-Lee, ¶64.)

A different concept, “user authentication,” is mentioned in connection with policy enforcement. (*See id.*, 5:23-28.) But user identification and user authentication are technically distinct operations. User identification is often public, such as an email address or a username. (Wenke-Lee, ¶¶65-66.) Conversely, authentication often uses private information, such as a password. (*Id.*)

User identification is also often a passive, simpler process where a user’s identity (e.g., “Bob”) is taken at face value, and a policy enforcement based on user identification can be, for example, checking against a list of allow/block identities (e.g., allow “Bob”). (Wenke-Lee, ¶66.) This operation is often repeated multiple

times and offers no confidence in the user being who the user represents. (*Id.*) Conversely, user authentication is a more complicated, less frequent process that determines if a user is who they claim to be, often involving cryptographic operations between the parties to establish a chain of trust. (*Id.*) A POSITA would not have understood a security policy that was based on “user authentication” to be based on user identification. (*Id.*) Notably, by referencing “user identification” instead of “user authentication,” Patent Owner has impermissibly broadened the scope of its claims to require a much simpler and commonplace operation of user identification to allow user access to data, which was not described or even contemplated in the original application. (Wenke-Lee, ¶66.)

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

Given the above, the challenged claims introduce new matter and, consequently, cannot claim priority to any prior applications under 35 U.S.C. § 112.

Therefore, the challenged claims should be entitled to a priority date no earlier than November 5, 2013. (Wenke-Lee, ¶68.)

V. SECTION 325(D)

The Board should institute IPR notwithstanding its discretion under §325(d). *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8 (P.T.A.B. Dec. 15, 2017) (identifying six-factor test); *Advanced Bionics, LLC v. Med-El Electromedizinische Geräte GMBH*, IPR2019-01469, Paper 6 at 10 (Feb. 13, 2020) (precedential).

Factors (a) and (b): None of the references cited herein were a basis for rejection during examination. *RTI Surgical, Inc. v. LifeNet Health*, IPR2019-00573, Paper 20 at 27 (Aug. 12, 2019). FortiGate was cited in an IDS that included **250+ references**, and only half the reference was submitted. (See '968 Prosecution History, Nov. 5, 2013 IDS (FortiGate listed on page 6).) Fang—the published application of a '968 grandparent—was cited on an IDS that included over 60 other references. (See *id.*, May 29, 2014 IDS (Fang listed on page 1).) But there is no indication the examiner considered the '968 patent's priority issue or considered Fang prior art. Accordingly, the Petition is not cumulative of the original prosecution because it relies on different references and combinations than those before the Examiner and references that disclose all recited steps.

Factor (d): Since the Examiner did not consider Fang or FortiGate as noted above, there is no overlap in arguments made during prosecution and in the Petition, and the remaining references were neither cited to, nor considered by, the patent office.

Factor (c): While Fang and FortiGate were cited in IDSs, listing a reference in an IDS is not enough under § 325(d). *Bowtech, Inc. v. MCP IP, LLC*, IPR2019-00379, Paper 14 at 18 (P.T.A.B. July 3, 2019).

Factors (e) and (f): The combinations of Jordan-FortiGate, Lee-Magdych, Lee-Magdych-Judge, and Fang-FortiGate present new and distinct invalidity challenges that better reflect the state of the prior art than the references and combinations considered in the '968 Patent's prosecution. These grounds demonstrate fatal defects in the challenged claims of the '968 Patent that are not only new but also not cumulative over the prior art of record.

VI. LEVEL OF ORDINARY SKILL

A POSITA at the time of the alleged invention in November 2013 would have had at least a B.S. in computer science, computer engineering, or electrical engineering (or equivalent experience) and at least two years of practical experience with network and data security. The level of ordinary skill would be the same if the '968 patent's prior date was around 2004. (Wenke-Lee, ¶29.)

VII. CLAIM CONSTRUCTION

Petitioner does not believe any claim constructions are needed.

VIII. GROUNDS

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

Jordan: Jordan recognized the need for “a system or method capable of adequately protecting a network, such as a system or method that [was] *capable of dynamically adapting to new threats*.”² (*Id.*, 2:52-55.) With the U.S. Department of Defense’s support, Jordan addressed that exact problem, inventing ways “of dynamically adapting to new threats in real-time (e.g., preventing so-called ‘zero-day attacks’), or on an as-needed or on a ‘just-in-time’ (JTT) basis.” (*Id.*, 2:65-3:1; *see also, id.*, 1:11-20, 1:24-25; Wenke-Lee, ¶74.)

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

² Emphasis has been added by Petitioner unless indicated otherwise.

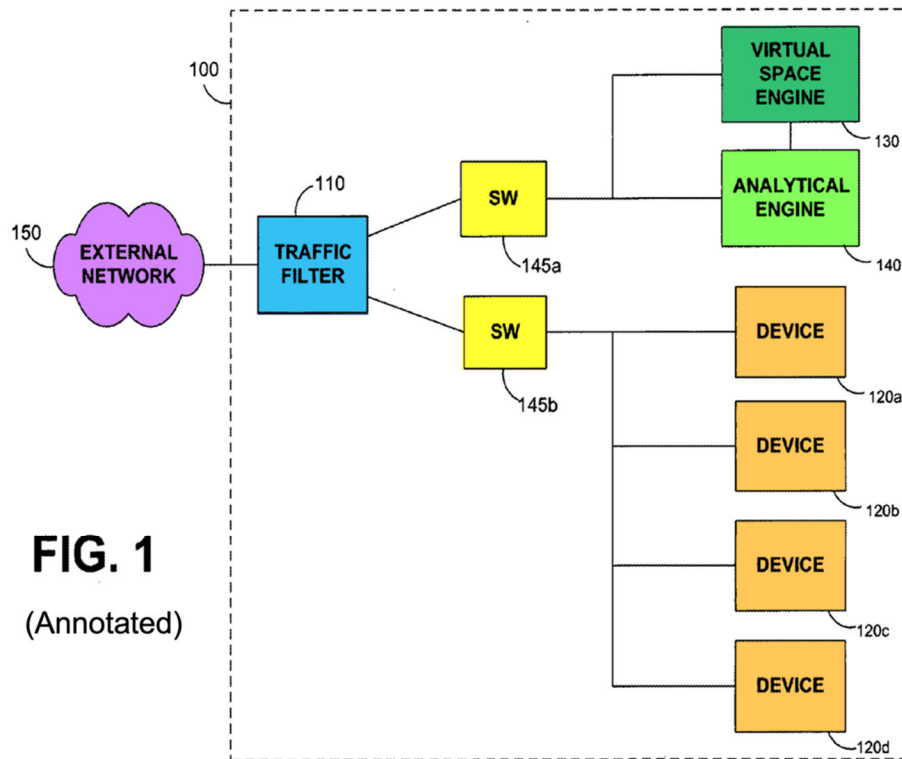


Figure 3 (below) depicted a process flow for handling traffic using this system. (*Id.*, 14:6-11.) The traffic filter (110) first received a communication (Step 302). (*Id.*) The traffic filter (110) then applied predefined rules (Step 304) to determine how to handle the communication. (*Id.*, 14:10-11, 15:19-22.) If the communication was normal, the traffic filter (110) passed the communication to its intended device (120(a)-(d)) (Step 308). (*Id.*, 14:19-24.) However, if an anomaly was detected, the traffic filter (110) passed the communication to the analytical engine (140) (Steps 306-310). (*Id.*, 14:27-36.) The analytical engine (140) then performed pattern recognition on the communication and, after doing so, generated

new detection rules or updated existing ones (Steps 310 and 312). (*Id.*, 14:36-36, 14:49-53.) Those rules would be sent to the traffic filter (110), which used them to improve anomaly detection. (*Id.*, 14:54-56; *see also, e.g., id.*, 8:33-9:64; Wenke-Lee, ¶76.)

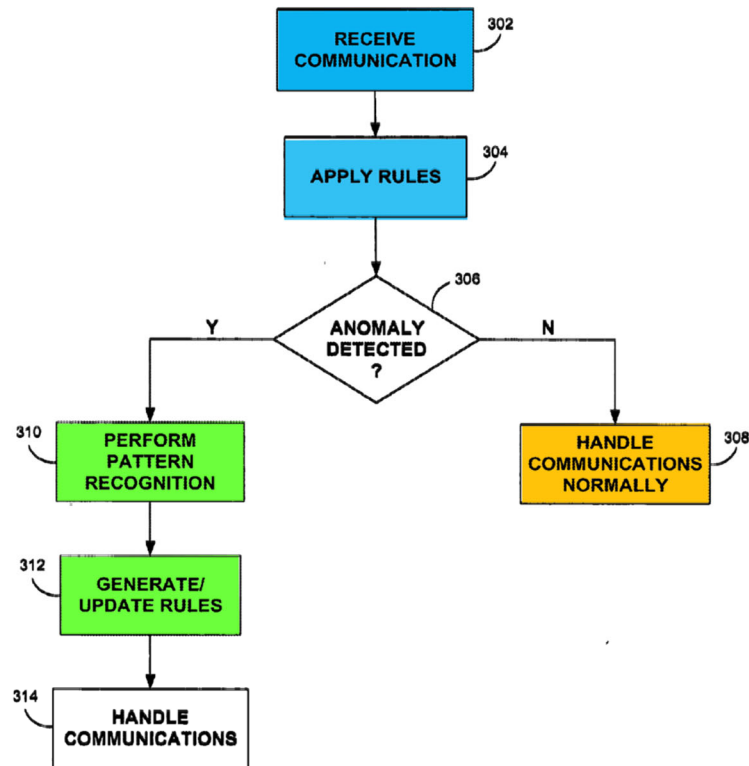


FIG. 3 (Annotated)

FortiGate³: FortiGate described a “FortiGate-3600” firewall unit for enterprise networks. (See, FortiGate, 13.) FortiGate disclosed an “AntiVirus Firewall” that included “services such as virus protection and content filtering” and “firewall [and] intrusion detection...” (FortiGate, 13.) According to FortiGate, the

³ FortiGate Manual was publicly available before the '968 patent. For example, it was identified as a reference during prosecution of U.S. Patent 9,716,644 (“Wei”), another Patent Owner patent. (Wenke-Lee, ¶78-86, & n.4; Ex. 1020.) Fortinet identified FortiGate on an April 27, 2015 IDS, and the copy available in the file history is Exhibit 1008. In the IDS, Fortinet identified the FortiGate’s date as “2003,” the same representation it made here. (See '968 Prosecution History, June 3, 2015 IDS (listed on 6th page of document).) The first half of FortiGate (submitted during prosecution of Wei) and the first half of FortiGate (submitted during prosecution of the '968 patent) are mirror images, except for Bates Nos. on one copy. (Compare Ex. 1008 with Ex. 1021.) Further, FortiGate shows a September 5, 2003 date, shows “© Copyright 2003 Fortinet Inc.,” states it is a publication, and includes contact information. (FortiGate, 2.) A copyright notice and IDS listing are sufficient to show a reference is prior art. *LG Elecs., Inc., v. Advanced Micro Devices, Inc.*, IPR2015-00329, Paper 13, 12-13 (P.T.A.B July 10, 2015).

FortiGate unit applied policies to filter traffic and protected network users from external threats, like viruses. (*See* FortiGate, 13-16.) Below is an image of a FortiGate unit. (FortiGate, 13; Wenke-Lee, ¶77.)



1. Motivation to combine

A POSITA would have been motivated to combined Jordan and FortiGate with a reasonable expectation of success. (Wenke-Lee, ¶87.)

Both Jordan and FortiGate concerned network and data security. Jordan disclosed systems for detecting and handling anomalous/malicious communications. (Jordan, 5:28-30, 5:36-39.) Jordan further disclosed generating or updating rules to improve malware detection. (Jordan, 6:37-46; *see also id.*, FIG 1, 5:21-54, 8:33-9:6, 9:33-50.) For example, Jordan described a system that dynamically adapted to new threats to limit attacks on a network. (Jordan, 2:52-3;5, 9:33-50.) Likewise, FortiGate concerned analyzing and filtering network traffic. (FortiGate, 13, 14, 16, 20.) For example, FortiGate described an intrusion detection system that analyzed content (e.g., network traffic) in “real-time” and downloaded new/updated virus definitions to identify suspicious traffic. (FortiGate, 13, 16, 20.) Thus, a POSITA looking to improve Jordan would have look to complementary references like FortiGate. (Wenke-Lee, ¶¶87-88.)

Element-specific motivations to combine are discussed below.

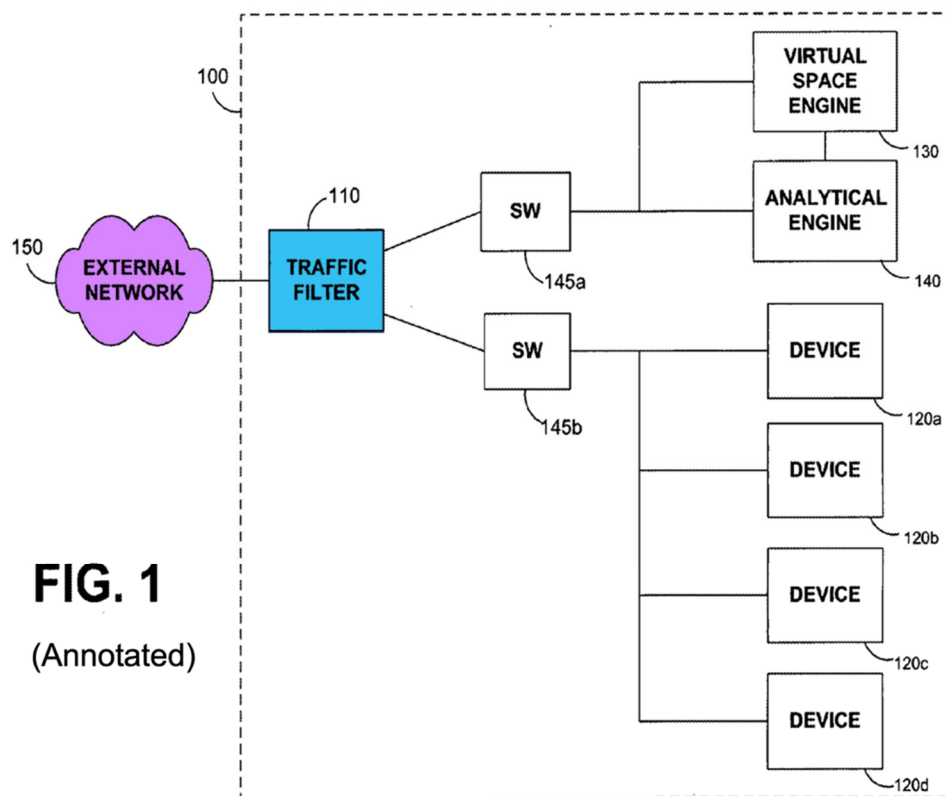
2. Claims 1/4/7

1[pre]: “A network gateway device comprising:”

Jordan disclosed a traffic filter (network gateway device). (Wenke-Lee, ¶¶90.)

Figure 1 (below) depicted traffic filter (110), which could be “used as a perimeter device to *control all communications crossing a network boundary.*”

(Jordan, 5:28-30, Fig. 1.; Wenke-Lee, ¶¶90-91.)



The '968 patent described the network gateway device in similar fashion. Like the traffic filter, the gateway device sits between an external network and a

local network. (*See* '968 Patent, 5:20-23.) Indeed, FortiGate—written by Fortinet—described a “gateway” as simply something “that links different networks.” (FortiGate, 297.) Accordingly, a POSITA would have understood that traffic filter was a network gateway device because both were perimeter devices that linked networks. (Wenke-Lee, ¶92.)

4[pre]: “A method comprising:”

Jordan disclosed a “method for defending a computer network.” (*Id.*, 2:63-65; Wenke-Lee, ¶93.)

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

1[a]: “at least one processor;”

1[b]: “at least one memory device;”

Jordan disclosed “a processor-readable medium comprising code representing instructions to cause a processor to” perform functions of the claimed invention. (Jordan, 3:6-8; *see also id.*, 3:25-30, 25:29-32, FIG. 11.) A POSITA would have understood that a “processor-readable medium” was a “non-transitory computer-readable medium” because a processor is the component of a computer that reads memory. (Wenke-Lee, ¶¶94-95; *see, e.g.*, '968 patent, 10:49-51.) A POSITA would have also understood that processor-readable medium referred to a

memory device because the processor-readable medium stored code or instructions.
(Wenke-Lee, ¶95.)

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

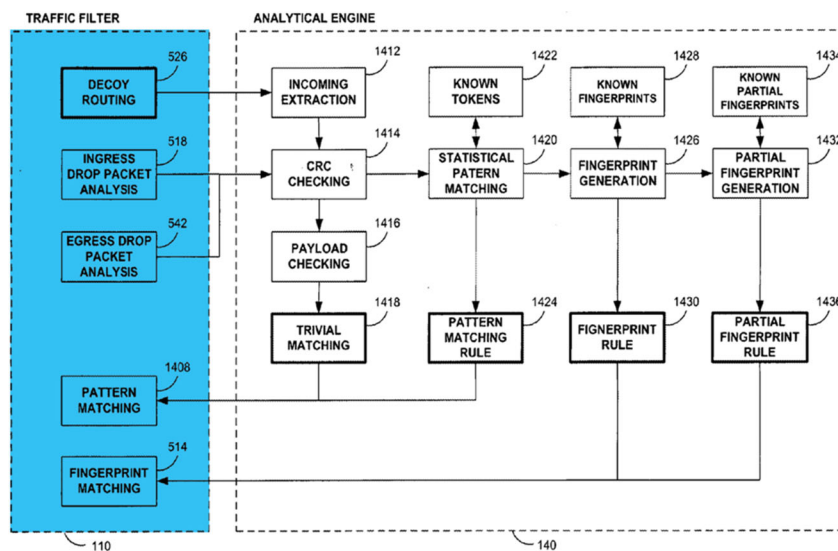
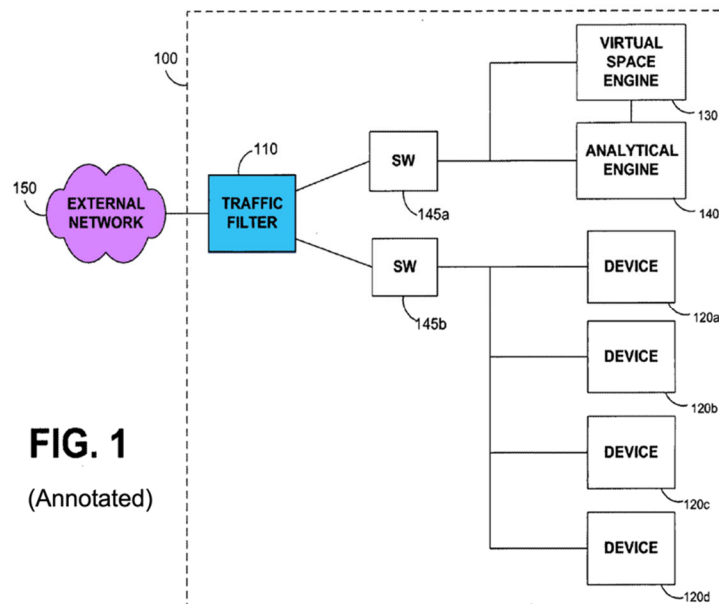


FIG. 14 (Annotated)

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

Jordan disclosed or strongly suggested that the traffic filter had a network interface device because it received and transmitted network communications.
(Wenke-Lee, ¶97.)

Figure 1 (below) showed that the traffic filter received traffic from an external network and routed it accordingly. (Jordan, 3:14-40, 14:19-36, 11:1-24, Figs. 1, 3, 5, 11; Wenke-Lee, ¶¶97-98.)



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

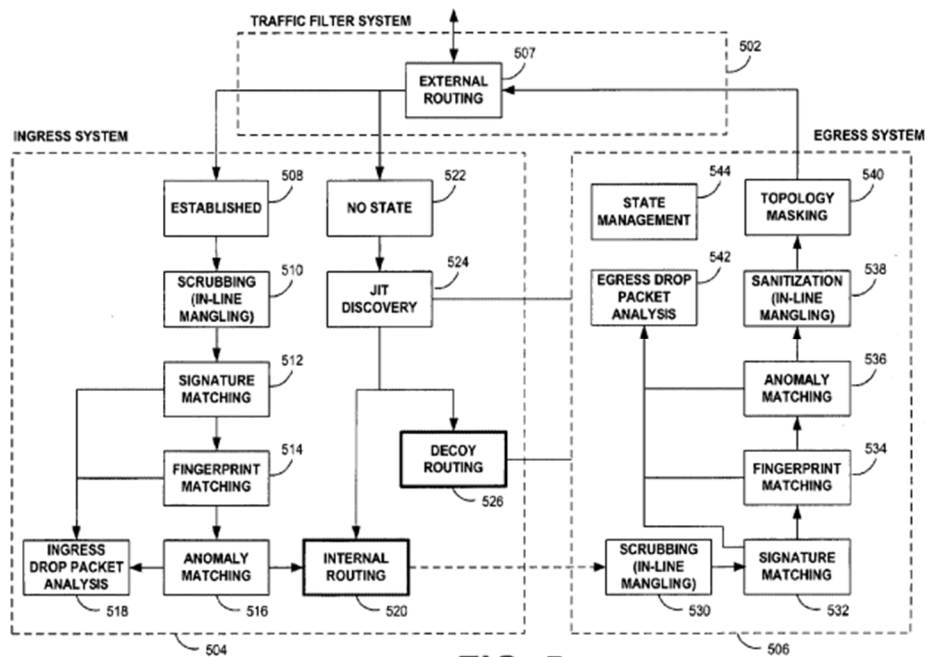


FIG. 5

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

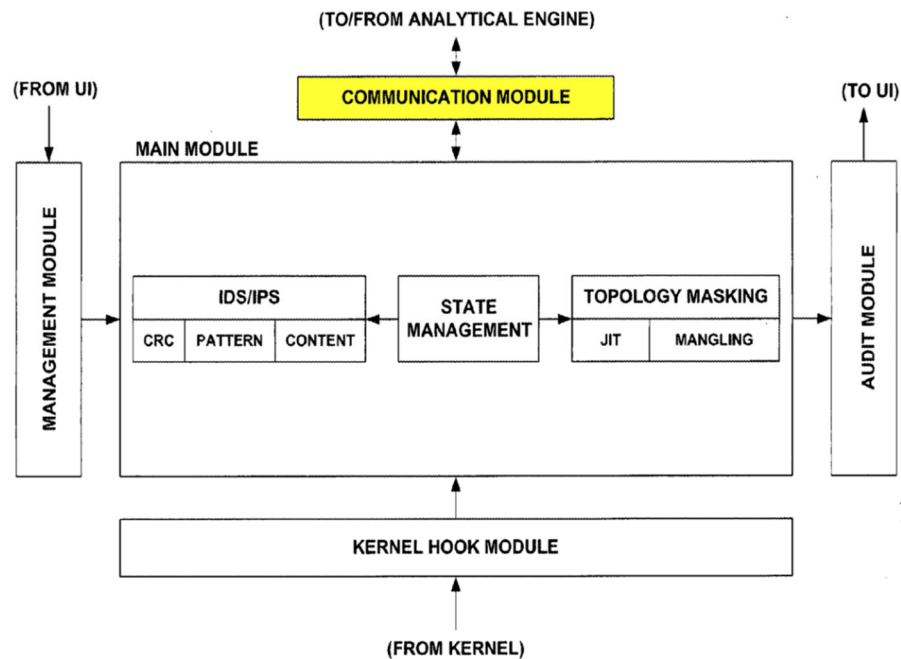


FIG. 11 (Annotated)

Accordingly, Jordan clearly disclosed or suggested a network interface device. Indeed, network interface devices were ubiquitous before the '968 patent and would have been installed in any network-enabled device. (Wenke-Lee, ¶101.)

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

Jordan disclosed that the traffic filter stored content detection data (e.g., virus lists or threat lists) on memory and suggested configuration data. If Jordan did not disclose configuration data, a POSITA would have found it obvious to add it based on FortiGate’s disclosures. (Wenke-Lee, ¶¶102-103.)

“content detection [data]”: The '968 patent broadly describes “content detection data”:

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

(’968, 6:33-38; Wenke-Lee, ¶105.)

Jordan disclosed a wide variety of content detection data, including virus and threat lists. Figure 11 (below) showed that the traffic filter contained modules for intrusion detection systems/intrusion protection systems. (*Id.*, 25:61-63.) Those modules included “pattern and content checking” modules, which [could] search certain patterns and/or contents ... of a communication data stream.” (*Id.*, 25:67-26:3; *see also* Wenke-Lee, ¶106.)

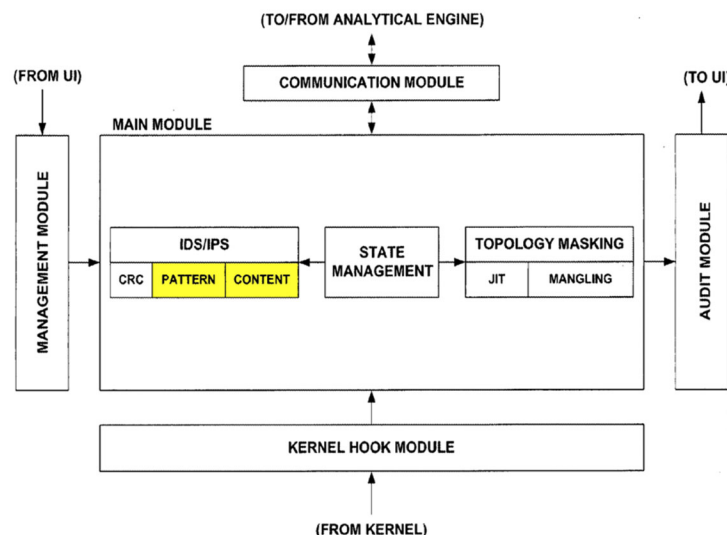


FIG. 11 (Annotated)

Jordan explained that the traffic filter used various data when processing traffic for suspect content. For example, the data included “*lists of patterns (e.g., virus lists, threat lists, etc.)*.” (*Id.*, 16:42-44; *see also, e.g.*, 17:39-41 (“*[T]raffic filter can use other rules, such as predetermined rules (e.g., virus lists)*, or other preexisting rules.”).) The data could also include signatures, fingerprints, and payloads for matching. (*See id.*, 18:20-26 (“[I]ngress system 504 [of traffic filter] can perform additional functions on the incoming communication, such as *signature matching...and fingerprint matching*.”), 17:36-38 (“More specifically, a ‘trivial rule’ is a rule configured to handle an attack that always has the same payload.”); 5:56-57.) (Wenke-Lee, ¶106.)

This data was stored in the traffic filter. (Jordan, 9:35-40, 14:9-12, 14:54-56, 15:29-31.) Figure 14 (below) showed that rules were sent from the analytical engine to the traffic filter, which applied the rules to recognize a pattern. (*See e.g., id.*, 14:31-33.) A POSITA would have understood that the content detection data (e.g., a virus signature) was stored on the traffic filter because it would need local access to that data to process traffic. (Wenke-Lee, ¶107.)

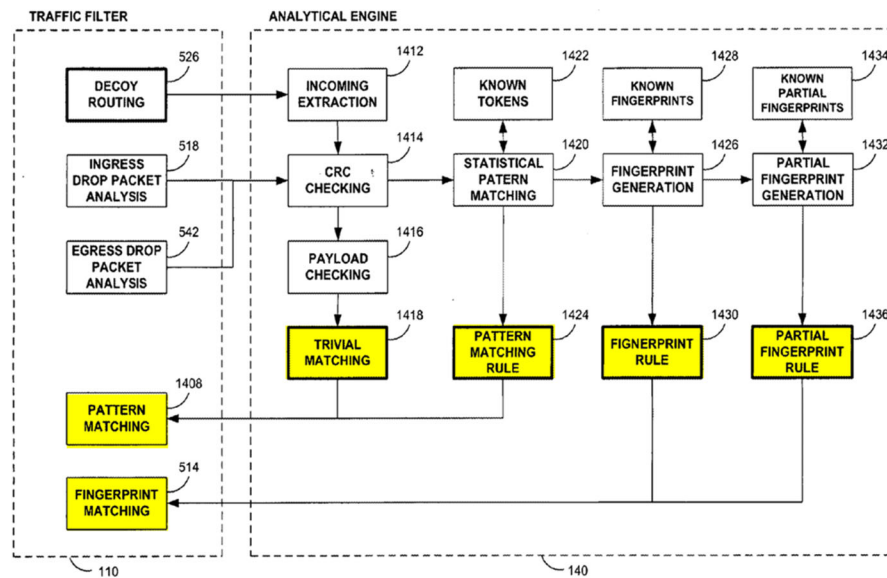


FIG. 14 (Annotated)

“configuration data”: Jordan explained that the traffic filter had a management module (shown in Figure 5, below). The management module provided an interface between the traffic filter and a system administrator. (*Id.*, 25:50-51.) Using the module, an administrator could **“manage rules** of the traffic filter 110 [or] **set system parameters.**” (*Id.*, 25:52-54; Wenke-Lee, ¶¶108-09.)

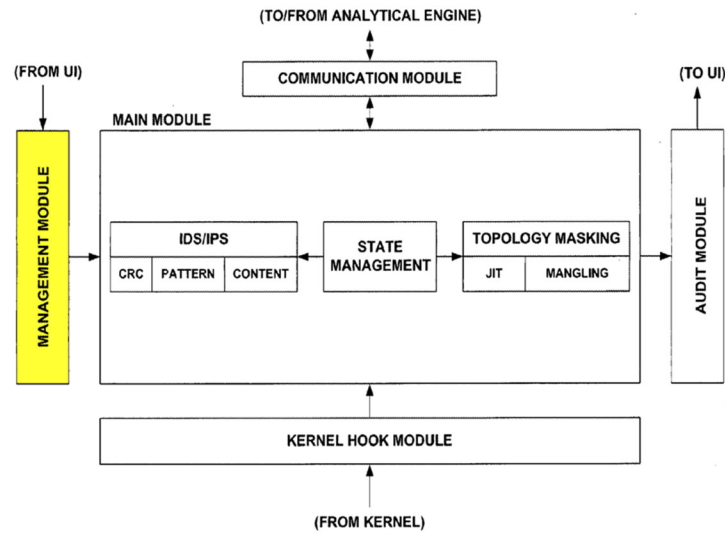


FIG. 11 (Annotated)

Thus, Jordan recognized its system could be configured, but simply did not explain what those configurations could be. FortiGate did. (Wenke-Lee, ¶110.)

FortiGate described a firewall configuration (below). An administrator could configure various things, like “Service,” “NAT,” “Traffic Shaping,” “Authentication,” and “Antivirus & Web Filter.” (FortiGate, 34-35; *see also id.*, 16; Wenke-Lee, ¶111.)

Table 4: Factory default firewall configuration

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

	☐ Authentication	Authentication is not selected. Users do not have to authenticate with the firewall before connecting to their destination address. You can configure user groups and select this option to require users to authenticate with the firewall before they can connect through the firewall.
	☐ Antivirus & Web Filter	Antivirus & Web Filter is not selected. This policy does not include a content profile that applies antivirus protection, web content filtering, or email filtering to content traffic processed by this policy. You can select this option and select a content profile to apply different levels of content protection to traffic processed by this policy.
	☐ Log Traffic	Log Traffic is not selected. This policy does not record messages to the traffic log for the traffic processed by this policy. You can configure FortiGate logging and select Log Traffic to record all connections through the firewall that are accepted by this policy.

The figure below illustrates policy configurations through the user interface. (*Id.*, 173; Wenke-Lee, ¶111.)

Figure 5: Adding a NAT/Route policy

The screenshot shows the 'Policy' configuration window in FortiGate. The 'Source' is set to 'Internal_All', 'Destination' to 'External_All', 'Schedule' to 'Always', 'Service' to 'ANY', and 'Action' to 'ACCEPT'. The 'NAT' checkbox is checked, with 'Dynamic IP Pool' selected. The 'Traffic Shaping' checkbox is checked, with 'Guaranteed Bandwidth' set to 100 (KBytes/s), 'Maximum Bandwidth' set to 100 (KBytes/s), and 'Traffic Priority' set to 'High'. The 'Authentication' checkbox is checked, with 'User_Group_1' selected. The 'Anti-Virus & Web filter' checkbox is checked, with 'Content Profile' set to 'Scan'. The 'Log Traffic' checkbox is unchecked. The 'Comments' field contains the text: 'Policy: Traffic Shaping, Authentication, and Virus Scanning'. The 'OK' and 'Cancel' buttons are at the bottom.

Source	Internal_All
Destination	External_All
Schedule	Always
Service	ANY
Action	ACCEPT
☒ NAT	
Dynamic IP Pool	☒
Fixed Port	☐
☒ Traffic Shaping	
Guaranteed Bandwidth	100 (KBytes/s)
Maximum Bandwidth	100 (KBytes/s)
Traffic Priority	High
☒ Authentication	
User_Group_1	☐
☒ Anti-Virus & Web filter	
Content Profile	Scan
☐ Log Traffic	
Comments: maximum 53 characters	
Policy: Traffic Shaping, Authentication, and Virus Scanning	
OK	Cancel

FortiGate disclosed other configurations. (See, e.g., *id.*, 14 (blocking files of specified file types), 14 (blocking all or some of a webpage).) (Wenke-Lee, ¶112.)

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

data. These configurations would naturally be incorporated into Jordan's traffic filter, which processed traffic like the FortiGate firewall. (Wenke-Lee, ¶113.)

"stored on the at least one memory device": Jordan did not explicitly disclose this element, Jordan suggested it and a POSITA would have found it obvious. (Wenke-Lee, ¶¶114-15.) Jordan's traffic filter applied rules for processing traffic. (Jordan, 14:9-11, 16:42-47, 14:30-34; *see id.*, 5:21-54.) A POSITA would have found it obvious to store the content detection and configuration data on the traffic filter's memory because the traffic filter would need to retrieve the data when processing traffic or after it was restarted. (Wenke-Lee, ¶¶114-15.)

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

Jordan disclosed or strongly suggested an instruction set stored in the traffic filter's memory and executable by a processor. (*See* 7[pre] and 1[a]-1[b]; *see, e.g., id.*, 3:25-30; Wenke-Lee, ¶116.)

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

Jordan disclosed that the traffic filter received incoming communications. (Wenke-Lee, ¶117.)

Jordan explained that the traffic filter was ***"configured to receive communications*** from the external network 150...." (*Id.*, 8:39-42; *see also, e.g.,* 5:23-25.) Figure 3 (below) also showed that the traffic filter received

communications (Step 302). (*Id.*, 14:9-11; *see also, e.g.*, 15:50-53.) A POSITA would have understood that the communications were network traffic because they were data transmitted over a network. (Wenke-Lee, ¶¶117-119.)

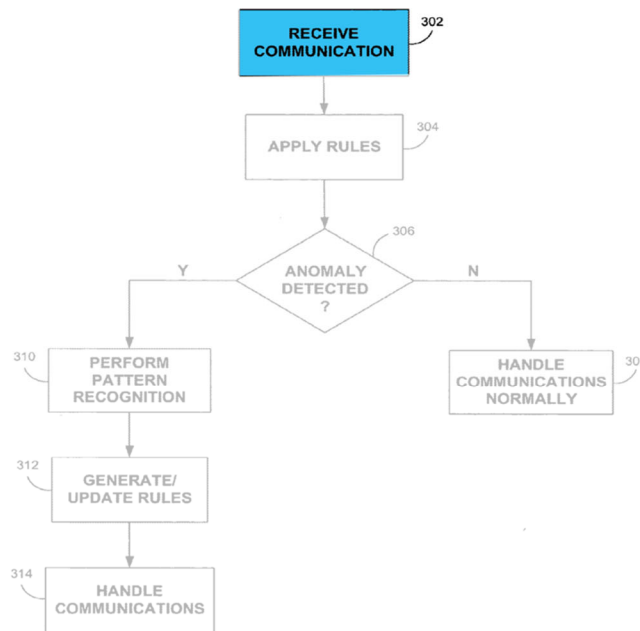


FIG. 3 (Annotated)

Jordan disclosed or suggested that the traffic filter had a network interface device for receiving communications. (*See* Element 1[c].) Thus, in the above disclosures, a POSITA would have understood that those communications were received over the network interface. (*See, e.g., id.*, 15:50-53; Wenke-Lee, ¶120.)

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

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

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

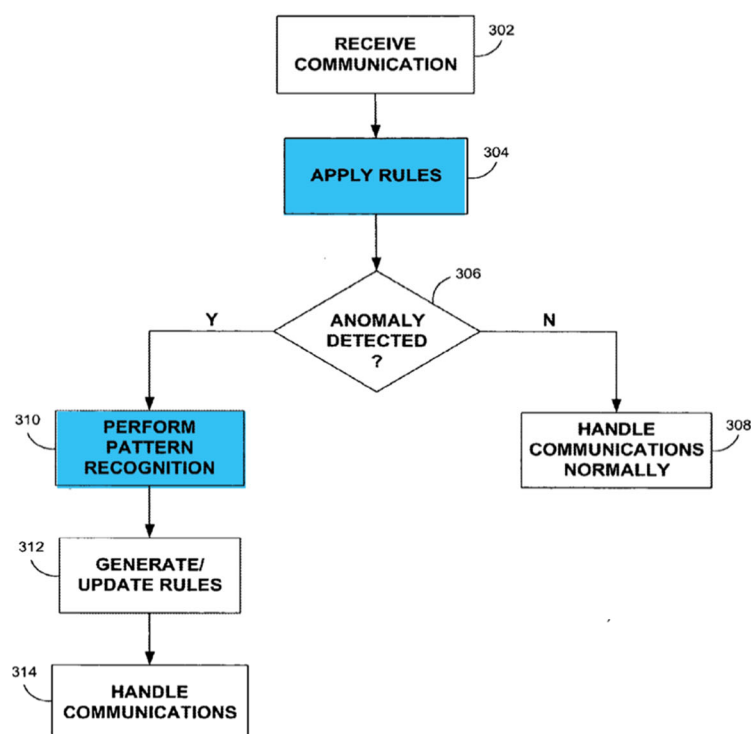


FIG. 3 (Annotated)

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

Similarly, Figure 5 (below) depicted a process flow for handling incoming communications (Ingress System (504)) of the traffic filter. (*Id.*, 17:50-52.) Steps 512-516 depicted various operations that used the content detection data (e.g., signatures and fingerprints). (*Id.*, 18:20-29; *see also, e.g., id.*, Fig. 14; Wenke-Lee, ¶124-25.)

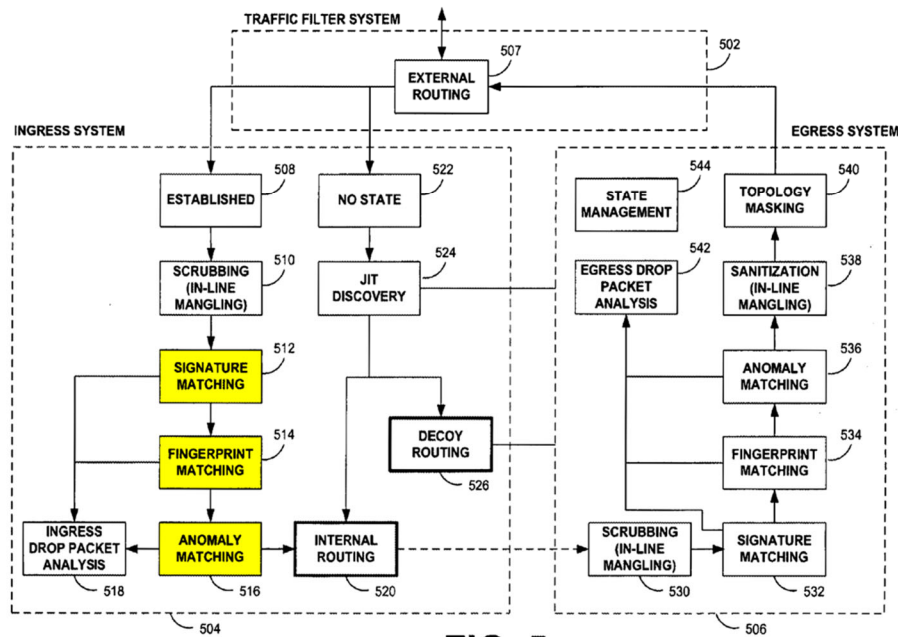


FIG. 5
(Annotated)

For configuration data, FortiGate disclosed processing received traffic according to firewall policies defined by such data. An administrator could, for example, configure “Traffic Shaping” or “Authentication.” (See Element 1[d]; FortiGate, 16, 34-35.) The exemplary configuration below showed how the FortiGate unit received and processed traffic according to the configuration data. (Wenke-Lee, ¶126.)

Table 4: Factory default firewall configuration

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

(Annotated)

“Traffic Shaping,” for example, “control[led] the maximum or minimum amount of bandwidth available to traffic processed by the policy.” (Wenke-Lee, ¶127-28.)

A POSITA would find it obvious the content detection data and configuration data were stored on the traffic filter’s memory. (Element 1[d]; Wenke-Lee, ¶128.)

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

Jordan disclosed that the traffic filter enforced policies defined by content detection (e.g., virus lists), and FortiGate disclosed enforcing policies defined by configuration data (e.g., Traffic Shaping, Authentication). Jordan also disclosed that the traffic filter included a policy that defined a suspicious category of network traffic (e.g., communications with uncommon attributes). (Wenke-Lee, ¶129.).

“to enforce policies defined at least in part by the content detection and configuration data”: As discussed for element 1[e][ii], Jordan disclosed that the traffic filter applied rules (enforce policies) to control network traffic. (See Element 1[e][ii]; *see also, e.g.*, Jordan, 17:39-41 8:52-54, 9:35-38, 14:9-11, 14:30-33, 31:10-11.) Figure 3 (below) depicted this. (Wenke-Lee, ¶130.)

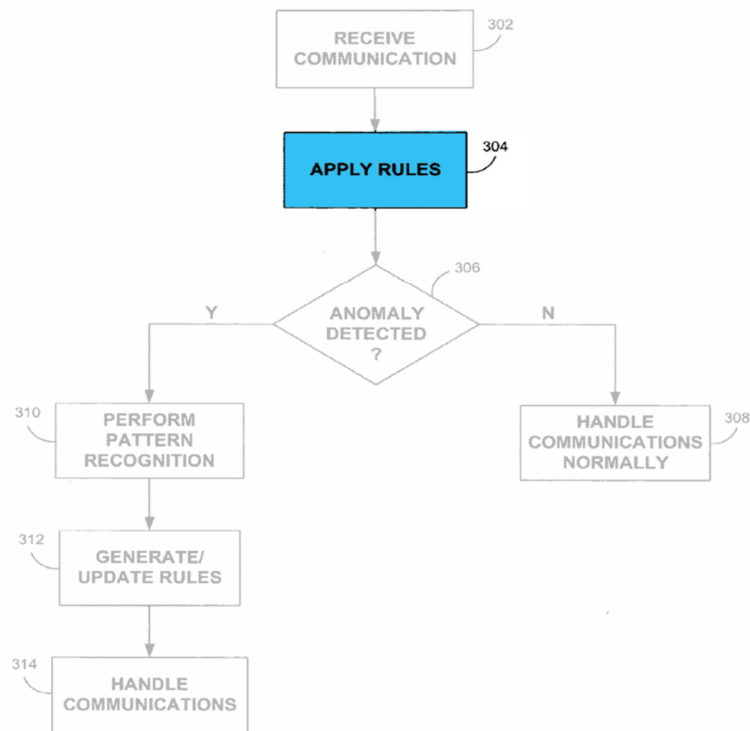


FIG. 3 (Annotated)

The content detection data (e.g., virus lists, pattern matching) and configuration data (e.g., Traffic Shaping, Authentication) were defined in rules/policies. (See Element 1[e][ii].) For example, the traffic filter applied rules defined by content detection data: “traffic filter 110 [could] apply [rules predetermined by analytical engine 140] *to recognize a pattern within the received communications.*” (Jordan, 14:31-33; see, e.g., *id.*, 16:42-44 (“[S]ystems [could] perform malicious detection by relying on continuously updated lists of patterns (e.g., virus lists, threat lists, etc.)”), 17:39-41; Wenke-Lee, ¶131.)

including at least one policy that defines a suspicious category of network

traffic: Jordan disclosed a rule (policy) that defined an anomalous communication. For context, Jordan explained that “anomalous communications” were “communications having attributes that [were] not common from the perspective of the traffic filter or the network of the traffic filter.” (Jordan, 5:41-44.) Jordan also disclosed that malicious communications could be considered anomalous. (*Id.*, 5:55-56; Wenke-Lee, ¶132.)

These categories of traffic were defined in rules. (Jordan, 3:28-30 (“determine if the received network communication *include[d] an anomaly according to a predetermined rule set....*”); *see also, e.g., id.*, 6:40-43, 10:21-23 (“If the traffic filter 110 deem[ed] a communication anomalous and the structure of that anomalous communication form[ed] the basis of a filtering rule....”).) Anomalous communications could be ones that requested a service that did not exist or were unrecognizable by the traffic filter 110. (*See* Jordan, 15:10-28.) A rule could also define as “anomalous” a user logging in both remotely and from their desktop. (*Id.*, 17:5-8; Wenke-Lee, ¶132-33.)

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

⁴ “On a data network” applies to only claims 4 and 7.

Jordan disclosed that the traffic filter applied rules to determine whether to allow network traffic to pass. (Wenke-Lee, ¶134.)

Jordan disclosed that “traffic filter 110 [] *implement[ed] even recent rules* ...and consequently the *traffic filter 110 [] prevent[ed] [] malicious payloads from being delivered* to one or more devices 120 within the network.” (Jordan, 9:40-45; *see also, e.g., id.*, 21:33-37 (“malicious communications can be used to generate and/or *update a filtering rule that can be used by the traffic filter 110* to improve the filtering capability of the network system 100 generally, and *to prevent similar attacks in the future*”), 6:40-44.) For example, if signature, fingerprint, or anomaly matching failed, an analysis was performed to “determine if at least a portion of the incoming communication...*should be dropped,*” as shown by Step 518. (*Id.*, 18:35-38, Fig. 5; Wenke-Lee, ¶135.)

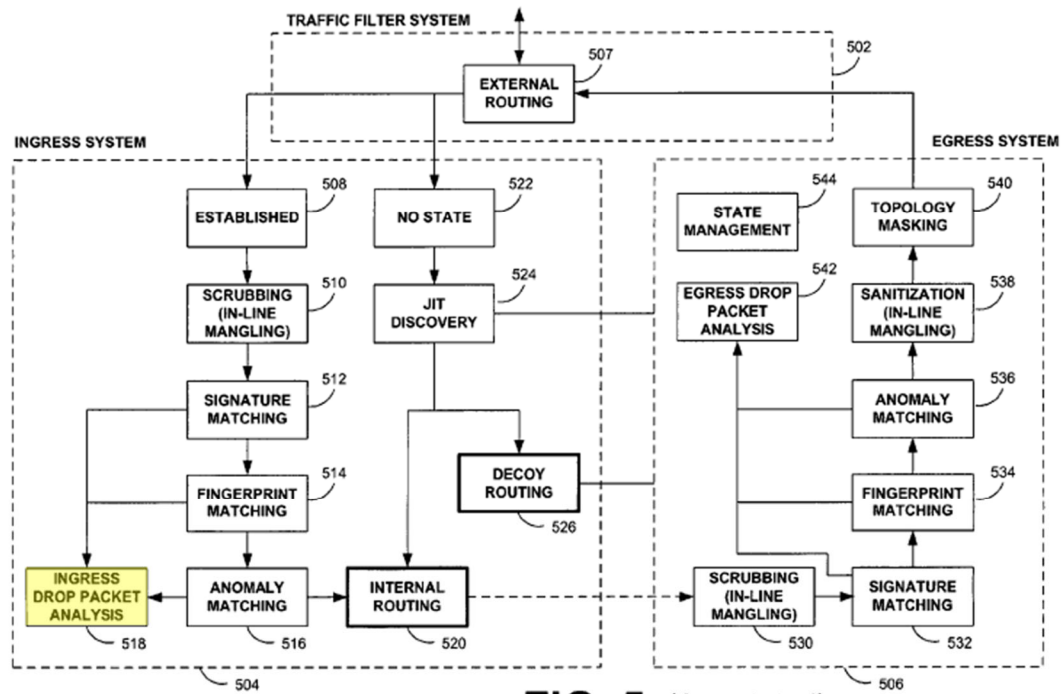
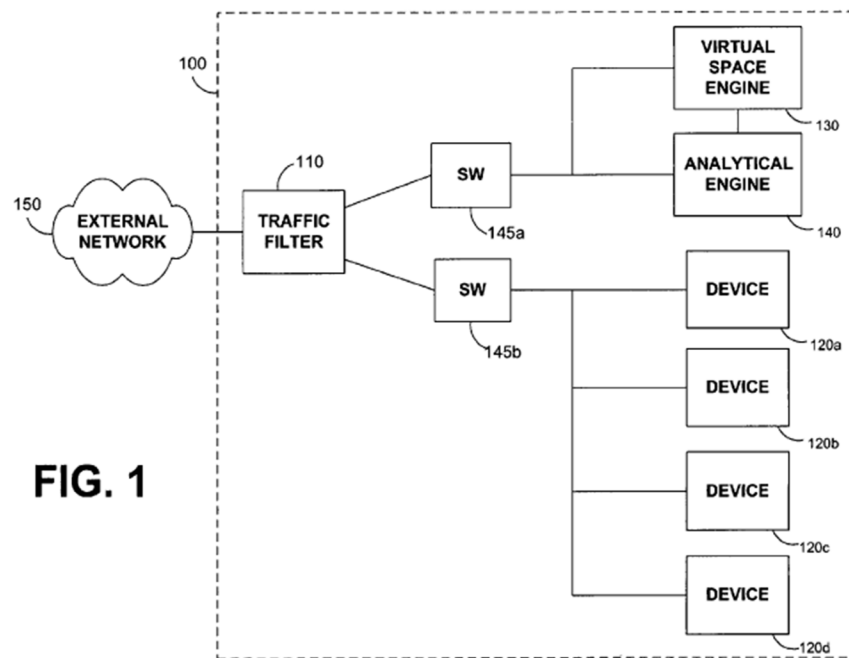


FIG. 5 (Annotated)

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



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

Jordan disclosed or suggested user identification, content identification, source verification, and destination verification. If Jordan did not disclose one or more of these, FortiGate did. (Wenke-Lee, ¶137.)

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

If Jordan did not disclose user identification, FortiGate did. (Wenke-Lee, ¶139.)

As one example, FortiGate described filtering email based on the sender appearing on an Email block list. (FortiGate, 15.) A POSITA would understand that email lists could include user identification (e.g., usernames). (Wenke-Lee, ¶¶140-41.)

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

content identification: Jordan disclosed that rule enforcement by the traffic filter included an intrusion detection system module that checked the contents of data streams (e.g., traffic). (Jordan, 25:61-26:3 (“The IDS/IPS module can also include pattern and *content checking modules, which can search certain patterns and/or contents, respectively, of a communication data stream.*”).) Similarly, the ’968 patent described that content scanning included “detecting undesirable content, such as ... porno website, etc.” (’968 Patent, 5:23-31.) A POSITA would have found it obvious that the content searching disclosed in Jordan would include searching for malicious or inappropriate content because content filters commonly checked for such things. (Wenke-Lee, ¶142.)

Further, FortiGate disclosed scanning email contents to determine if they contained a banned word or phrase. (FortiGate, 15 (“[i]f an *email is found to contain a word or phrase in the banned word list*, the FortiGate adds an Email tag

to subject line of the email. Receivers can then use their mail client software to filter messages based on the Email tag.”), 281.) FortiGate disclosed a similar feature for pornography. (*Id.*, 277; Wenke-Lee, ¶143.)

source and destination verification:

Jordan disclosed at least source verification because it described rules about “receipt of communications from *certain network sources*...” (Jordan, 6:51-55.) Likewise, a POSITA would have it obvious that Jordan’s rules could include destination verification since source and destination verification were commonly used together. (Wenke-Lee, ¶144.)

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

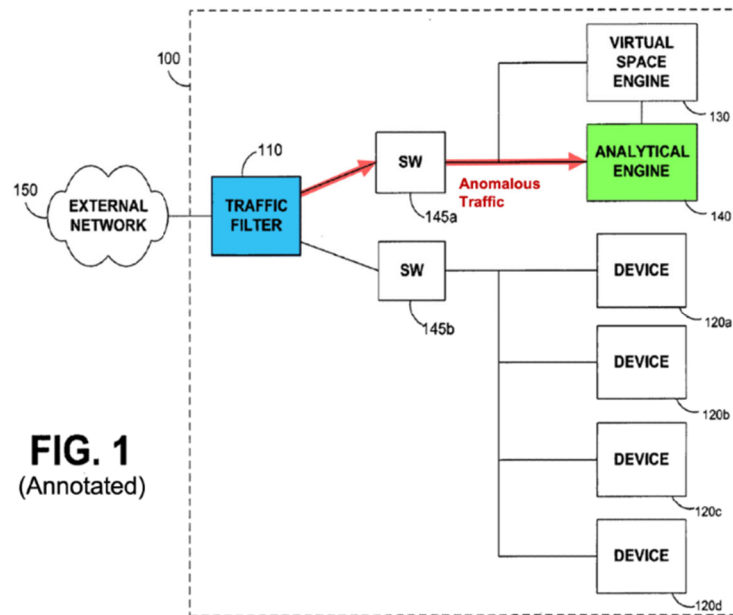
Table 4: Factory default firewall configuration (Annotated)

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

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

Jordan disclosed that the traffic filter determined if traffic violated a policy defining anomalous traffic and, if so, it forwarded the traffic to an analytical engine that analyzed the traffic to determine if it needed to generate or update rules. (Wenke-Lee, ¶146.)

As discussed for element 1[e][iii], Jordan disclosed a policy that defined anomalous communications, such as a communication that requested a non-existent service. (Jordan, 6:16-18; *see also id.*, 9:7-11.) If a communication violated the policy, the traffic filter routed the communication to analytical engine (140). (Jordan, 15:19-24 (“if a received communication violate[d] a predefined rule....the traffic filter 110 can consider the communication to be anomalous. Such anomalous communications can be handled by...the analytical engine 140.”), 5:58-63 (“[I]f the traffic filter determine[d] that a payload of a network communication contain[ed] malicious data (e.g., a virus, worm, etc.), then the anomalous network communication can be routed to the analytical engine by the traffic filter....”), 9:51-53, 11:6-9, 14:16-48.) This is illustrated in Figure 1 (below). (Wenke-Lee, ¶147.)



Jordan also showed this process in Figure 3 (annotated below). If an anomalous communication was detected (Step 306), pattern recognition was performed by the traffic filter or analytical engine depending on if rules existed. (Jordan, 14:27-30.) If no rules existed, a copy of the communication could be sent to the analytical engine. (*Id.*, 14:33-36.) The analytical engine then analyzed the communication for malware and performed pattern recognition on it. (*Id.*, 14:37-44; *see also id.*, 5:58-6:2, 6:16-28, 6:37-46, 6:47-55, 9:35-42, 9:51-64, 16:19-23; Wenke-Lee, ¶¶148-49.)

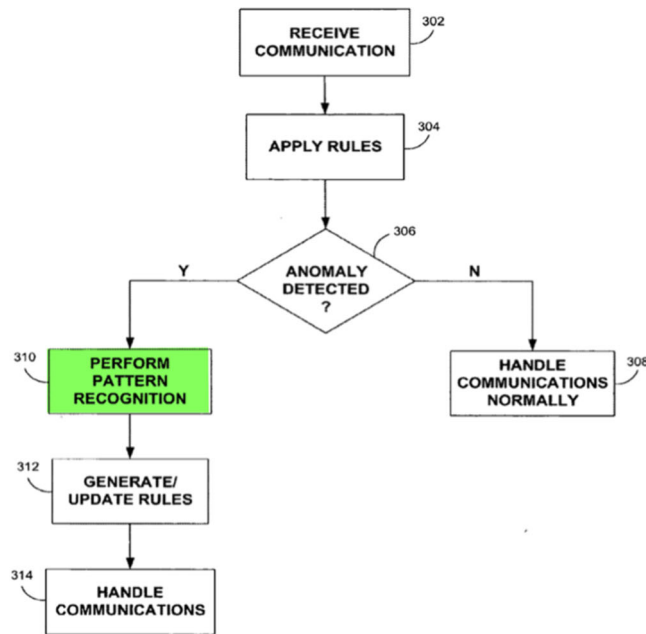


FIG. 3 (Annotated)

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

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

Jordan disclosed that the analytical engine generated or updated rules if the anomalous communication (e.g., traffic) was malicious. (Jordan, 6:37-44 (stating the analytical engine “analyze[d] a received communication, and determine[d] if rules should be generated or updated [for use by the traffic filter (110)] to prevent

any harm from the received communication[,] ... [such as] generat[ing] rules that prevent additional, similar anomalous communications from being received by a network associated with the anomalous communication.”); *see also id.*, 5:58-63.) The traffic filter updating/generating new rules is shown in Figure 3 (below). (Wenke-Lee, ¶151-52.)

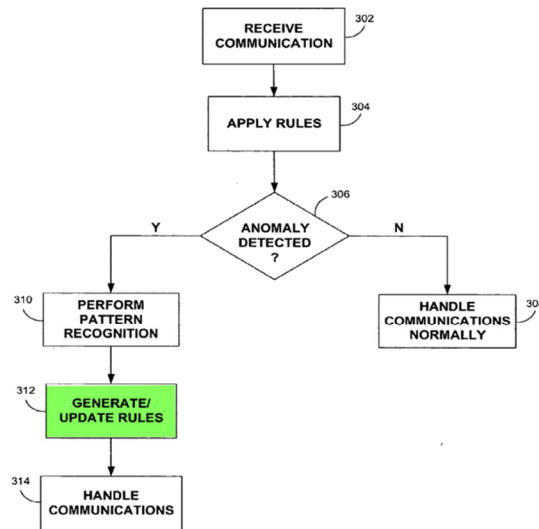


FIG. 3 (Annotated)

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

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

Jordan disclosed that the traffic filter received new/updated rules via a switch.

If Jordan’s switch does not constitute an “update station,” a POSITA would have found it obvious to incorporate FortiGate’s update station into Jordan. (Wenke-Lee, ¶154.)

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

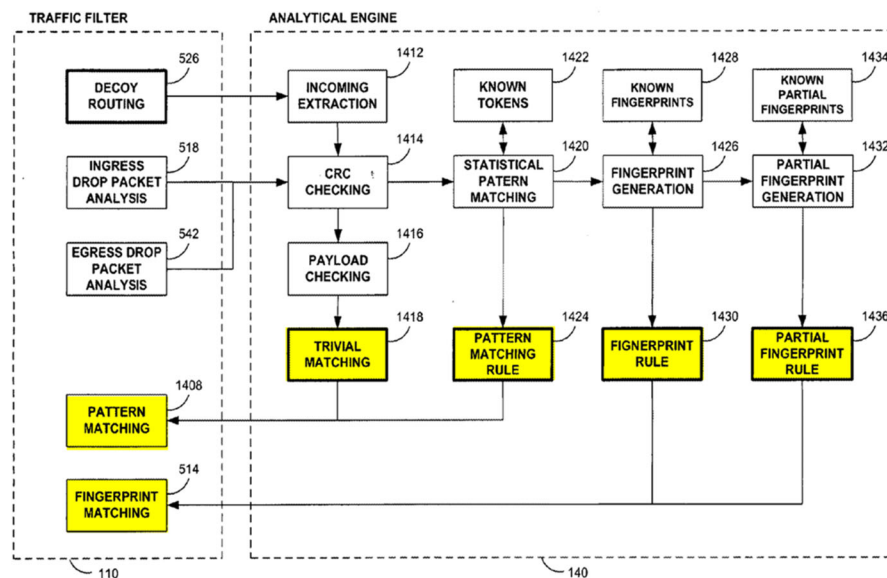
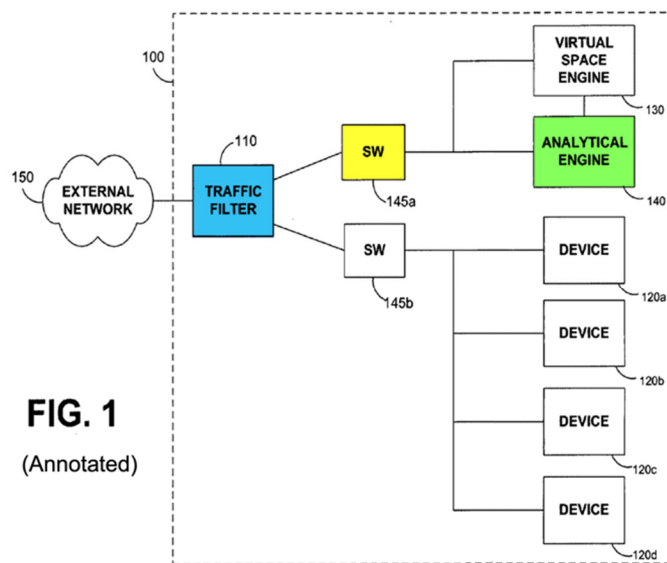


FIG. 14 (Annotated)

The rules contained additional content detection data (e.g., virus lists). (Jordan, 9:38-40, 14:54-56; Wenke-Lee, ¶155.)

The additional content detection data went indirectly from the analytical engine to the traffic filter. Figure 1 (below) showed that a switch (145a) intermediated traffic between analytical engine (140) and the traffic filter (110). Accordingly, the traffic filter received the data indirectly from the analytical engine because it went through the switch. (Wenke-Lee, ¶¶154-56.)



If Jordan’s traffic filter is not an “update station,” a POSITA would have found it obvious to incorporate that feature from FortiGate. FortiGate disclosed a FortiResponse Distribution Center (FDC) that received threats from Fortinet customer’s and updated antivirus definitions. (FortiGate, 335.) The FDC pushed updates to update stations called FortiResponse Distribution Servers (FDSs). (FortiGate, 336.) The FDSs (update stations) pushed updates to FortiGate firewalls. (*Id.*) Likewise, the ’968 patent describes update stations, such as a computer or

server, that receives content detection data and send it to CDMs. ('968, 4:53-5:13.)

A POSITA would have been motivated to incorporate FortiGate's update stations into Jordan to improve Jordan's scalability. Frequently, a network like Jordan's would have multiple traffic filters/gateways to guard various ingress/egress points. In that case, whenever the analytical engine generated additional content detection data, it could send the information to the update station, which could coordinate pushing the update to the various traffic filters/gateways. (Wenke-Lee, ¶157.)

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

Jordan did not explicitly disclose that the additional content detection data was stored on the traffic filter's memory. But a POSITA would have found it obvious that it was because the traffic filter would need to retrieve and use it, and retain it if the unit is turned off. (Wenke-Lee, ¶¶158-59.)

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

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

Jordan disclosed that the traffic filter used the new/updated rules for improved anomaly detection against future traffic. (Jordan, 14:49-56; *see also id.*, 16:24-29 (“[T]he analytical engine 140 can generate or update rules in optional operation 418. These rules can be communicated to the traffic filter 110 for use by the traffic filter

in filtering future traffic received from devices external to the network system 100 (e.g., via the external network 150).”; Wenke-Lee, ¶160-61.)

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

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

Jordan disclosed that the traffic filter blocked traffic that violated a policy (e.g., a set of rules). (Jordan, 9:40-45 (“[T]he traffic filter 110 can implement even recent rules...and consequently the traffic filter 110 can prevent such malicious payloads from being delivered to one or more devices 120 within the network...”); Wenke-Lee, ¶163.)

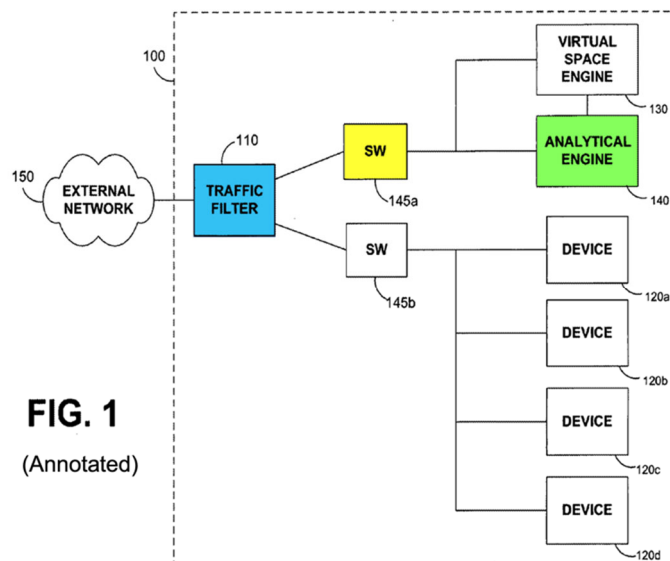
1[j]/4[h]/7[h]: “allow[ing] network traffic to pass that does not violate a policy.”

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

It would be obvious that the new rules would allow normal traffic to pass through the traffic filter. (Wenke-Lee, ¶¶164-65.)

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

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



4. Claims 3/6/9: “...wherein the analysis process facilitates an analysis that includes receiving administrator input.”

Jordan disclosed that the analytical engine could be accessed via a user interface (highlighted in red). The user interface allowed a user to access functions of the analytical engine, such as the rule generation capability (highlighted in green). (Jordan, 24:64-25:5 (“[T]he UI can access the rule generation capability of the analytical engine 140....”); *see also* Figure 3 (below); Wenke-Lee, ¶167.)

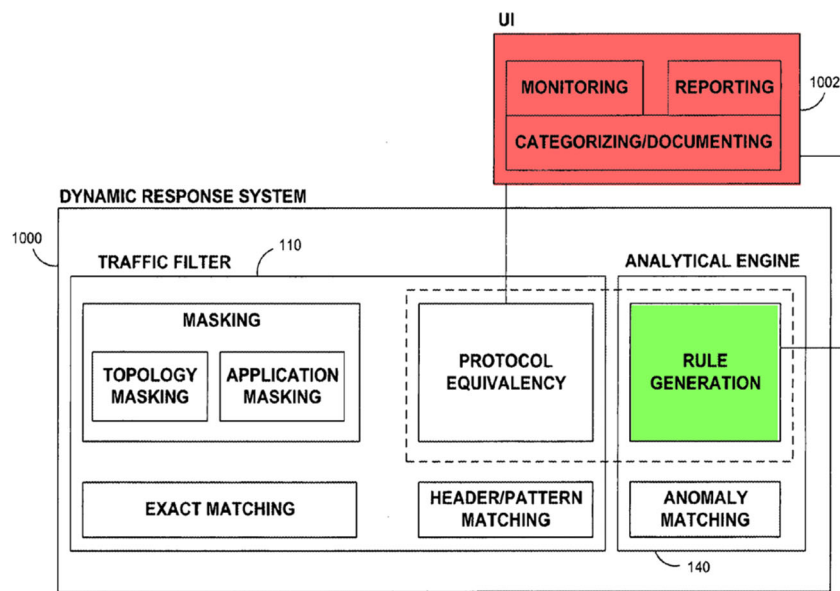


FIG. 10 (Annotated)

Because a user could access functions of the analytical engine, such as the rule generation capability, a POSITA would find it obvious that the analysis process permitted receiving administrator input. (Wenke-Lee, ¶¶167-70.) This is further evidenced by the fact that Jordan disclosed a “management module” that allowed a

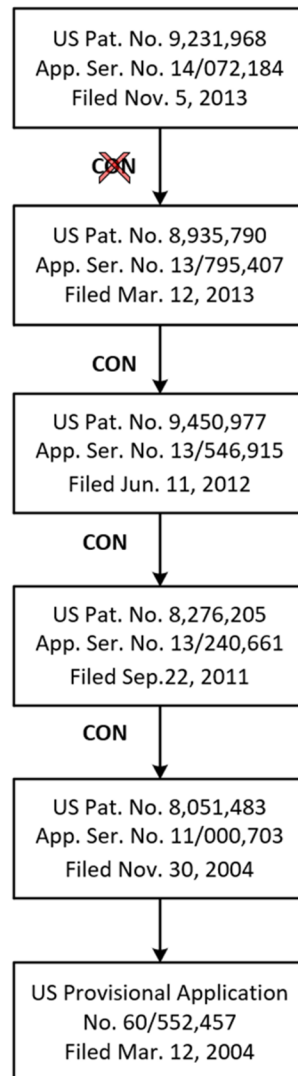
system administrator to set rules for the traffic filter. (Jordan, 25:50-54; *see also* Jordan, 25:28-49.)

5. Jordan Is Prior Art

Jordan is prior art for three independent reasons. (Wenke-Lee, ¶171.)

First, as discussed above (*see supra*, Section IV.D.3), the challenged claims are entitled to a priority date of no earlier than November 5, 2013 because they lack written description support in any parent application. Jordan's priority date is no later than November 17, 2004, the date on which its application was filed. (Wenke-Lee, ¶172.)

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

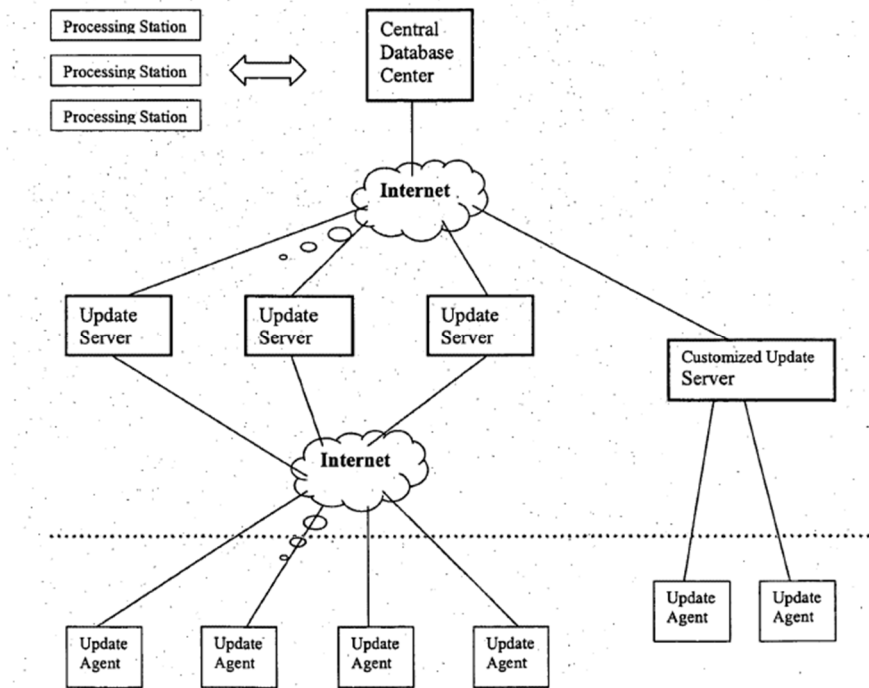


“configuration data” and “user identification”: Like the ’407 application, the ’457 provisional lacks written description support for “configuration data,” “user identification,” and their associated operations. For example, the ’457 provisional does not mention “user identification” (or even “user,”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

as virus definitions, NIDS signature, etc. They will be used by an Update Agent to check its pass-through traffic to ensure its protected network or host to be compromised by unsafe content.”); Wenke-Lee, ¶174.)

Limitations 1[e][vi]/4[c][i]/7[c][i]: The ’457 provisional application fails to disclose that a “network gateway device” (claim 1) or “data network device” (claims 4 and 9) “forward[] the network traffic to an analysis process.” (Wenke-Lee, ¶177.)

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



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

Limitation 1[e][v]/4[b][iv]/7[b][iv]: The challenged claims require “policy enforcement including user identification, content identification, and at least one of source verification and destination verification...” The ’457 provisional discloses only that an update agent used security update data (e.g., a virus definition or NIDS

signature) to process traffic. (See ’457 provisional, §§2 (update agent “inspect[d] the traffic passing through it to ensure they are safe based on the latest update it gets from an Update Server”), 3.1 (disclosing “two kinds of update data: security update data and subscriber data....”).) It discloses nothing about policy enforcement including user identification, content identification, source verification, or destination verification. (Wenke-Lee, ¶178.)

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

Jordan Claim 1	’577 Provisional Support
A method implemented by a first device, the method comprising:	Fig. 3; ¶17
receiving a first network communication from a second device, the first network communication requesting a service;	Fig. 3
responding to the second device by mangling a communication received from within a network, in response to a determination that the service is available on the network;	Fig. 3 (top half); ¶¶45, 47

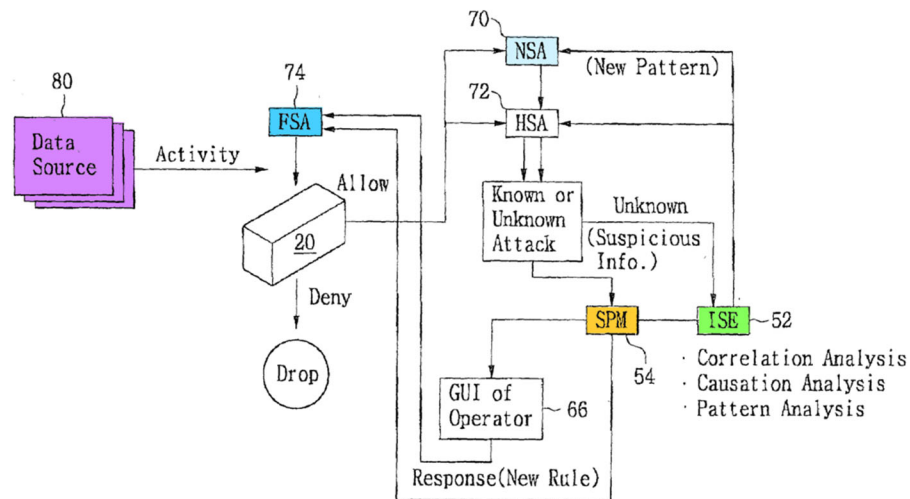
modifying a payload of a packet to remove traits of the network and to provide apparent traits;	¶¶47-48
responding to the second device that the service is available and mimicking protocols for the service, in response to a determination that the service is not available on the network;	Fig. 3 (bottom half); ¶¶47, 62
receiving a second network communication from the second device; and	Fig. 3 (bottom half)
identifying whether existing rules are present for performing a pattern recognition for the second network communication, wherein if the existing rules are not present at the first device, at least a portion of the second network communication is copied to generate new rules for pattern recognition activities.	¶¶16, 20, 32, 60, 61, 64

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

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

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

FIG 2 (Annotated)



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

modified rule was provided to the FSA (74) through the SPM (54). (*Id.*, ¶52; Wenke-Lee, ¶182.)

Magdych: Magdych disclosed a secure gateway for “detecting attacks on a network.” (Magdych, Title, Abstract; Wenke-Lee, ¶183.)

Magdych disclosed a firewall device that had a firewall portion (402) and an intrusion detection system portion (404). (*Id.*, 4:17-26.) The firewall device (210) intermediated traffic between a remote source (202) and target (206) to detect and address threats. (*Id.*, 4:17-26.) This is illustrated in Figure 4 of Magdych (below). (Wenke-Lee, ¶185.)

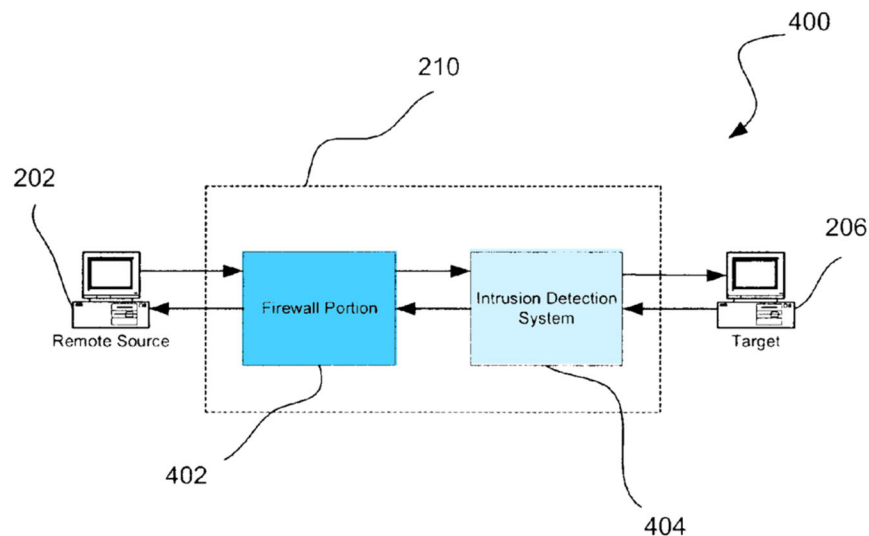


Figure 4 (Annotated)

1. Motivation to combine

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

First, both Lee and Magdych concerned network and data security; particularly, protecting networks from threats by analyzing network traffic for potential malware. (Lee, ¶10; Magdych, Abstract.) A POSITA looking to improve Lee would have looked to complementary references like Magdych. (Wenke-Lee, ¶186-87.)

Second, combining Magdych and Lee provided improved security, flexibility, efficiency, and control. (Wenke-Lee, ¶¶188-89.) As discussed above, Lee disclosed a system with separate agents performing separate functions. For example, firewall agent (FSA) applied a security policy, while network security agent (NSA) detected known attacks and suspicious traffic. (*Id.*, ¶¶46, 51-52.) Magdych recognized that existing firewalls could not filter traffic based on content, so they were less effective at screening malicious content. (Magdych, 2:29-36.) Thus, Magdych explained that there was a need to integrate intrusion detection capabilities, like the ones performed by the NSA, into firewalls. (*Id.*) With Magdych's disclosures in mind, a POSITA would have been motivated to combine Lee's NSA and FSA functions into a single firewall unit. (Wenke-Lee, ¶¶188-89.)

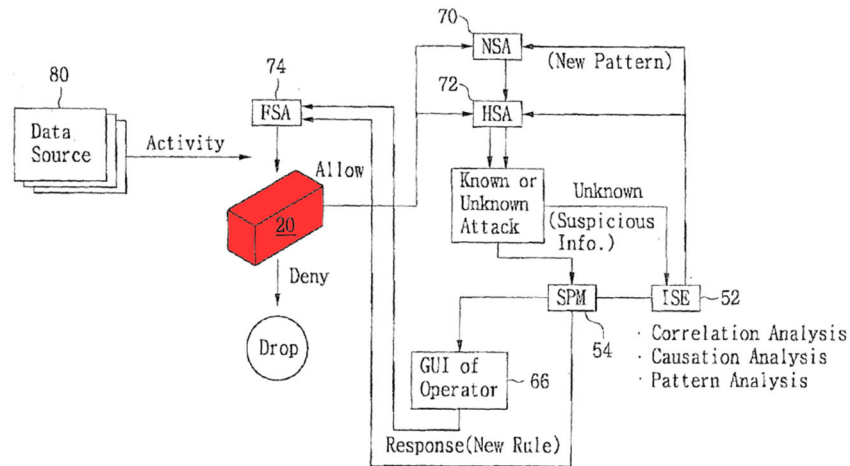
A POSITA would have made this combination for several reasons. First, it would save rack space and combining software into a single module would have been feasible given increased processor speeds and memory. Second, it would have improved Lee’s efficiency by having a single device perform at least two levels of analysis (firewall-level and content-level analyses). Third, it would have improved Lee’s security. Currently, Lee’s architecture has the NSA behind the FSA—i.e., deeper within the network. But it was standard practice to deploy traffic filters at a network’s perimeter to minimize any potential damage in the event of an attack. (Wenke-Lee, ¶190.) Finally, Lee explained that security agents could be readily added or removed. (Lee, ¶40; Wenke-Lee, ¶190-91.)

2. Claims 1/4/7

1[pre]:

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

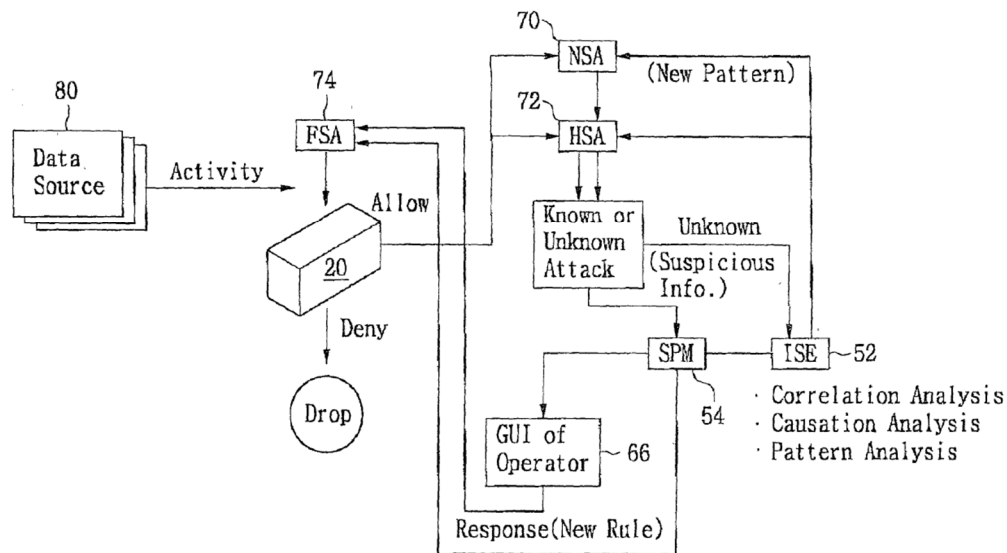
FIG 2 (Annotated)



4 ``` : ```

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

FIG 2



7[pre]:

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

Lee disclosed that “firewall 20 [was] a set of associated programs *located in a network gateway server.*” (*Id.*, ¶34.) FSA (74) was provided within the firewall and was a software program, as were the NSA and HSA. (Lee, ¶¶38-39; *see also id.*, ¶31.) Those agents “search[ed] for characteristic patterns of data over the network” and “produce[d] and transmit[ed] a security alert message to one or both of communicating devices and the security manager.” (*Id.*, ¶39.) For firewall-FSA, for example, Lee disclosed that it performed “the more CPU-intensive function of packet-by-packet inspection.” (*Id.*, ¶35.) “CPU,” of course, referred to a “central processing unit”—also known as a processor. (Wenke-Lee, ¶194-95.)

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

1[a]:

Lee’s network gateway had a processor. (Element 7[pre]; Wenke-Lee, ¶¶194-96, 197.)

1[b]:

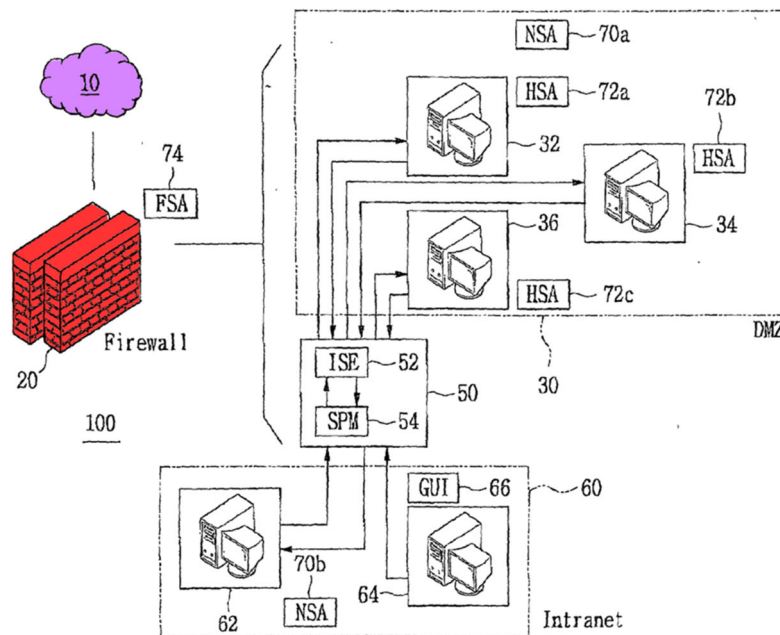
A POSITA would have found it obvious that Lee’s network gateway had memory. (Element 7[pre]; Wenke-Lee, ¶¶194-96, 198.)

1[c]:

Lee disclosed that the firewall had a network interface device.

The firewall intermediated traffic between a secure network connected and the insecure outside world. (Lee, ¶¶35-36; *see also, e.g., id.*, ¶7.) Figure 1 (below) depicted firewall 20 receiving traffic from an outside network (e.g., the Internet). (Lee, ¶35.) Lee explained that firewall 20 had “**three network interfaces**: one [went] to the inside of the intranet; one [went] to the unsecured external network 10; and the third [went] to the DMZ 30.” (Lee, ¶36; Wenke-Lee, ¶¶199-200.)

FIG 1 (Annotated)



1[d]:

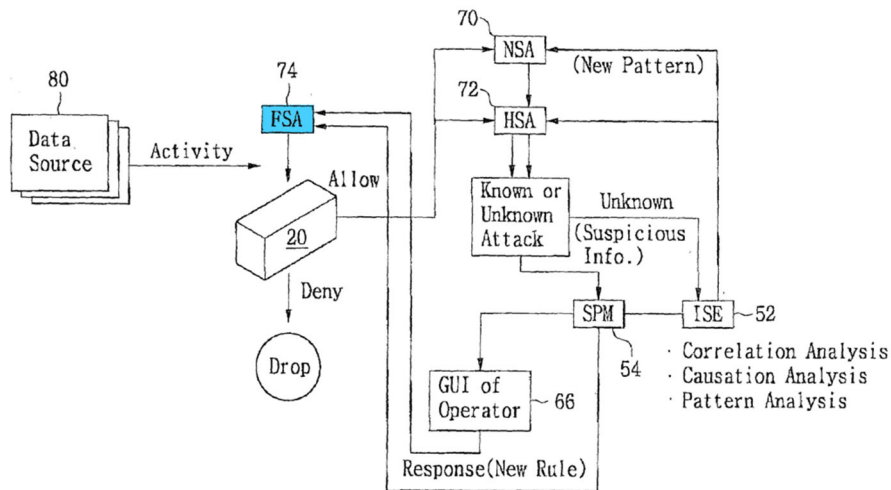
Lee disclosed content detection data used in the gateway system and Magdych disclosed configuration data used in its gateway. While neither Lee nor Magdych expressly disclosed that information being stored on memory, a POSITA would have found that obvious. (Wenke-Lee, ¶201.)

content detection data: As discussed in the Jordan-FortiGate ground, the '968 patent broadly describes “content detection data”. (See '968, 6:33-38.)

Lee disclosed that the FSA (highlighted in Figure 2 below) used various information to screen traffic. (Lee, ¶46.) For example, the firewall filtered traffic based on whether the traffic matched a signature of an attack. (Lee, ¶52 (“[W]hen

the *new pattern of attack is recognized*, a new or modified rule is given to FSA 74 through SPM 54 so that traffic from the attacker 80 can be blocked.”); *see also, e.g., id.*, ¶89.) As another example, the firewall filtered traffic based on *domain names and IP addresses*. (*Id.*, ¶34 (“The firewall 20 confirm[ed] if requests of an outsider [were] from permitted domain names or IP addresses.”); Wenke-Lee, ¶202-04.)

FIG 2 (Annotated)



configuration data: As described above, the '968 patent does not define “configuration data.” (See Section VI.A.2, limitation 1[d].) Still, Magdych disclosed that predefined firewall rules could process traffic based on selectable parameters, such as a port. For example, the configuration data indicating a particular port number, indicated the nature of a service associated with the data (e.g., HTTP, Telnet, etc.) and caused the policy to restrict access based on the port

number that was selected. (Magdych, 2:11-29, 2:55-53, 4:33-34, cl. 1; Wenke-Lee, ¶ 205.) These parameters were defined in policies. (Wenke-Lee, ¶205.)

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

While neither Lee nor Magdych expressly disclosed that the content detection data was stored in memory, a POSITA would have found that obvious based on Lee’s disclosures. As discussed, the firewall received information for analyzing traffic (e.g., rules, signatures). It was common practice at the time to store information for processing traffic on the device that was using the information to process the traffic because the device would need to retrieve the information and retain it in case the unit is shut off. (Wenke-Lee, ¶207.)

1[e]:

A POSITA would have found it obvious that Lee’s firewall had an instruction set, stored in the at least one memory device and executable by the at least one processor to detect malicious traffic and respond in real-time. (See Element 7[pre]; Wenke-Lee, ¶208.)

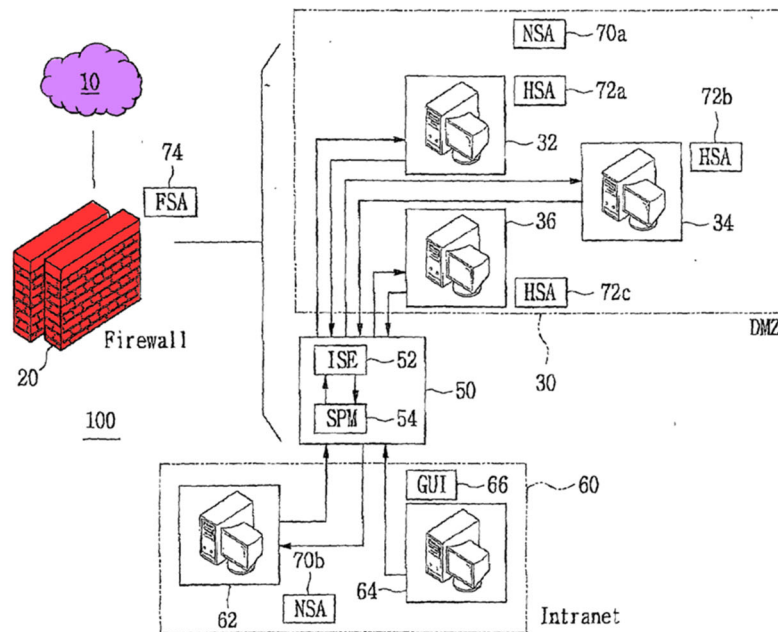
1[e][i]/4[a]/7[a]:

Lee disclosed that the firewall received data via the network interface device.

Lee disclosed that firewall 20 had a network interface device. (Element 1[c].)

Lee explained that firewall 20 had a network interface that went to an unsecured external network. (Lee, ¶36.) Moreover, as illustrated in Figure 1 (below), Lee disclosed that firewall 20 received traffic from an outside network (10). (Lee, ¶35, ¶51; Wenke-Lee, ¶209-210.)

FIG 1 (Annotated)

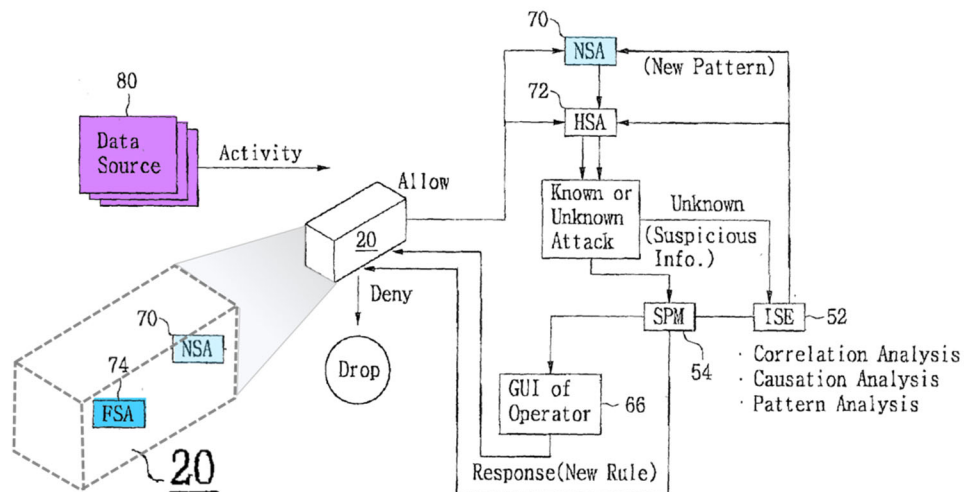


1[e][ii]/4[b][i]/7[b][i]:

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

As shown in modified Figure 2 (below), firewall 20 received traffic from a data source. The FSA then processed traffic according to the content detection data. (Lee, ¶52 (“[W]hen the new pattern of attack [was] recognized, a new or modified rule [was] given to FSA 74 through SPM 54 so that traffic from the attacker 80 can be blocked.”); *see also, e.g., id.*, ¶¶89, 34 (“The firewall 20 confirm[ed] if requests of an outsider [were] from permitted domain names or IP addresses.”), ¶36; Wenke-Lee, ¶211-12.)

FIG 2 (Redrawn and Annotated)



Magdych disclosed that received traffic was processed according to the selected parameter (e.g., port). (Magdych, cl. 1 (“discarding the data based on a predetermined set of rules....”); Wenke-Lee, ¶213.)

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

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

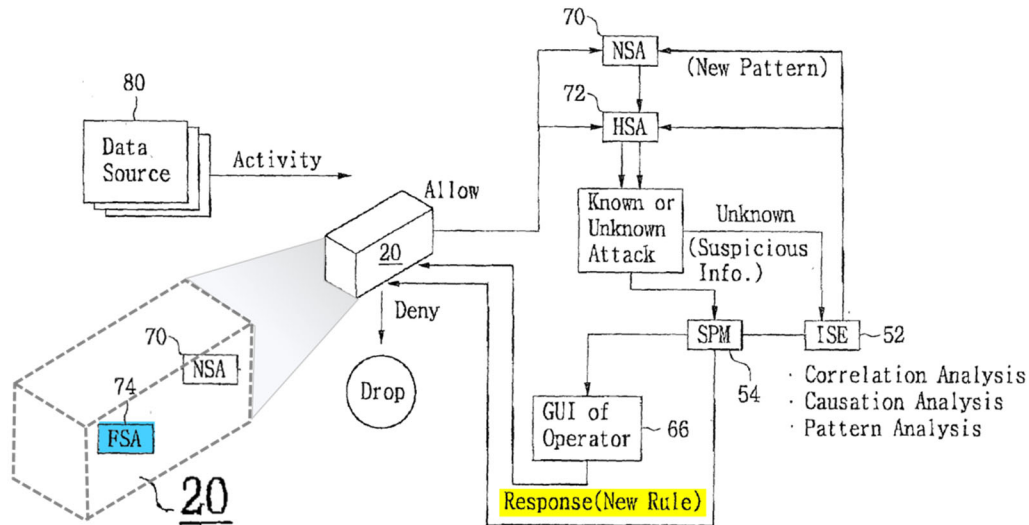
Lee disclosed or suggested that the firewall enforced policies defined by the content detection data (e.g., virus pattern or signature), and Magdych showed that a POSITA would have found it obvious that the firewall enforced policies defined by configuration data (e.g., port). (Wenke-Lee, ¶215.)

The FSA in the firewall applied policies to traffic to determine if traffic should be allowed or blocked. (Lee, ¶14; *see also, e.g.*, ¶34, 46 (“FSA 74 [was] an active agent that *adopt[ed] modified security policy* according to the decision and analysis of ISE 52 and SPM 54, and *[made] the firewall react accordingly. In order to block traffic from the attackers, FSA 74 applie[d] a security policy to the firewall 20* based on information transferred from SPM 54.”), 52, 92; Wenke-Lee, ¶216.)

The policies were defined by content detection and configuration data. For content detection data, for example, Lee disclosed that the policies/rules included a pattern or signature for detecting malicious data. (*See, e.g., id.*, ¶52 (“At the same time, when the new pattern of attack [was] recognized, a new or modified rule [was] given to FSA 74 through SPM 54 so that traffic from the attacker 80 [could] be

blocked.”).) The firewall receiving the updated rule with the content detection data is shown in Figure 2 (annotated below).) (Wenke-Lee, ¶217.)

FIG 2 (Redrawn and Annotated)



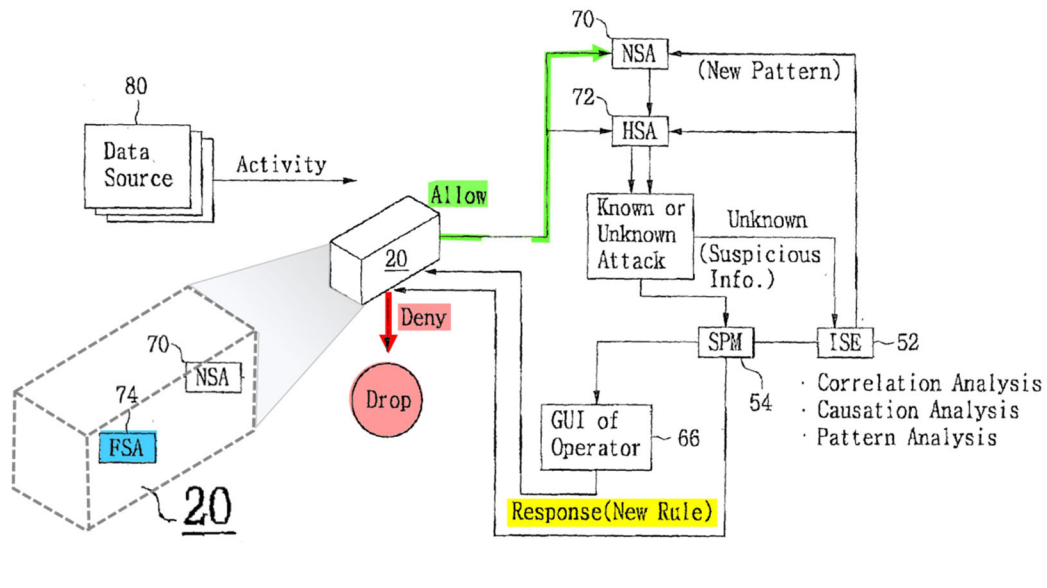
(*Id.*, Fig. 2.) A POSITA would have found it obvious that configuration data defined a policy based on Magdych and the knowledge of a POSITA. For example, Magdych disclosed that the configuration data discussed above (*see supra*, VIII.B.2, element 1[d]) defined rules and the rules instructed the firewall how to process traffic (e.g., which port(s) could traffic pass through). (Magdych, 4:36-38; Wenke-Lee, ¶¶215-17.)

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

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

Lee disclosed that the firewall included a GUI for “security policy enforcement.” (Lee, ¶32.) As part of that, the firewall determined whether to allow network traffic to pass. (*See, e.g., id.*, Lee, ¶¶47, 52.) (Wenke-Lee, ¶¶218-19.) This was shown in Figure 2 (annotated below) where firewall (20) could either allow or deny traffic.

FIG 2 (Redrawn and Annotated)



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

Lee disclosed or suggested that policy enforcement included user identification and content identification, and Magdych disclosed source and destination verification. (Wenke-Lee, ¶220.)

User identification: Lee suggested that the FSA used malware signatures that could include user identification. Specifically, Lee described an exemplary scenario where an attacker created a user ID. (Lee, ¶81.) The system could detect and react to that situation by determining if the attacker was a root user. (*Id.*, ¶84 (“If normal, ISE 52 confirms if the actor of the operation is a root user (step 154). When the actor is not a root user, a confrontation action is performed (step 156).”).) A POSITA would have found it obvious that this analysis could occur in the firewall and user identification could be used by the firewall to enforce policies that specified a particular user ID. (Wenke-Lee, ¶221.)

Content identification: Lee disclosed that policy enforcement included analyzing traffic using a malware signature. (Lee, ¶¶46, 49 (“SPM 54 instruct[ed] the application of dynamic policy to associated agents, and applie[d], to the agents, **dynamic security policies** according to a change of services provided by hosts and **the detection signatures 52 generated by ISE.**”), 92.)

If Lee did not disclose content identification, Magdych did. Magdych disclosed that “assembled packets [were] analyzed in operation 506 for the purpose of ***identifying content*** that may be harmful to known vulnerabilities of the target devices 206.” (Magdych, 5:1-3.) Obviously, this could include content like pornography. (Wenke-Lee, ¶¶222-23.)

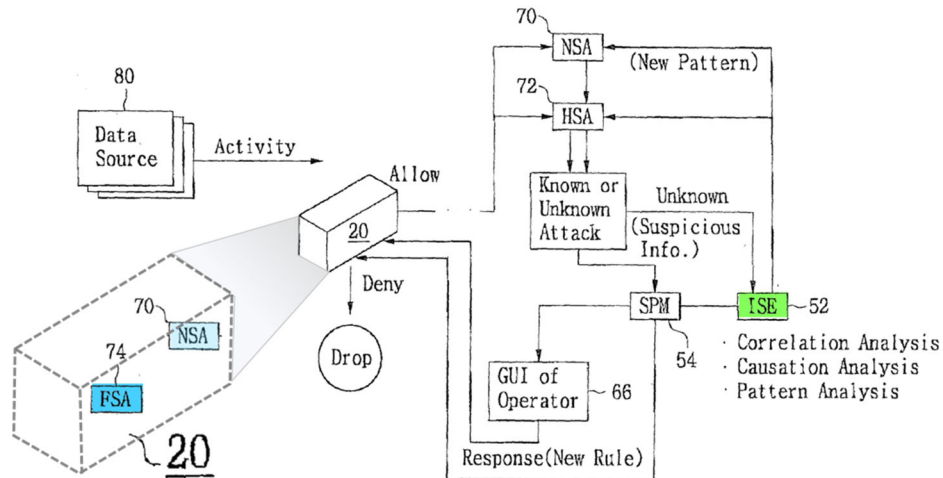
source and destination verification: Magdych disclosed that the firewall “utilize[d] a predetermined set of rules to discard received data as a function of a plurality of parameters such as **a source, a destination**, and/or a port associated with data.” (Magdych, 4:36-39; Wenke-Lee, ¶224.)

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

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

Lee disclosed that the NSA (70) “detect[ed] known attacks, suspicious information and traffic, and generate[d] a report to ISE 52.” (Lee, ¶51.) Then, the suspicious traffic was analyzed by the ISE 52, which determined if there was an attack. (*Id.*, ¶52; *see also, e.g., id.*, ¶41 (“NSA 70 confront[ed] threats against network security issue and provide[d] **analysis of suspicious traffic** and alert messages to known attacks...**NSA 70 deliver[ed] all traffic in a reduced form to ISE 52**, and ISE 52 then perform[ed] anomaly detection based on the delivered traffic.”), 68; Wenke-Lee, ¶225-26.)

FIG 2 (Redrawn and Annotated)

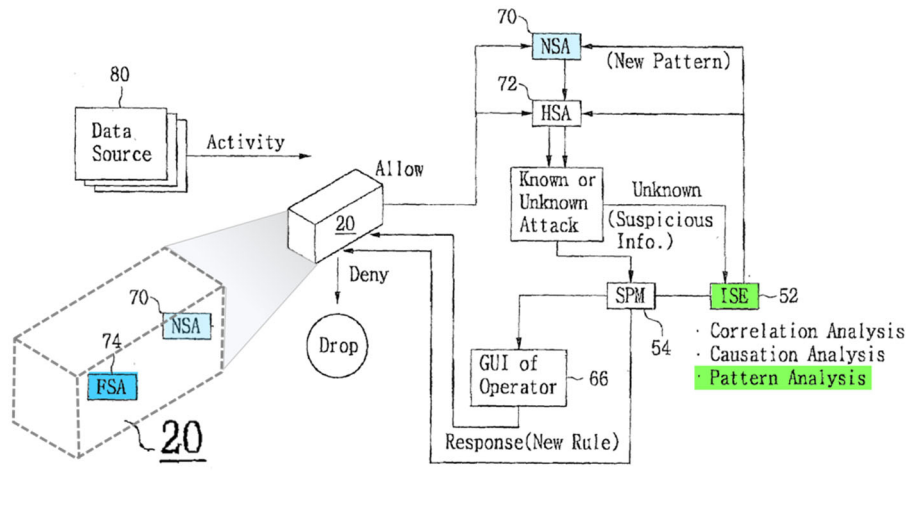


1[e][vii]/4[c][ii]/7[c][ii]:

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

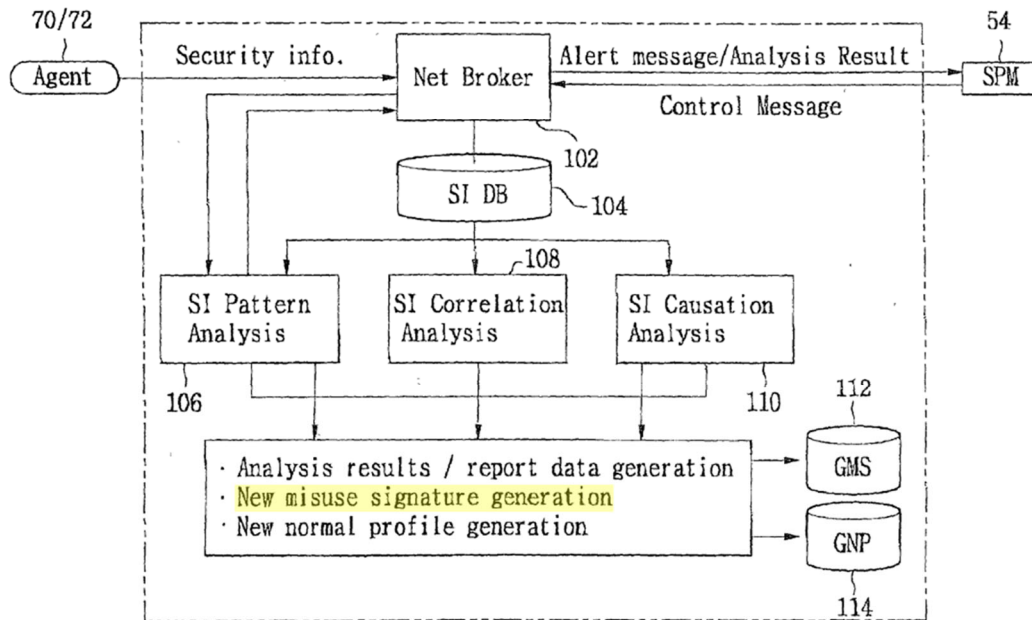
Lee disclosed that ISE (52) “perform[ed] an anomaly detection and, if detected as an attack and the attack [was] an unknown pattern, a new signature [was] generated through a learning process.” (Lee, ¶52; *see also, e.g., id.*, ¶48 (disclosing that ISE “analyze[d] alert messages from agents installed within each of individual security systems [and] determine[d] if there if [sic] an attack and **generate[d] a signature through learning.**”).) This was shown in Figure 2 (below), which depicted a Pattern Analysis in ISE (52). (Wenke-Lee, ¶227-28.)

FIG 2 (Redrawn and Annotated)



The updated signature was then transmitted to the various agents. (*Id.*, ¶52 (“The generated signature [was] transferred to NSA 70 and HSA 72, so that more rapid confrontation in response to future attacks of the same pattern [was] made possible. At the same time, when the new pattern of attack [was] recognized, a new or modified rule [was] given to FSA 74 through SPM 54 so that traffic from the attacker 80 can be blocked.”); *see also, e.g., id.* Lee, ¶¶55-56, Figure 4 (depicting “new misuse signature generation”); Wenke-Lee, ¶229-30.)

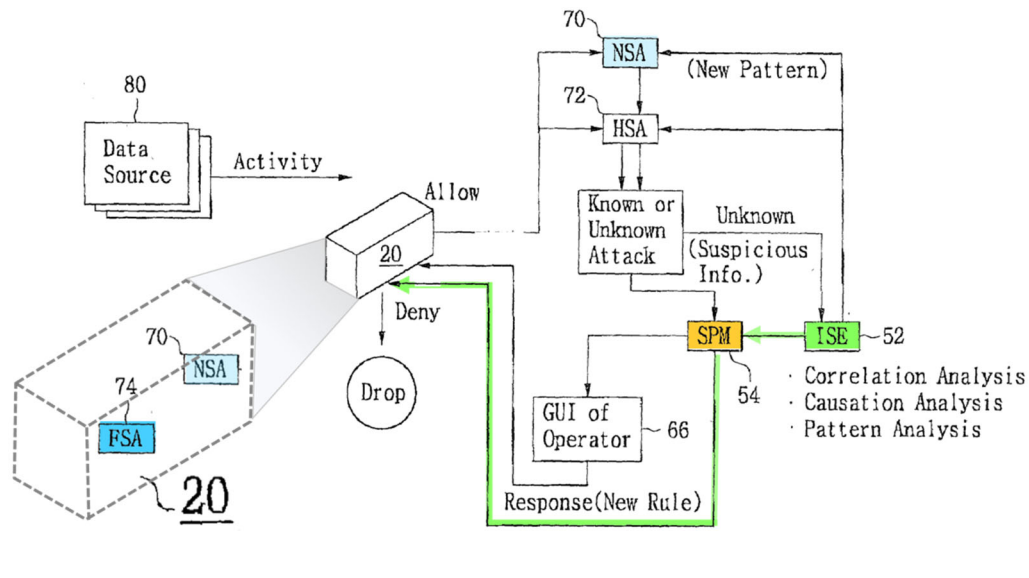
FIG 4 (Annotated)



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

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

FIG 2 (Redrawn and Annotated)



While Lee did not explicitly describe that the new rules was received via a network interface device, a POSITA would have found it obvious because the SPM transferred digital information to firewall (20) and firewall (20) had a network interface for the internal part of the network. (Lee, ¶¶36, 46; Wenke-Lee, ¶¶231-32.)

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

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

associated firewalls. (Lee, ¶89.) In view of this information, a POSITA would have understood that the FSA and NSA would store the updated rules or signatures, such as content detection data, in order to use it and apply it to future traffic. Otherwise, the information would be lost. (Wenke-Lee, ¶233.)

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

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

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

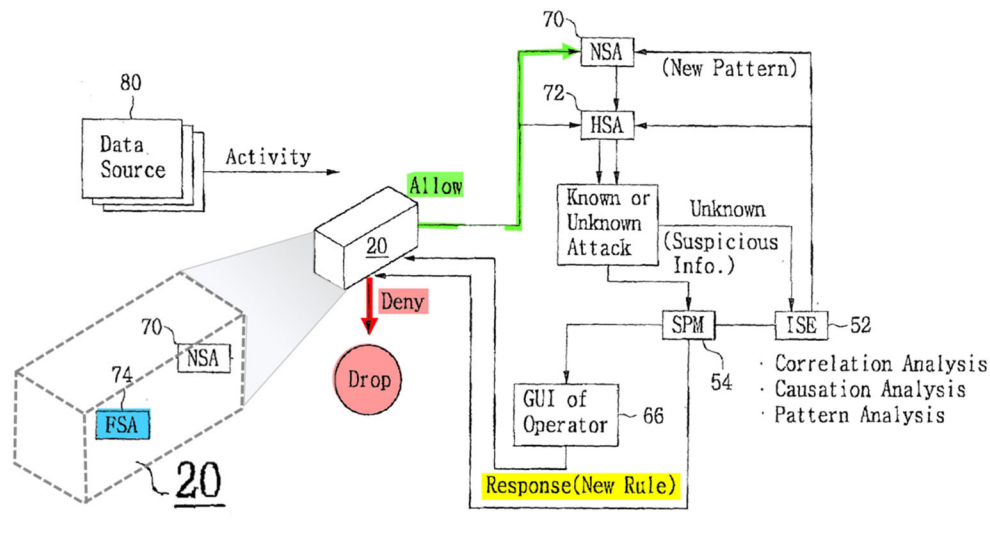
Lee disclosed blocking networking traffic determined to violate a policy. Specifically, Lee disclosed that “a new or modified rule is given to FSA (74)...so that traffic from attacker 80 *[could] be blocked.*” (Lee, ¶52; *see also, id.*, ¶¶46, 51, 89, Claim 10; Wenke-Lee, ¶235.)

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

While Lee did not explicitly disclose that traffic that did not violate a policy was allowed to pass, a POSITA would understand that Lee obviously disclosed this

limitation. Specifically, Lee describes that the FSA can block traffic from an attacker (Lee, ¶14; *see also id.*, ¶¶46, 51, 52, 89.) Lee also illustrated in Figure 2 (modified and annotated below) that the firewall allowed traffic to pass. A POSITA would understand that since the FSA would block traffic that matched the policy with the new signature, it would allow signatures to pass that did not match the policy. (Wenke-Lee, ¶236.)

FIG 2 (Redrawn and Annotated)

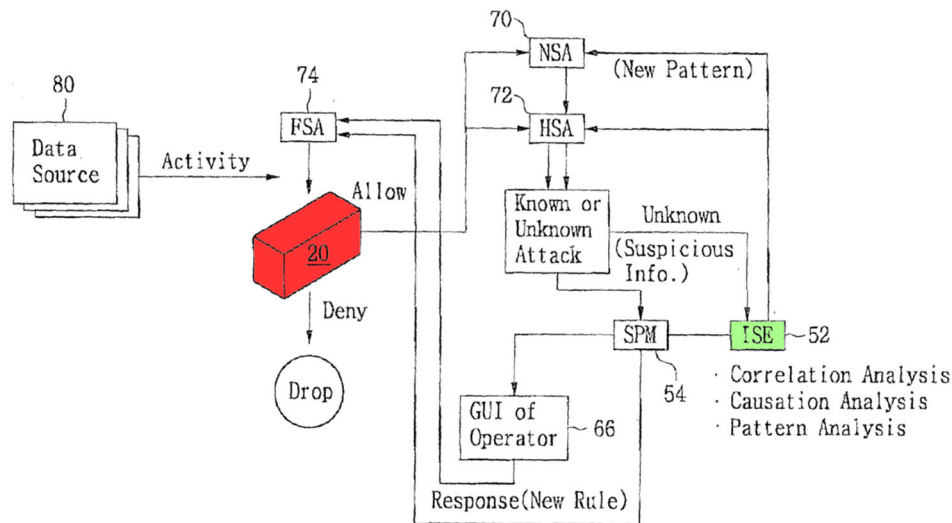


3. Claims 2/5/8:

Lee disclosed that the analysis process was executed on a computing device distinct from the data network device. Specifically, Lee described that the ISE may be located in a security management module (50), which was distinct from the firewall. (Lee, ¶38 describing that an “intelligent security management module 50 comprises ISE...” but “[f]or the firewall 20, an FSA ... is provided.”); *see also id.*,

¶47, 48 (“ISE 52 is one of the analysis engines which analyzes alert messages from agents [e.g., FSA, NSA] installed within each of individual security systems.”).) The distinction between the ISE and other components of the system was show in in Lee’s Figure 2 (annotated below). (Wenke-Lee, ¶237.)

FIG 2 (Annotated)

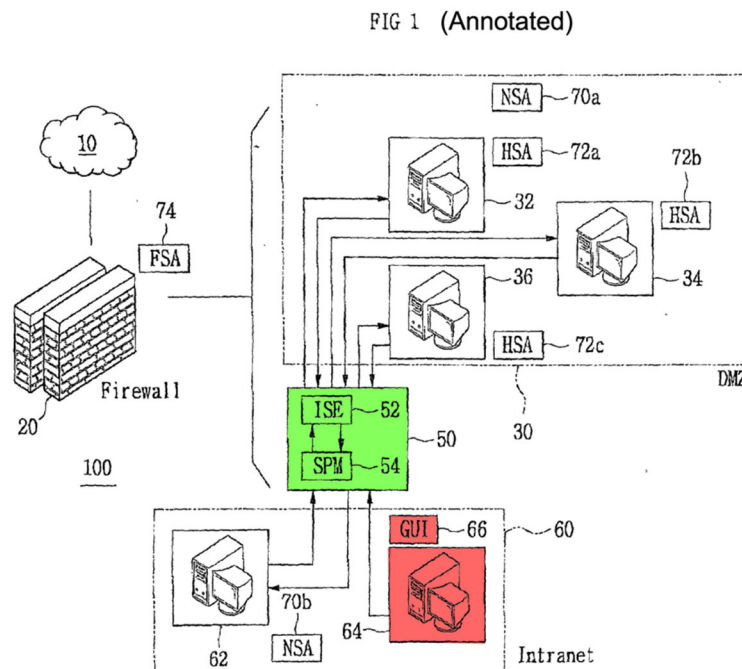


4. Claims 3/6/9:

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

Specifically, Lee described that a manager system could “control an intelligent security management module 50, [which comprised ISE 52], **through GUI.**” (Lee, ¶38.)

A POSITA would find it obvious that a GUI, which could control the ISE (e.g., the analysis process), meant that an administrator could provide input into the ISE’s analysis. This is further demonstrated in Figure 1 (annotated) of Lee. (Wenke-Lee, ¶¶238-40.)



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

If that the Board does not agree with Petitioner’s previous interpretations of “configuration data” for elements 1[d]; 1[e][ii]/4[b][i]/7[b][i];

1[e][iii]/4[b][ii]/7[b][ii], a POSITA would have found it obvious to combine the teachings of Lee and Magdych with Judge’s express disclosure of “configuration data” and examples thereof in the context of determining rules/policies for responding to threats that included viruses, spam, worms, and other attacks. (Judge, Abstract, ¶¶2, 44.) Thus, Judge serves the dual purpose of curing deficiencies, if any, of the Lee-Magdych combination and confirming the knowledge of a POSITA regarding configuration data used in detection systems. (Wenke-Lee, ¶241.)

Judge described analogous technology to the ’968 patent, Lee and Magdych, which is network security. (Judge, Abstract.) Specifically, Judge described an “architecture” with “a centralized threat management center [(TMC)] that can collect threat information and create rules and/or policies for messaging security systems [(MSS)].” (Judge, ¶40.) “An MSS is an example of an application layer security system such as hardware device 210” (*id.*, ¶78) and that hardware device “could be incorporated into a standard firewall system 140 or router...” (*id.*, ¶81; Wenke-Lee, ¶242.)

Judge further described when analyzing threat information (e.g., threatening communications/traffic), “features in conjunction with configuration data such as goals are used to produce rules.” (*Id.*, Abstract.) Judge also recognized that “configuration information can include threat types of interest and weights associated therewith.” (*Id.*, ¶44.) This configuration information could also be

modified. (*Id.*, ¶45 (“[A]n interface may be provide to allow establishing configuration information regarding one or more threat types....”); Wenke-Lee, ¶243.)

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

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

Fang⁵: Fang has the same description as the '968 patent, and precisely like the '968 patent, disclosed “systems and methods for updating content detection devices and systems.” (Fang, Title.) Thus, the description in Fang also describes or renders obvious the alleged invention of the '968 patent (*see, supra*, Section IV.D.3; Wenke-Lee, ¶245.)⁶

FortiGate: FortiGate is described in Section VIII.A.

1. Motivation to combine

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

Fang and FortiGate concerned network security and, specifically, detecting malicious content. Fang generally described means for processing network traffic content in view of certain information to determine how to handle the traffic. (Fang, Abstract.) Likewise, FortiGate provided antivirus protection and content filtering. (FortiGate, 13.)

⁵ Given that these claims lack §112 support, Fang is prior art. (*See, supra*, Section IV.D.3.)

⁶ Fang is the printed publication of the '915 application (the '968's grandparent) and has the same figures and detailed description as the '968 Patent.

Moreover, both Fang and FortiGate described higher-level modules that created updated signatures and distributed those updated signatures to the firewall units (e.g., CDMs) via an intermediary device, such as an “update station.” (*See* Section VIII.D.2, Limitation 1[f] *infra.*) Thus, a POSITA would have been motivated to improve Fang by incorporating complementary techniques from other references, like FortiGate, that addressed the same problem and consisted of a similar architecture. (Wenke-Lee, ¶247-49.)

Element-specific motivations to combine are discussed below.

2. Claims 1/4/7:

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

1[pre]:

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

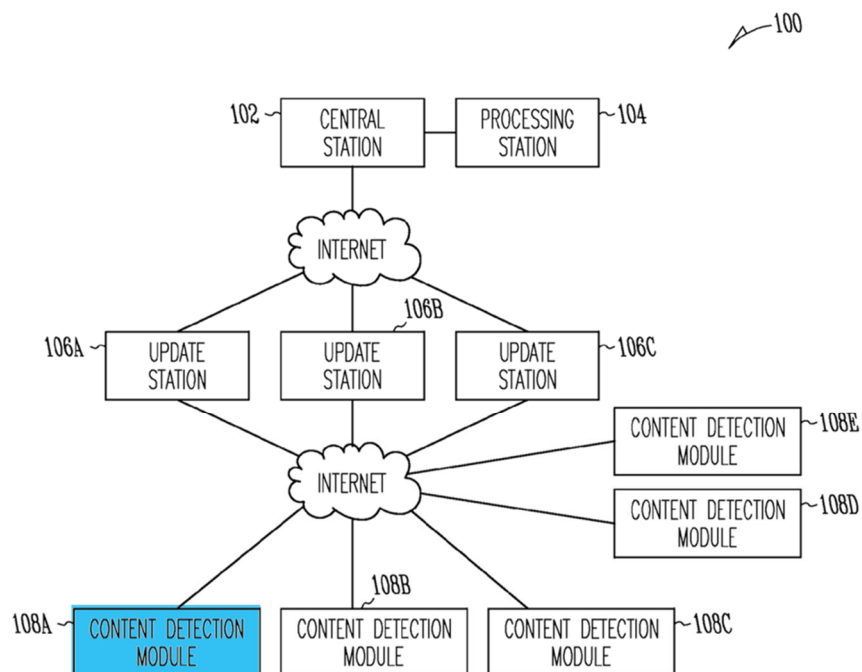


Fig. 1
(Annotated)

4[pre]:

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

7[pre]:

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

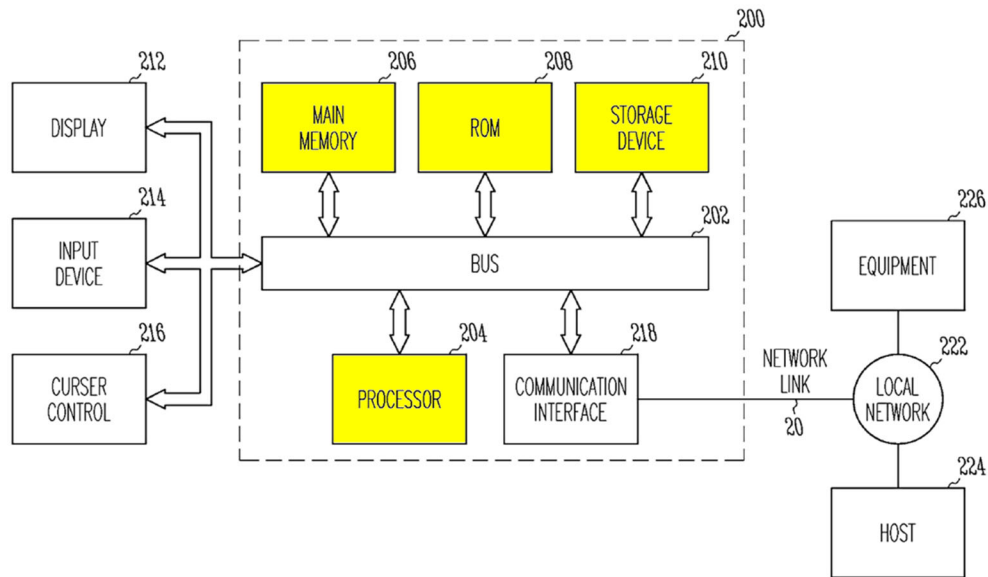


Fig. 2
(Annotated)

1[a]:

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

1[b]:

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

1[c]:

Fang disclosed that CDM (108) had a communication interface, annotated in Figure 2 (below). (Fang, ¶53.) Communication interface (218) provided, *inter alia*,

“a two-way data communication coupling to a network link 220 that is connected to a local network 222.” (*Id.*; Wenke-Lee, ¶1257.)

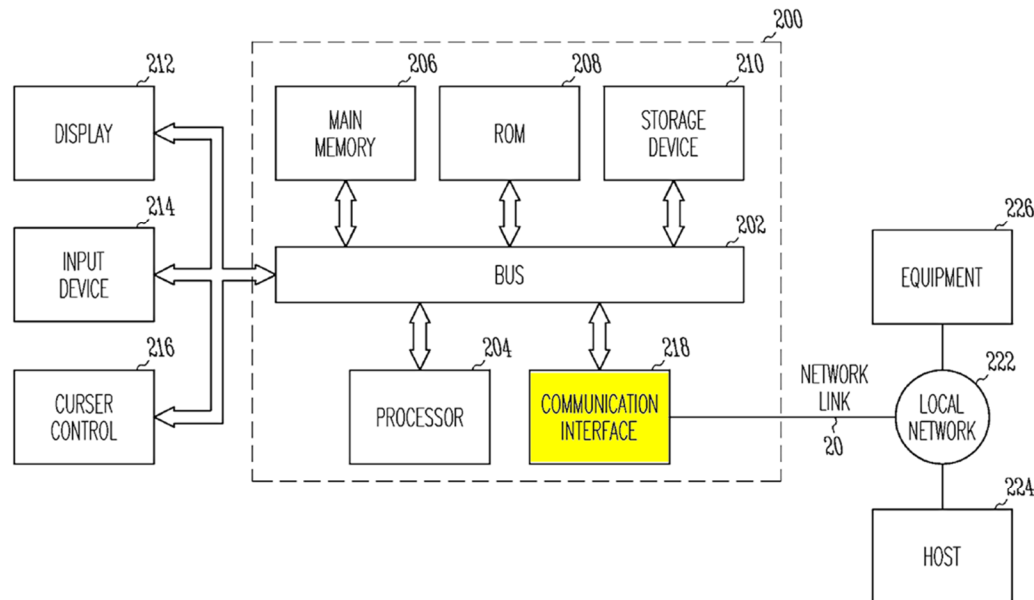


Fig. 2
(Annotated)

Moreover, Fang disclosed a “network interface device” for receiving content. (*Id.*, Cls. 7, 13.) A POSITA would have understood that the network interface device and communication interface referred to the same thing because POSITAs generally used them interchangeably. (Wenke-Lee, ¶¶257-59.) Figure 2 of Fang illustrates the communication interface (218).

1[d]:

Fang disclosed content detection data (e.g., a virus signature) stored in the memory device. While Fang did not disclose configuration data, a POSITA would

have found it obvious to add that information to Fang based on FortiGate’s disclosures. (Wenke-Lee, ¶260.)

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

configuration data: As discussed in Section IV.D.3, the challenged patent failed to disclose “configuration data.” By extension, Fang too failed to disclose “configuration data.” Still, a POSITA would have found it obvious to add configuration data to Fang based on FortiGate’s disclosures. (*See supra*, Section VIII.A.2, Limitation 1[d].)

By incorporating these configurations and the others disclosed in FortiGate, Fang’s CDM would have improved flexibility and features for filtering and controlling traffic. These configurations could naturally be incorporated into Fang’s CDM, which likewise processed incoming traffic like the FortiGate firewall. (Wenke-Lee, ¶¶262-63.)

stored on the at least one memory device: While Fang and FortiGate did not expressly disclose storing content detection and configuration on memory, a POSITA would have found that was obvious based on their disclosures. For starters,

Fang disclosed that CDM (108) received content detection data from an update station and determined whether content contained undesirable content based on the content detection data. (*Id.*, ¶31.) For example, CDM (108) could use a virus signature received from an update station to detect a virus. (*Id.*) Likewise, FortiGate disclosed that configuration data was defined in a policy that FortiGate used when processing traffic. (*E.g.*, FortiGate, 35.) Accordingly, a POSITA would have found it obvious that the content detection and configuration data were stored in memory because it would need to be in order for the network gateway/firewall to access and use the information when processing traffic. (Wenke-Lee, ¶264.)

1[e]:

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

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

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

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

While Fang did not expressly disclose that CDM (108) received the traffic over a network interface device, a POSITA would have found it obvious that it did based on Fang's disclosures. Specifically, as discussed above, Fang disclosed a communication interface that provided two-way data communications. Fang also disclosed that the communication interface "[c]omputer system 200 [could] send messages and receive data, including program code, through the network(s), network link 220, and communication interface 218." (*Id.*, ¶54; *see also, e.g., id.*, cl. 7 ("receiving content via a network interface device;").) Moreover, Figure 1 depicted CDMs connecting to the Internet. Consequently, a POSITA would have found it obvious that the CDM received traffic over a network interface device. (Wenke-Lee, ¶266-68.)

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

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

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

configured to detect virus based on a virus signature received from update station 106.” (Wenke-Lee, ¶269-770.)

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

Table 4: Factory default firewall configuration

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

(Annotated)

Further, as described above in limitation 1[d], and as shown above, the firewall processed traffic according to the configuration settings. (FortiGate, 34-35; *see also* Section VIII.A.2, limitation 1[d] (FortiGate discussion in Ground I).) Thus, FortiGate disclosed that the received traffic was processed in view of configuration data. (Wenke-Lee, ¶271-72.)

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

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

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

As addressed above in element 1[d], Fang did not disclose that policy enforcement included configuration data. However, FortiGate described configuration settings, which included allowing an administrator to configure

“Traffic Shaping” and “Authentication,” for example. (FortiGate, 34-35; *see also id.* 16.)

Fang also disclosed that the CDM detected “suspicious” content. For example, Fang disclosed that “content detection module 108 may detect a content that [was] suspicious (or that requires further processing)....” (*Id.*, ¶33.) While Fang did not disclose that what constituted “suspicious” was defined in a policy, a POSITA would have found that obvious because a network administrator would need to define what constitutes “suspicious” so the CDM could detect the data. (Wenke-Lee, ¶273-77.)

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

Fang disclosed that the CDM performed policy enforcement. (Fang, ¶31.) Fang explained that “‘policy enforcement’ refer[ed] to a process or procedure, an execution of which *create[d] a result that can be used to determine whether to pass data to user*....” (*Id.*; Wenke-Lee, ¶278.)

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

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

While Fang did not explicitly disclose user identification, a POSITA would have found it obvious to add the parameter based on FortiGate, as discussed above. (*See supra*, Section VIII.A.2, Limitation 1[e][v]; Wenke-Lee, ¶279-80.)

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

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

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

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

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

[could] be used by content detection modules 108 to detect the undesirable content.”

(*Id.*; Wenke-Lee, ¶281-83.)

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

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

Fang disclosed that “[u]pdate stations 106 receive[d] the content detection data from central station 102, and sen[t] the content detection data to content detection modules 108 (e.g., through the Internet).” (Fang, ¶30.) Figure 1 of Fang is also illustrative of the update station acting as an intermediary device distributing the content detection data to the CDMs. (Wenke-Lee, ¶284-85.)

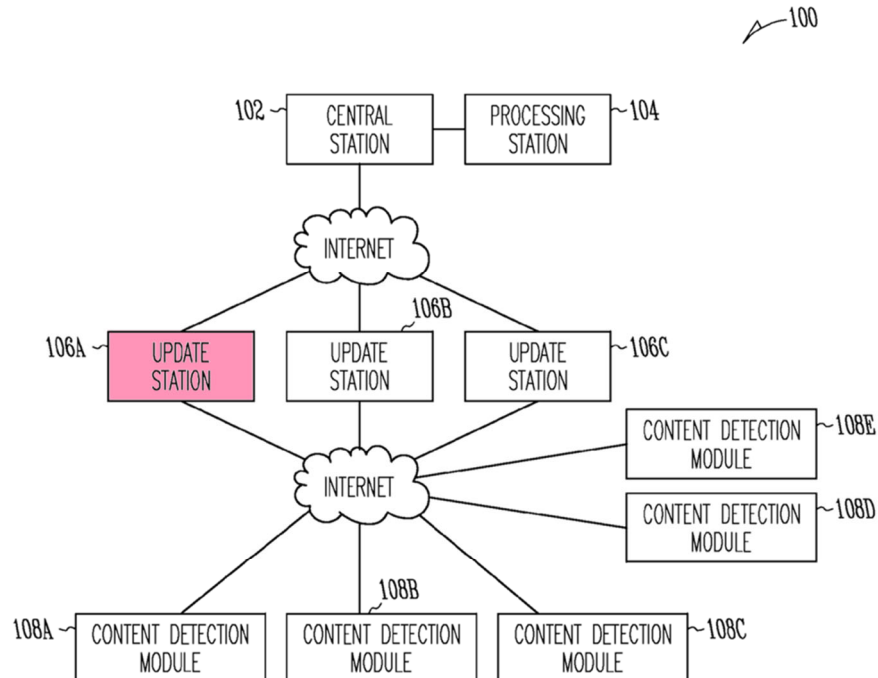


Fig. 1
(Annotated)

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

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

Fang disclosed that the CDM received content detection data from an update station and determined whether content contained undesirable content based on the content detection data. (*Id.*, ¶31.) For example, CDM could use a virus signature received from an update station to detect a virus. (*Id.*). A POSITA would have found it obvious that the content detection data was stored in memory because it would need to be for the CDM to access and use the information when processing traffic. (Wenke-Lee, ¶286-87.)

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

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

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

Fang did not explicitly disclose that the CDMs “block” traffic, but a POSITA

would find it obvious. Fang described that the CDM performed policy enforcement to determine whether to pass data (e.g., traffic) to a user. (Fang, ¶31.) A POSITA would understand that by determining whether to “pass” data means that data may alternatively be blocked. (Wenke-Lee, ¶289.)

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

Fang described that the CDM allowed traffic that did not violate a policy to pass. (Fang, ¶31; Wenke-Lee, ¶290.)

3. Claims 2/5/8:

Fang disclosed that the analysis process was executed on a distinct device. For example, Figure 1 of Fang disclosed that the processing station was a separate device from the CDM and was separated by the Internet. (Wenke-Lee, ¶291.)

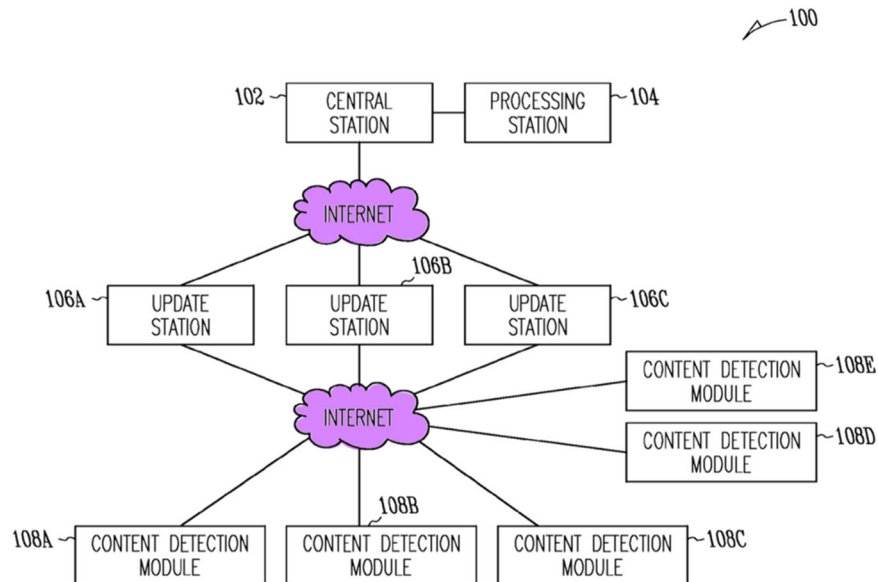


Fig. 1
(Annotated)

4. Claims 3/6/9:

Fang disclosed the analysis processing facilitated an analysis that included receiving administrator input. (Fang, ¶34 (“Alternatively, the set of content data can be analyzed by an administrator....”); Wenke-Lee, ¶292.)

IX. CONCLUSION

Accordingly, the Board should institute IPR and cancel the challenged claims.

Dated: February 1, 2023

Respectfully submitted,

/Thomas Millikan/

Lead Counsel

Thomas Millikan, Reg. No. 72,316

PERKINS COIE LLP
11452 El Camino Real, Ste 300
San Diego, California 92130-2080
858.720.5700 (phone)
858.720.5799 (fax)

Back-up Counsel
Babak Tehranchi, Reg. No. 55,937
Andrew Klein, *PHV to be requested*
Cory Owan, Reg. No. 79,841

Attorneys for Netskope, Inc.

CERTIFICATE OF WORD COUNT UNDER 37 CFR §42.24(d)

Under 37 C.F.R. §42.24(d), the undersigned certifies that the word count for this *Petition for Inter Partes Review* totals 13,876 words, excluding the parts exempted by 37 C.F.R. §42.24(a).

The word count was made using the built-in word count function in the Microsoft® Word 365 software used to prepare this document.

Dated: February 1, 2023

Respectfully submitted,

/Thomas Millikan/

Lead Counsel

Thomas Millikan, Reg. No. 72,316

PERKINS COIE LLP
11452 El Camino Real, Ste 300
San Diego, California 92130-2080
858.720.5700 (phone)
858.720.5799 (fax)

Back-up Counsel
Babak Tehrani, Reg. No. 55,937
Andrew Klein, *PHV to be requested*
Cory Owan, Reg. No. 79,841

Attorneys for Netskope, Inc.

CERTIFICATE OF SERVICE

The undersigned hereby certifies that true copies of this *PETITION FOR INTER PARTES REVIEW* and supporting materials (Exhibits 1001-1021 and Power of Attorney) have been served via USPS Express Mail and electronically on counsel for patent owner as follows:

Law Office of Dorian Cartwright
P.O. Box 6629
San Jose, CA 95150
UNITED STATES
info@cartwrightesq.com

In addition, the undersigned certifies that a true copy of the petition and supporting materials have been served on Patent Owner via email to its legal counsel:

Andrew Michael Holmes - drewholmes@quinnemanuel.com
Jordan Brock Kaericher - jordankaericher@quinnemanuel.com
Sean S. Pak - seanpak@quinnemanuel.com
Tigran Guledjian - tigranguledjian@quinnemanuel.com

Dated: February 1, 2023

Respectfully submitted,

/Thomas Millikan/

Lead Counsel

Thomas Millikan, Reg. No. 72,316

PERKINS COIE LLP
11452 El Camino Real, Ste 300
San Diego, California 92130-2080
858.720.5700 (phone)
858.720.5799 (fax)

Back-up Counsel
Babak Tehranchi, Reg. No. 55,937
Andrew Klein, *PHV to be requested*
Cory Owan, Reg. No. 79,841

Attorneys for Netskope, Inc.