

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Fortinet, Inc.,

Petitioner,

v.

Netskope, Inc.,

Patent Owner.

Case No. 2025-0041

U.S. Patent 8,397,282

PETITION FOR *INTER PARTES* REVIEW

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

	<u>Page</u>
TABLE OF ABBREVIATIONS	viii
EXHIBIT LIST	ix
I. INTRODUCTION	1
II. Requirements for inter partes review	2
A. GROUNDS FOR STANDING (37 C.F.R. §42.104(a)).....	2
B. Identification of Challenge.....	3
1. The Specific Art on Which the Challenge is Based.....	3
2. Statutory Grounds on Which the Challenge is Based.....	3
3. How the Challenged Claims are Unpatentable	4
III. THE PATENT AND STATE OF THE ART	4
A. The 282 Patent.....	4
B. File History.....	6
C. POSITA.....	7
IV. Claim Construction.....	7
A. "wherein the set of firewall rules is dynamically self- configurable during runtime"	7
B. "chains of rules forming various paths through a hierarchical structure"	8
C. "defined places for dynamically updating the set of firewall rules during runtime"	8
V. Grounds of Unpatentability	9

A.	Grounds 1 and 2 – Claims 1-35 Are Anticipated by And At Minimum Obvious Over Coss	10
1.	Overview of Coss.....	10
2.	Independent Claim 1	13
3.	Claim [2]: "A method according to claim 1, wherein dynamically updating the set of firewall rules during runtime further comprises, during runtime, adding a rule to the set of firewall rules, deleting a rule from the set of firewall rules, or modifying a rule in the set of firewall rules without operator interaction."	26
4.	Claim [3]: "A method according to claim 1, wherein associating the set of firewall rules with the at least one node further comprises associating a first subset of the set of firewall rules with the first node."	27
5.	Claim [4]: "A method according to claim 3, wherein the at least one node further comprises at least two nodes including a second node, further comprising associating a second subset of the set of firewall rules with the second node."	29
6.	Claim [5]: "A method according to claim 4, wherein the first subset of firewall rules is the same as or at least partially different from the second subset of firewall rules."	30
7.	Claim [6]: "A method according to claim 4, wherein one of the first subset of firewall rules or second subset of firewall rules equals the entire set of firewall rules."	31
8.	Claim [7]: "A method according to claim 1, further comprising, after receiving the packet and prior to accepting or denying the packet, conditioning the packet based on the set of firewall rules."	33

9.	Claim [8]: "A method according to claim 7, wherein conditioning the packet based on the set of firewall rules further comprises rewriting a portion of a network packet header associated with the packet."	34
10.	Claim [9]: "A method according to claim 1, wherein each of the at least one node is associated with at least two network interfaces."	35
11.	Claim [10]: "A method according to claim 1, wherein each of the two or more network interfaces is connected with at least one physical device."	35
12.	Claim [11]: "A method according to claim 1, wherein the set of firewall rules being dynamically self-configurable further comprises dynamically updating the set of firewall rules during runtime without operator interaction."	36
13.	Independent Claim 12	37
14.	Claim [13]: "A device according to claim 12, wherein the data controlling computer program code is further executable to, while the firewall is processing traffic through the at least one node, add a rule to the set of firewall rules, delete a rule from the set of firewall rules, or modify a rule in the set of firewall rules."	39
15.	Claim 14	39
16.	Claim 15	39
17.	Claim [16]: "A device according to claim 15, wherein the first subset of the set of firewall rules is the same as or different from the second subset of the set of firewall rules."	40
18.	Claim [17]: "A device according to claim 15, wherein either the first subset of the set of firewall rules or the second subset of the set of firewall rules equals the entire set of firewall rules."	40

19.	Claim [18]: "A device according to claim 12, wherein the data controlling computer program code is further executable to condition the packet based on the set of firewall rules."	40
20.	Claim [19]: "A device according to claim 18, wherein conditioning the packet based on the set of firewall rules further comprises rewriting a portion of a network packet header associated with the packet."	40
21.	Claim [20]: "A device according to claim 12, wherein each of the at least one node is associated with at least two network interfaces."	40
22.	Claim [21]: "A device according to claim 12, wherein each of the plurality of network interfaces is connected with at least one physical device."	41
23.	Claim [22]: "A device according to claim 12, wherein each of the plurality of network interfaces is physically connected to every other network interface of the plurality of network interfaces and wherein physical connection between the plurality of network interfaces comprises indirect physical connection between the plurality of network interfaces."	41
24.	Claim [23]: "A device according to claim 12, wherein the data controlling computer program code is further executable to dynamically update the set of firewall rules during runtime without operator interaction."	42
25.	Independent Claim 24	42
26.	Claim [25]: "A data controlling computer program product according to claim 24, wherein the computer instructions are further executable to dynamically update the set of firewall rules during runtime without operator interaction."	43

27.	Claim [26]: "A data controlling computer program product according to claim 25, wherein the computer instructions are further executable to, while the firewall is processing traffic through the at least one node, add a rule to the set of firewall rules, delete a rule from the set of firewall rules, or modify a rule in the set of firewall rules."	44
28.	Claim 27.....	44
29.	Claim 28.....	44
30.	Claim [29]: "A data controlling computer program product according to claim 28, wherein the first subset of the set of firewall rules is the same as or different from the second subset of the set of firewall rules."	45
31.	Claim [30]: "A data controlling computer program product according to claim 28, wherein either the first subset of the set of firewall rules or the second subset of the set of firewall rules equals the entire set of firewall rules."	45
32.	Claim [31]: "A data controlling computer program product according to claim 24, wherein the computer instructions are further executable to condition the packet based on the set of firewall rules."	45
33.	Claim [32]: "A data controlling computer program product according to claim 31, wherein conditioning the packet based on the set of firewall rules further comprises rewriting a portion of a network packet header associated with the packet."	45
34.	Claim [33]: "A data controlling computer program product according to claim 24, wherein each of the at least one node comprises at least two network interfaces."	45

35.	Claim [34]: "A data controlling computer program product according to claim 24, wherein each of the two or more network interfaces is connected with at least one physical device."	46
36.	Claim [35]: "A data controlling computer program product according to claim 24, wherein each of the two or more network interfaces is physically connected to every other network interface of the two or more network interfaces and wherein physical connection between the two or more network interfaces comprises indirect physical connection between the two or more network interfaces."	46
B.	Ground 3 – Claims 1-35 Are Obvious over Coss In View of Ke	46
VI.	MANDATORY Notices Under 37 C.F.R. §42.8(a)(1)	51
A.	37 C.F.R. §42.8(b)(1): Real Parties-In-Interest.....	51
B.	37 C.F.R. §42.8(b)(2): Related Matters	51
C.	37 C.F.R. §42.8(b)(3)-(4): Lead And Back-Up Counsel And Service Information.....	51
VII.	FEES UNDER 37 C.F.R. §42.103	52
VIII.	CONCLUSION	52

TABLE OF AUTHORITIES

Cases

Phillips v. AWH Corp.,
415 F.3d 1303 (Fed. Cir. 2005) 7

Statutes

35 U.S.C. §102..... 3
35 U.S.C. §103..... 3, 6
35 U.S.C. §311..... 1

Rules

37 C.F.R. §42.103 52
37 C.F.R. §42.104(a) 2
37 C.F.R. §42.104(b) 3
37 C.F.R. §42.104(b)(4) 4
37 C.F.R. §42.104(b)(5) 4
37 C.F.R. §42.21 1
37 C.F.R. §42.8(a)(1)..... 51
37 C.F.R. §42.8(b)(1) 51
37 C.F.R. §42.8(b)(2) 51
37 C.F.R. §42.8(b)(3) 51
37 C.F.R. §42.8(b)(4) 51

TABLE OF ABBREVIATIONS

Abbreviation	Full Name
'282 Patent	U.S. Patent No. 8,397,282
Challenged Claims	Claims 1-35 of the '282 Patent
Patent Owner, PO	Patent Owner Netskope, Inc.
Petitioner	Petitioner Fortinet, Inc.
POSITA	Person of Ordinary Skill In The Art

EXHIBIT LIST

Exhibit No.	Document
1001	U.S. Patent No. 8,397,282 ("the '282 patent")
1002	File History of the '282 patent
1003	Declaration of John R. Black
1004	U.S. Patent No. 6,154,775 ("Coss")
1005	U.S. Patent Publication No. 2003/0041266 ("Ke")

I. INTRODUCTION

Petitioner Fortinet, Inc. ("Petitioner") respectfully petitions, under 35 U.S.C. §311 and 37 C.F.R. §42.21, for *inter partes* review ("IPR") of claims 1-35 ("the Challenged Claims") of U.S. Patent No. 8,397,282 ("the '282 patent") (EX1001) on the grounds below.¹

The '282 patent generally relates to a network firewall protecting one or more nodes on a network, such as the entry point to a LAN, WAN, or VPN—each node with at least two network interfaces. By approaching the network as a series of nodes, the firewall can associate different sets of rules with each node. The firewall processes data through the firewall on a packet-by-packet model using hierarchical "chains of rules" to apply to each packet entering and leaving the firewall. The purported invention is that these chains of rules can be updated "dynamically" at runtime without human intervention. But neither processing packets through a firewall using firewall rules, nor allowing those rules to be updated dynamically is novel or non-obvious.

The specification explains that these chains of rules are merely "serialized sequences of one or more rules" representing "classic 'if-then' logic, as might be supported by the syntax of a software programming language." EX1001, 5:5-8,

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

6:57-59. As the specification admits, the disclosed "firewall functions are relatively standard and basic capabilities that can be found in most firewall implementations." *Id.*, 6:62-65.

It is therefore unsurprising that the claims are obvious in view of the prior art. Specifically, **Coss** (EX1004) already discloses a firewall designed to protect multiple LANs—*e.g.*, domains and sub-domains—using a dynamically adaptable firewall model that can add and change rule sets during runtime. EX1004, 2:33-46, 3:36-4:3. **Coss** does not go into explicit detail on how to configure a given domain, but **Ke** (EX1005) provides such implementation details. Specifically, **Ke** discloses an internet security system protecting a series of virtual local area networks (VLANs) using a packet-based firewall model. EX1005, [0032]-[0035]. Like **Coss**, **Ke's** firewall model allows for different policies to be applied to each such VLAN. *Id.*, [0040] And **Ke** provides express details on steps for configuring the protected VLAN domains. *Id.*, *e.g.*, [0069]-[0073].

II. REQUIREMENTS FOR INTER PARTES REVIEW

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

Petitioner certifies that the '282 patent is available for IPR and that Petitioner is not barred or estopped from requesting IPR of the Challenged Claims on the grounds in this petition. The '282 patent issued more than 9 months ago, and Petitioner was served with the complaint alleging infringement of the '282 patent

less than one year ago.

B. Identification of Challenge

Pursuant to §§42.104(b), Petitioner requests IPR of claims 1-35, and that the Board cancel the same as unpatentable.

1. The Specific Art on Which the Challenge is Based

Name	Exhibit	Patent / Publication	Filed	Issued / Published	Prior art under at least
Coss	1004	U.S. 6,154,775	9/12/1997	11/28/2000	102(b)
Ke	1005	U.S. 2003/0041266	9/27/2001	2/27/2003	102(b)

Each of the above references is prior art to the '282 patent based on the purported 03/10/2004 priority dates for the provisional applications to which the '282 patent claims priority.²

2. Statutory Grounds on Which the Challenge is Based

Ground	Claims	Basis	Prior Art
1	1-35	§102	Coss
2	1-35	§103	Coss
3	1-35	§103	Coss in view of Ke

² For purposes of this filing, Petitioner takes no position on whether the '282 patent is entitled to the 03/10/2004 priority date and the unpatentability of the claims over the art cited in this IPR does not require resolution of this issue.

3. How the Challenged Claims are Unpatentable

Petitioner provides the information required under §§42.104(b)(4)-(5) in §V.

III. THE PATENT AND STATE OF THE ART

A. The '282 Patent

The '282 patent generally relates to a dynamically configurable firewall. EX1001, Abstract. Specifically, "network firewalls that can dynamically adapt to changing conditions and operator requirements." EX1001, 1:35-38. The alleged invention purportedly "provides a new level of flexibility including, but not limited to, dynamically adding new network interface abstractions or groupings of interface abstractions and tailoring the behavior of those abstractions to the network client devices' specific needs." EX1001, 2:46-50. The sources and destinations of network traffic are referred to as "nodes." *Id.*, 2:56-64. EX1003, ¶42.

As the '282 specification explains, "[e]ach node...is simultaneously a source of and destination for network packets[and p]ackets travel between nodes over intra-firewall connections within the firewall model." *Id.* 4:35-39. The firewall determines whether to drop or allow packets based on the rules associated with the incoming and outgoing nodes. *E.g.*, EX1001, 8:5-25, 9:27-34. The specification further explains, "rules are applied to packets following on the connection between" the nodes. *Id.*, 9:38-39. Thus, the '282 addresses the behavior of the firewall itself and "the experience of a network packet as it travels through the firewall from its arriving node to its departing node"—*i.e.*, from its source to its destination. EX1001,

6:36-42. EX1003, ¶43.

Within the firewall itself, the rules are expressed as "dynamic chains of rules" which the specification defines as "serialized sequences of one or more rules." EX1001, 5:5-7; *see also id.*, 5:24-28 ("Chains of firewall rules are linear and serial, as the name implies."). While the specification provides an example of rule chains represented as leaves in a tree, the specification explains that the logic is far simpler: "Rule chains essentially represent predicate/antecedent rule logic and can be classified as classic production rules systems as known in the Artificial Intelligence community.... Said differently, rule chains represent classic "If-Then" logic, as might be supported by the syntax of a software programming language." EX1001, 6:48-59. Indeed, the specification admits that the "representative set of firewall rules" and disclosed "firewall functions are relatively standard and basic capabilities that can be found in most firewall implementations." *Id.*, 6:60-65. EX1003, ¶44.

Finally, the specification explains that, in addition to rules for how to treat incoming packets and rules for how to treat outgoing packets from the firewall, the firewall may include rule chains that have been "inserted and deleted dynamically." EX1001, 7:62-64. These rules are reached by "taps" in the main set of rules that indicate for the firewall to go to a different rule chain that has been "dynamically loaded as part of a firewalls configuration or runtime reconfiguration." *Id.*, 7:65-8:4. The specification does not provide any description on how these so-called

dynamic rules are generated or when/how they are added to the firewall's configuration (or re-configuration) file. *Id.*, EX1003, ¶45.

B. File History

U.S. Patent Application No. 13/092,488, which matured into the '282 patent, was filed 4/22/2011. In a series of § 103 rejections, the Examiner continuously maintained that the core elements of the initially submitted claims were disclosed in the prior art of record. Namely, the prior art already disclosed "a method and device for controlling data through a firewall...comprising: defining at least one node...associated with two or more network interfaces; associating a set of firewall rules with the at least one node ...and accepting or denying [a] packet based on the set of firewall rules" along with disclosure of "reconfiguring the firewall [] at runtime." *E.g.*, EX1002, [PDF547]. To overcome these rejections, Applicant amended its claims to require the firewall to be "dynamically self-configurable... during runtime without operation interaction." *Id.*, [PDF609]. This too, however, was found obvious in view of the prior art. *Id.*, [PDF633-635]. To secure allowance, Applicant amended the independent claims to specify that "the set of firewall rules comprises a plurality of chains of rules forming various paths through a hierarchical structure, and wherein the hierarchical structure comprises defined places for dynamically updating the set of firewall rules during runtime." *Id.*, [PDF663]. EX1003, ¶¶46-47.

C. POSITA

The relevant art for the '282 patent is the field of computer science generally, and specifically computer networking. A person of ordinary skill in that art, as of August 2004, would have been an individual with either (1) at least a bachelor's degree in computer science or computer engineering or an equivalent field plus at least one year of experience working on computer networking, or (2) at least 3 years of experience working on computer networking, even without a formal degree. EX1003, ¶¶16-20.

IV. CLAIM CONSTRUCTION

A claim is construed "using the same claim construction standard that would be used to construe the claim in a civil action," 37 C.F.R. § 42.100(b), which is governed by *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005). Under the standard set out in *Phillips*, this petition applies the plain and ordinary meaning to each term in the '936 Patent, which is "the meaning that the term would have to a [POSITA] in question at the time of the invention." *Id.* at 1313.

A. "wherein the set of firewall rules is dynamically self-configurable during runtime"

This term refers to "a set of firewall rules that are configured without any human operator interaction while the node is evaluating whether to accept or deny the packet." This is consistent with the file history. For example, this clause was used to distinguish prior art that allowed for "an advanced user to establish a firewall

policy for a computer." EX1002, [PDF617]. In contrast, the claimed process must be performed "during runtime" and "without operation interaction." *Id.* EX1003, ¶24.

B. "chains of rules forming various paths through a hierarchical structure"

This term refers to "a list of one or more linear and serialized sequence of firewall rules forming various paths through a hierarchical structure." For example, the '282 patent defines "chains of rules" as "serialized sequences of one or more rules" and "a list of one or more firewall rules... [that] are linear and serial." EX1001, 5:6-28. Likewise, the specification describes the rules as allowing a programming language to follow a path through a series of rules with names in a hierarchical name space. *Id.*, 6:36-59. EX1003, ¶25.

C. "defined places for dynamically updating the set of firewall rules during runtime"

This term refers to "one or more isolated locations within the sequence of firewall rules to add, remove, or change a rule while the node is evaluating whether to accept or deny the packet." For example, the specification explains that the "dynamic chains of rules....offer isolated, well-defined places for specific behavior to be introduced." EX1001, 5:5-12. Likewise, the specification distinguishes between rules added "as part of firewalls configuration [and] runtime reconfiguration." *Id.*, 7:62-8:4. In other words, rules added during runtime require

a "runtime reconfiguration"—*i.e.*, as the firewall is in operation processing packets—and not simply an initial configuration before the firewall is put into service. EX1003, ¶26.

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

V. Grounds of Unpatentability

The '282 patent is directed to a method and system for controlling a firewall with a rule-set that can be updated dynamically. Independent claim 1 (representative of the other independent claims) recites:

1. A method for controlling data through a firewall performed on at least one data controlling computer having computer instructions stored on at least one non-transitory computer readable medium, comprising:

defining at least one node, wherein the at least one node is associated with two or more network interfaces;

associating a set of firewall rules with the at least one node;

receiving a packet at a first node of the at least one node; and

accepting or denying the packet based on the set of firewall rules, wherein the set of firewall rules is dynamically self-configurable during runtime without operator interaction, wherein the set of firewall rules comprises a plurality of chains of rules forming various paths through a hierarchical structure, and wherein the hierarchical structure comprises defined places for dynamically updating the set of firewall rules during runtime.

At their core, the claims are directed to a basic firewall model that processes data through the firewall on a packet-by-packet model following a serialized

sequence of one or more rules that can be added to, or otherwise changed, during runtime without requiring human intervention. The claimed features are anticipated by, and at minimum obvious in view of the prior art, as explained below. This Petition is supported by the Declaration of John R. Black, Jr., which describes the scope and content of the prior art at the time of the alleged invention. EX1003, ¶¶34-155.

A. Grounds 1 and 2 – Claims 1-35 Are Anticipated by And At Minimum Obvious Over Coss

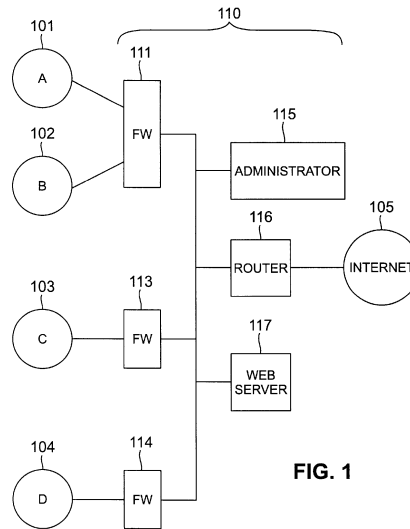
1. Overview of Coss

Like the '282 patent, **Coss** discloses a configurable firewall with dynamically adaptable rule chains. Specifically, **Coss** discloses an "improved computer network firewall" that "can support multiple security policies...by applying any one of several distinct sets of access rules." *Coss*, Abstract. "At the firewall, packets are inspected and filtered, i.e., passed on or dropped depending on whether they conform to a set of predefined access rules." *Id.*, 1:22-24. In addition, "[d]ynamic rules may be used in addition to pre-loaded access rules." *Id.*, Abstract. EX1003, ¶49.

Like the one or more nodes in the '282 patent, the firewall of **Coss** is designed to support "multiple security domains...each with a separate security policy." *Coss*, 3:35-40. For example, as illustrated in Figure 1, **Coss** discloses "four user sites...of corporations A through D, with firewall protection in their connections to the Internet." *Id.*, 3:43-57. The "firewall facility" includes one or more "firewall

processors," each configurable to protect one or more domains (or corporate "sites").

Id.



Coss, FIG. 1. EX1003, ¶50. As **Coss** teaches, firewall processors 113 and 114 are each configured to protect a single user site (sites C and D, respectively) while "[f]irewall processor 111 is configured to serve the two sites 101 and 102... implement[ing] separate firewall policies for each of the two sites vis-à-vis the Internet 105, as well as for communications between the two sites." *Id.*, 3:43-57. Likewise, a site may be broken into multiple "sub-sites," each protected by the firewall using separate security policies. *Id.*, 3:58-4:3, FIG. 2. Finally, **Coss** also discloses defining "host group[s]", allowing administrators to "add or drop different hosts" from the group without modifying other aspects of the rule sets. *Id.*, 2:41-46. EX1003, ¶51-53.

Also like the '282 patent, **Coss** discloses hierarchical chains of rules. In

particular, **Coss** discloses processing rules on a packet-by-packet basis. *Coss*, 1:61-2:6. According to **Coss**, these rules are implemented as distinct "sets of access rules which are represented in tabular form" and specify actions to be taken for a given packet being processed. *Coss*, 4:4-11. "[T]he rules are applied sequentially until a rule is found which is satisfied by the packet (or until the rule table is exhausted, in which case the packet is dropped)." *Id.*, 4:26-30. Like the "if-then" rules in the '282 patent, **Coss**'s rules implement a similar predicate/antecedent format: "For a packet to satisfy a rule, each condition included in the rule must be met." *Id.*, 4:30-31. Figure 3 shows such a rule table as implemented in **Coss**:

RULE NO.	SOURCE HOST	DEST. HOST	SERVICE	ACTION
10	A	B	FTP	PASS
20	A	*	*	DROP
30	B	C	TELNET	PROXY
40	*	D	MAIL	PASS

FIG. 3

Coss, FIG. 3. As **Coss** explains, "the categories 'Source Host,' 'Destination Host' and 'Service' impose conditions which must be satisfied by data included in a packet for the specified action to be taken on that packet." *Id.*, 4:13-16. And the specification

confirms that these categories are applied in a hierarchical manner—as illustrated in Fig. 5A-B, the rule processing requires examining the "Source" domain rules before examining the "Destination" rules, etc. *See, e.g., Id.*, FIGs. 5A-5B, 6:18-7:9 (describing searching the "rule set for the source domain" at step 504 followed by "rule set look-up for the destination domain" at step 507). EX1003, ¶¶54-56.

Finally, **Coss** teaches implementing its firewall using "[d]ynamic rules...which are included with the access rules as a need arises, for processing along with the access rules, e.g., by a rule processing engine." *Coss*, 8:28-30. Dynamic rules "can be loaded at any time by trusted parties" which includes "a trusted application, [and a] remote proxy." *Id.*, 8:30-35. Dynamic rules in **Coss** also can be removed from the rule set once it has served its function and, generally, "dynamic rules allow a given rule set to be modified based on events happening in the network without requiring that the entire rule set be reloaded." *Id.*, 8:35-40. EX1003, ¶57.

2. Independent Claim 1

- (a) Claim [1PRE]: "A method for controlling data through a firewall performed on at least one data controlling computer having computer instructions stored on at least one non-transitory computer readable medium, comprising:"

Coss discloses a method for controlling data through a firewall (*e.g.*, "information...transmitted in the form of packets" is "filter[ed] at a...firewall") performed on at least one data controlling computer (*e.g.*, "firewall for

controlling the flow of data") **having computer instructions stored on at least one non-transitory computer readable medium** (e.g., "computer system software...on general-purpose PC hardware"). EX1003, ¶64.

For example, **Coss** discloses an "improved computer network firewall" that "can support multiple security policies...by applying any one of several distinct sets of access rules." *E.g.*, Coss, Abstract; *see also id.*, 1:9-11 ("This invention relates to the prevention of unauthorized access in computer networks and, more particularly, to **firewall protection within computer networks.**"). The firewall is "implemented as computer system software...on general-purpose PC hardware." Coss, 3:25-35. Coss's firewall applies "a set of predefined access rules" to packets as they move through the firewall to inspect and filter (*i.e.*, pass or drop) the packets as necessary. Coss, 1:14-26. EX1003, ¶65.

(b) Claim [1A]: "defining at least one node, wherein the at least one node is associated with two or more network interfaces;"

Coss discloses, and at minimum renders obvious, defining at least one node (e.g., "domain", "[sub-]site", "LAN", "subnets"), **wherein the at least one node is associated with two or more network interfaces** (e.g., "each domain is associated with one or more network interfaces", "[incoming/outgoing] network interface"). EX1003, ¶66.

Coss discloses the firewall "support[s] multiple security domains...each with

a separate security policy" and "in the firewall, each domain is associated with one or more network interfaces." Coss, 3:36-4:3, 6:18-28. For example, Coss Figure 1 "shows four user sites 101-104, e.g., of corporations A through D, with firewall protection in their connections to the Internet 105. Such protection is provided by a firewall facility, here in the form of a LAN 110, including firewall processors 111, 113 and 114, an administrator processor 115, a router 116 and a web server 117." Coss, 3:36-4:3. Likewise, Coss discloses that the firewall can protect "sub-sites" with each also having its own security policy. *Id.* As Coss explains, each "firewall processor" can serve one or more domains—e.g., "firewall processor 111 is configured to serve the two sites 101 and 102." *Id.* Conceptually, because the firewall is designed "support multiple security domains" with "separate security polic[ies]," a POSITA would therefore have understood Coss to disclose, and at minimum render obvious, defining the domains—the firewall entry and departure networks—that are protected by the firewall. EX1003, ¶¶67-68. For example, the Figure 6 "domain table" confirms that the firewall must define Domain A, Domain B, etc., to associate the respective IP range to that Domain.

INTERFACE	ADDRESS RANGE	DOMAIN
0	10.50.0.0 - 10.50.255.255	A
0	10.60.0.0 - 10.60.255.255	B
1	*	C
2	*	*

FIG. 6

Coss, FIG. 6; *see also id.*, 7:10-15 ("For convenient linking of each network interface to a domain, a domain table is used. In cases where an interface is shared by multiple domains, an address range is included. This is illustrated by FIG. 6 which shows non-overlapping address ranges."); *see also id.*, 1:61-2:6, 3:25-35, 6:33-37. EX1003, ¶¶67-68.

(c) Claim [1B]: " associating a set of firewall rules with the at least one node;"

Coss discloses, and at minimum renders obvious, associating a set of firewall rules ("distinct set of access rules";) with the at least one node ("domain"). EX1003, ¶69.

Coss discloses that "firewall 110" can support multiple domains by "applying any one of several distinct sets of access rules for a given packet" depending on the domains being crossed. *E.g.*, Coss, 1:61-2:6 ("[C]omputer network firewall is able to support (a) multiple security policies, (b) multiple users, or (c) multiple security

policies as well as multiple users, by applying any one of several distinct sets of access rules for a given packet. The particular rule set that is applied for any packet can be determined based on information such as the incoming and outgoing network interfaces as well as the network source and destination addresses.". For example, when processing a packet the firewall identifies the "rule set" being used based on the source and/or destination domains. *Id.* As Coss explains, "security polices can be represented by sets of access rules which are represented in tabular form and which are loaded into the firewall..." Coss, 4:4-19. The disclose rule table "can provide for categories, including rule number, designations of source and destination hosts, a designation of a special service which can be called for in a packet, and a specification of an action to be taken on a packet." *Id.*

RULE NO.	SOURCE HOST	DEST. HOST	SERVICE	ACTION
10	A	B	FTP	PASS
20	A	*	*	DROP
30	B	C	TELNET	PROXY
40	*	D	MAIL	PASS

FIG. 3

Coss, FIG. 3. "In FIG. 3, the categories 'Source Host,' 'Destination Host' and

'Service' impose conditions which must be satisfied by data included in a packet for the specified action to be taken on that packet." Coss, 4:4-19. EX1003, ¶¶70-71.

Moreover, as Coss explains, processing packets "includes determining the domains which the packet is to cross, examining the applicable rules to ascertain whether the packet may pass, and determining whether any special processing is required." *E.g.*, Coss, 6:18-28; *see also id.* 6:29-65 (disclosing steps for determining the "source domain" (step 503) and then searching "the rule set for the source domain" (step 504), in addition to determining the "destination domain" (step 506) and then performing "rule set look up for the destination domain" (step 507)). *See also* Coss at 2:33-46, 3:36-4:3, 7:10-15. A POSITA thus would have understood and at minimum found obvious that applying access rules to packets based on the domains necessarily requires associating the distinct sets of rules to the appropriate domains. EX1003, 72.

(d) Claim [1C]: "receiving a packet at a first node of the at least one node; and"

Coss discloses receiving a packet at a first node of the at least one node.

EX1003, ¶73.

For example, Coss discloses firewall 110 processing packets as they are received for, and from, each domain A-D. *E.g.*, Coss, 6:29-30 ("501: an IP packet is received by the firewall at an interface."). The firewall then implements the appropriate rule set based on the domains "which the packet is to cross." *E.g.*, Coss,

6:18-28 ("FIGS. 5A and 5B illustrate over-all flow for packet processing by a firewall which supports multiple domains. Such processing includes determining the domains which the packet is to cross, examining the applicable rules to ascertain whether the packet may pass, and determining whether any special processing is required. In the firewall, each domain is associated with one or more network interfaces. Interfaces that support more than one domain are separated using an IP address range to distinguish the packets."). A POSITA would have understood this necessarily requires receiving the packet at the domain. EX1003, ¶74. *See also*, Coss at 7:64-65, 9:33, 9:52-53.

- (e) Claim [1D]: "accepting or denying the packet based on the set of firewall rules, wherein the set of firewall rules is dynamically self-configurable during runtime without operator interaction, wherein the set of firewall rules comprises a plurality of chains of rules forming various paths through a hierarchical structure, and wherein the hierarchical structure comprises defined places for dynamically updating the set of firewall rules during runtime."

Coss discloses, and at minimum renders obvious, accepting (e.g., "pass") **or denying** (e.g., "drop") **the packet based on the set of firewall rules, wherein the set of firewall rules is dynamically self-configurable during runtime without operator interaction** (e.g., "dynamic rule[]" which acts to alter the operation of the...initial set of rules under specified conditions"; "dynamic rules...can be loaded at any time by...a trusted application"), **wherein the set of firewall rules comprises**

a plurality of chains of rules forming various paths through a hierarchical structure (e.g., "rules are applied sequentially"), and wherein the hierarchical structure comprises defined places for dynamically updating the set of firewall rules during runtime (e.g., at step 1012 of FIG 10's packet processing process).
EX1003, ¶75.

Coss discloses a plurality of chains of rules forming various paths through a hierarchical structure and using these rules to accept or deny packets at the firewall. For example, Coss implements its firewall policies as "sets of access rules" stored in a "rule table" that must be "applied sequentially until a rule is found" (*i.e.*, linearly and serialized) or the packet is dropped. E.g., Coss, 4:4-19 ("The security policies can be represented by sets of access rules which are represented in tabular form... As illustrated in FIG. 3, such a table can provide for categories including rule number, designations of source and destination hosts, a designation of a special service which can be called for in a packet, and a specification of an action to be taken on a packet... In FIG. 3, the categories "Source Host," "Destination Host" and "Service" impose conditions which must be satisfied by data included in a packet for the specified action to be taken on that packet..."), 4:27-37 ("In rule processing for a packet, the rules are applied sequentially until a rule is found which is satisfied by the packet (or until the rule table is exhausted, in which case the packet is dropped). For a packet to satisfy a rule, each condition included in the rule must be met. For

example, with reference to FIG. 3, a packet from source host A to destination host D and representing mail will be dropped under Rule 20...). As illustrated in Figures 3 and 5A-B, the rule table includes the rule in a hierarchical structure as a chain of rules that follow various paths:

RULE NO.	SOURCE HOST	DEST. HOST	SERVICE	ACTION
10	A	B	FTP	PASS
20	A	*	*	DROP
30	B	C	TELNET	PROXY
40	*	D	MAIL	PASS

FIG. 3

Cross, FIG. 3. For example, a packet originating from Domain A may follow a different path through a different series of rules than a packet arriving from Domain B, etc., thus comprising a plurality of chains of rules forming various paths through a hierarchical structure. EX1003, ¶76. Specifically, the flowchart of Fig. 5A-B confirms that first rules associated with the source domain are checked; then rules associated with the destination domain, etc, and based on these rules "pass[ing]" or "drop[ping]" the packet:

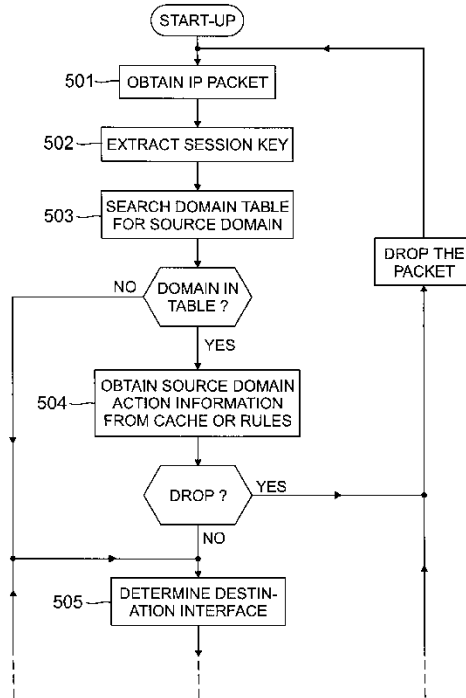


FIG. 5A

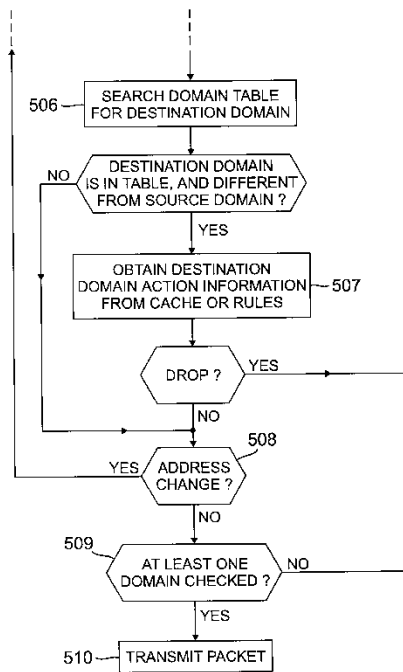


FIG. 5B

Coss, FIGs. 5a-5B; *see also id.*, 6:18-7:9 ("FIGS. 5A and 5B illustrate over-all flow

for packet processing by a firewall which supports multiple domains. ...503: on the basis of which interface received the packet and the source IP address of the received packet, the source domain is determined...; 504: ...the rule set for the source domain is searched for a match; if a match is found in the rules and if the action is not "drop," the process continues with step 505; if a match is found in the rules and the action is "drop," ..., the packet is dropped, and the process returns to step 501; if no match is found in the rules, the packet is dropped and the process returns to step 501;...506: using the destination interface and the destination address of the packet, the destination domain is determined; if the destination domain is not found, or if the destination domain matches the domain just checked, the process skips to step 508; 507: ..., rule set look-up for the destination domain are carried out in a manner analogous to that employed for the source domain in step 504;...509: if the packet was not processed with respect to any domain, the packet can be dropped, as a firewall owner has no interest in supporting communications between interfaces which are not subject to any access rules; 510: with all actions having resulted in "pass," the packet is sent out the appropriate network interface."). Further, as Coss explains, "[t]he particular rule set that is applied for any packet can be determined based on information such as the incoming and outgoing network interfaces as well as the network source and destination addresses." *E.g.*, Coss, 1:61-2:6. **Coss** thus expressly discloses applying a particular

set of rules based on, *e.g.*, the source domain, which confirms the firewall rules as a plurality of chains with a hierarchical structure. EX1003, ¶76.

Coss further discloses the set of firewall rules is dynamically self-configurable during runtime without operator interaction. EX1003, ¶77.

In particular, **Coss** discloses implementing a set of dynamically adaptable rules "which acts to alter the operation of [an]...initial set of rules under specified conditions." *E.g.*, Coss, Claim 35. "A method for providing a firewall service in a computer network, comprising the steps of: forming an augmented set of rules by including, in an already-loaded initial set of access rules, at least one dynamic rule which acts to alter the operation of the already-loaded initial set of rules under specified conditions without reloading at least one unaltered rule of the already-loaded set of access rules; and using the augmented set of rules in validating a packet; *wherein the at least one rule is a dynamic rule...*").

Coss's dynamic rules "are included with the access rules as a need arises" and "can be loaded at any time by trusted parties, *e.g.*, a trusted application," as distinguished from a "firewall administrator," confirming that the dynamic rules may be added during runtime without operator interaction. *E.g.*, Coss, 8:27-59 ("Dynamic rules are rules which are included with the access rules as a need arises, for processing along with the access rules, *e.g.*, by a rule processing engine.... They can be loaded at any time by trusted parties, *e.g.*, a trusted application, remote proxy

or firewall administrator, to authorize specific network sessions.... Once a dynamic rule has served its function, it can be removed from the rule set. The dynamic rules allow a given rule set to be modified based on events happening in the network without requiring that the entire rule set be reloaded..." *E.g.*, Coss, 8:27-59; *see also id.*, 2:33-46. A POSITA would have understood and at minimum found obvious that—by separately enumerating a computer "application" and a human "administrator"—**Coss** discloses loading the dynamic rules both with, and without, human intervention. Further, that these rules "allow a given rule set to be *modified based on events happening in the network*" (*id.*) and that they may be loaded during the processing of a given packet (*e.g.*, at step 1012 of FIG. 10), confirms that Coss's rules are dynamically self-configurable during runtime. *See, e.g.*, Coss, 9:28-10:21 (describing the processing of packets in FIGs. 10A-10B using a proxy reflection where "dynamic rules can be used as described below" including "1012: the firewall loads a dynamic rule to perform this action; 1013: the remote proxy sends the packet to the firewall; based on the dynamic rule loaded in step 1012").

Finally, **Coss** discloses packet processing steps by, for example, a remote proxy firewall, that includes **defined places** where dynamic rules may be loaded in the process. *Id.* (disclosing the steps in the packet processing flow where a dynamic rule may be added.). In particular, **Coss** describes an exemplary flow for processing packets that includes at least one isolated location within the sequence of firewall

rules to "load[] a dynamic rule"—namely, at step 1012 of the packet processing flow.

Id. As **Coss** explains, after a packet is received by a firewall (step 1001), the "action associated with the packet is determined by looking...in the appropriate rule set" (step 1002) and "if the action indicates a remote proxy...the packet is routed to the remote proxy server" (steps 1004-1005) where, at step 1012, "the firewall loads a dynamic rule." **Coss**, 9:28-10:21. EX1003, ¶77.

See also **Coss**, 4:21-26, 2:7-20, 2:21-32, 5:38-52, 7:15-27.

3. **Claim [2]: "A method according to claim 1, wherein dynamically updating the set of firewall rules during runtime further comprises, during runtime, adding a rule to the set of firewall rules, deleting a rule from the set of firewall rules, or modifying a rule in the set of firewall rules without operator interaction."**

Coss discloses, and at minimum renders obvious, a method according to **claim 1, wherein dynamically updating the set of firewall rules during runtime further comprises, during runtime** (*e.g.*, when a packet is being processed by a proxy application), **adding a rule to the set of firewall rules** (*e.g.*, "load[ing]"/"add[ing]" a rule), **deleting a rule from the set of firewall rules** (*e.g.*, "remov[ing]" a rule), **or modifying a rule in the set of firewall rules without operator interaction.** EX1003, ¶78.

As discussed in [1D], **Coss** discloses that the dynamic rules "can be loaded at any time...without requiring that the entire rule set be reloaded" (including, as illustrated in FIG. 10 *during* the processing of a given packet's traversal of the

firewall). *E.g.*, Coss, 8:28-40 ("Dynamic rules are rules which are included with the access rules as a need arises,.... They can be loaded at any time by trusted parties, e.g., a trusted application, remote proxy or firewall administrator, to authorize specific network sessions. A dynamic rule can be set for single-session use, or its use can be limited as to time. Once a dynamic rule has served its function, it can be removed from the rule set. The dynamic rules allow a given rule set to be modified based on events happening in the network without requiring that the entire rule set be reloaded."). Coss further discloses that the dynamic rules "can be removed from the rule set," and that these "dynamic rules allow a given rule set to be modified based on events happening in the network." *Id.*; *see also id.*, 2:33-46. Further, the dynamic rules can be implemented "to alter the operation of the rule without changing the rule itself." Coss, Claim 22; *see also id.*, 8:41-59, 10:7-9. EX1003, ¶79.

4. **Claim [3]: "A method according to claim 1, wherein associating the set of firewall rules with the at least one node further comprises associating a first subset of the set of firewall rules with the first node."**

See [1]. Coss discloses, and at minimum renders obvious, a method according to claim 1, wherein associating the set of firewall rules with the at least one node further comprises associating a first subset of the set of firewall rules with the first node. EX1003, ¶80.

As discussed above in [1], Coss discloses implementing different security

policies for each node (*e.g.*, for each "domain" connected to the firewall) and further associating sets of rules with source domains and destination domains. *Coss*, 3:36-4:3 ("With a capability for supporting multiple security domains, a single firewall can support multiple users, each with a separate security policy. Also, as different security policies can apply for communications between sub-sites, such a capability can be used within a site. Respective configurations are illustrated by FIGS. 1 and 2. ¶ FIG. 1 shows four user sites 101-104, *e.g.*, of corporations A through D, with firewall protection in their connections to the Internet 105.... ¶ FIG. 2 shows a user site 201 connected to the Internet 105 via a firewall processor 211.") A POSITA thus would have understood that **Coss** implicitly discloses associating a first subset of the set of firewall rules (*e.g.*, only rules applicable to Domain A) to the first node (*e.g.*, domain A). EX1003, ¶81.

RULE NO.	SOURCE HOST	DEST. HOST	SERVICE	ACTION
10	A	B	FTP	PASS
20	A	*	*	DROP
30	B	C	TELNET	PROXY
40	*	D	MAIL	PASS

FIG. 3

5. **Claim [4]: "A method according to claim 3, wherein the at least one node further comprises at least two nodes including a second node, further comprising associating a second subset of the set of firewall rules with the second node."**

See [1], [3]. **Coss** discloses, and at minimum renders obvious, a method according to claim 3, wherein the at least one node further comprises at least two nodes including a second node, further comprising associating a second subset of the set of firewall rules with the second node. EX1003, ¶82.

As discussed above in [1], **Coss** discloses multiple nodes (e.g., Domains A-D) and the firewall 110 is configured to apply different security policies for each domain. *Coss*, 3:36-4:3. A POSITA thus would have understood that **Coss** implicitly discloses associating security policies applicable to domain A at the first node (domain A) and security policies applicable to domain B (*i.e.*, a second subset of rules) at the second node (domain B.). EX1003, ¶83.

RULE NO.	SOURCE HOST	DEST. HOST	SERVICE	ACTION
10	A	B	FTP	PASS
20	A	*	*	DROP
30	B	C	TELNET	PROXY
40	*	D	MAIL	PASS

FIG. 3

6. **Claim [5]: "A method according to claim 4, wherein the first subset of firewall rules is the same as or at least partially different from the second subset of firewall rules."**

See [1], [3], [4]. **Coss discloses, and at minimum renders obvious, a method according to claim 4, wherein the first subset of firewall rules is the same as or at least partially different from the second subset of firewall rules.**

EX1003, ¶84.

A POSITA would have understood that any two sets of rules are necessarily either the same, or *at least partially* (if not completely) different, so **Coss** discloses this. EX1003, ¶85. In particular, **Coss** discloses rules applicable both to domains A and B (e.g., rule 10), rules applicable to only A (rule 20), rules applicable to B and C (rule 30), and rules applicable to only D (rule 40). A POSITA thus would have understood **Coss** to disclose wherein the first subset of rules (rules applicable to domain A) are at least partially different from the second set of firewall rules—*i.e.*, rules 10 and 20 are associated with domain A, while rules 10 and 30 are associated with domain B. *Id.*

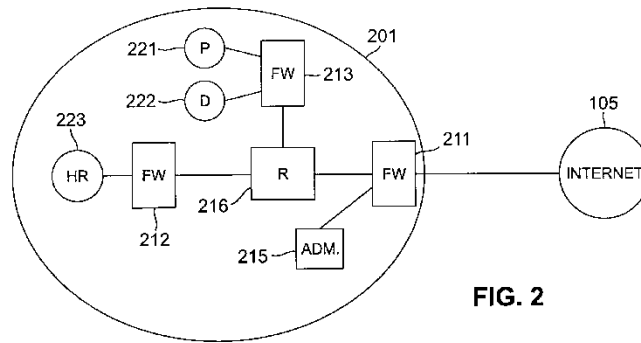
RULE NO.	SOURCE HOST	DEST. HOST	SERVICE	ACTION
10	A	B	FTP	PASS
20	A	*	*	DROP
30	B	C	TELNET	PROXY
40	*	D	MAIL	PASS

FIG. 3

7. **Claim [6]: "A method according to claim 4, wherein one of the first subset of firewall rules or second subset of firewall rules equals the entire set of firewall rules."**

See [1], [3], [4]. Coss discloses, and at minimum renders obvious, a method according to claim 4, wherein one of the first subset of firewall rules or second subset of firewall rules equals the entire set of firewall rules. EX1003, ¶86.

In addition, Coss discloses its firewall can be configured to act as corporate boundary to the Internet, while also protecting internal communications between "sub-sites," as illustrated in Fig. 2:



Coss, FIG. 2; *see also id.*, 3:58-4:3 ("FIG. 2 shows a user site 201 connected to the Internet 105 via a firewall processor 211. An administrator processor 215 and a router 216 are connected to the firewall processor 211. The router 216 is connected to additional firewall processors 212 and 213 which are internal to the user site 201. The firewall processor 212 protects a single sub-site 223, such as Human Resources (HR). The firewall processor 213 is configured for protecting two sub-sites, such as Payroll (P) and Disbursements (D), vis-a-vis the remainder of the site 201 as well as with respect to communications between sub-sites 221 and 222..."). EX1003, ¶87.

As Coss discloses, the "firewall is able to support...multiple security policies as well as multiple users, by applying any one of several distinct sets of access rules for a given packet. The particular rule set that is applied for any packet can be determined based on information such as the incoming and outgoing network interfaces as well as the network source and destination addresses." E.g., Coss, 1:61-2:6. EX1003, ¶88.

With the embodiment of Figure 2, insofar as "firewall processor 211" is

positioned to protect the entire site—*i.e.*, all the subsites, collectively—from the Internet, a POSITA would have understood and at minimum found obvious to associate "firewall processor 211" with the entire set of firewall rules to ensure that no communications between a sub-site and the Internet are able to pass without being inspected and acted on appropriately. EX1003, ¶89.

8. **Claim [7]: "A method according to claim 1, further comprising, after receiving the packet and prior to accepting or denying the packet, conditioning the packet based on the set of firewall rules."**

Coss discloses a method according to claim 1, further comprising, after receiving the packet and prior to accepting or denying the packet, conditioning the packet (*e.g.*, "processing"; "address change"; "network address translation", "encryption") based on the set of firewall rules. EX1003, ¶90.

Coss discloses further conditioning packets during rule processing when the rules include "a designation of a special service which can be called for in a packet, and a specification of an action to be taken on a packet. Special services can include...network address translation, and encryption, for example..." Coss, 4:4-19. In addition, rules may "call[]" for an address change, *e.g.*, to a proxy or for insertion of one packet into another ("tunnel option")," in which case, "the packet's destination address is replaced with the address of the remote proxy [and] the destination port can be changed as well." Coss, 6:66-7:3; 9:42-48. In these cases, "the original packet header data is recorded in the session cache along with any

changed values." *Id.* EX1003, ¶¶91-93.

9. **Claim [8]: "A method according to claim 7, wherein conditioning the packet based on the set of firewall rules further comprises rewriting a portion of a network packet header associated with the packet."**

See [7]. **Coss discloses a method according to claim 7, wherein conditioning the packet based on the set of firewall rules further comprises rewriting (e.g., "changing") a portion of a network packet header associated with the packet (e.g., "source port", "destination port" "destination address", "source address").** EX1003, ¶94.

As discussed in [7], **Coss** discloses performing "address change[s]" and "Network Address Translation"³ (*i.e.*, NAT), both of which require rewriting a portion of the network packet header associated with that packet. *See* Coss, 6:66-7:3 ("if a rule that applies to the packet calls for an address change"), 8:61-9:9 (disclosing proxy reflection in which "the firewall replaces the destination address in the packet with the host address of the proxy application..."), 9:42-48 ("the packet's destination address is replaced with the address of the remote proxy; if configured, the destination port can be changed as well" and "the original packet

³ Indeed, the '282 patent itself confirms that "conditioning" packets includes performing NAT, *i.e.*, "hav[ing] source and or/or destination addresses translated to/from an internal network address that is not publicly available on the general network (or Internet)." EX1001, 6:18-28, 6:60-7:12.

header data is recorded...along with any changed values."). Further, Coss's disclosure that "IP header values are changed back to the original values" after passing through the firewall (Coss, 10:10-17) confirms that Coss's conditioning involves changing (*i.e.*, rewriting) a portion of the header *before* the packet passed through the entire firewall—*i.e.*, before the firewall determined whether the packet should be allowed or dropped. EX1003, ¶¶95-96.

10. Claim [9]: "A method according to claim 1, wherein each of the at least one node is associated with at least two network interfaces."

Coss renders obvious a method according to claim 1, wherein each of the at least one node is associated with at least two network interfaces. EX1003, ¶97.

See [1a]. As a POSITA would recognize, the fact that a "node is associated with two or more network interfaces" as recited in claim 1, necessarily means that it is associated with "at least" two network interfaces. EX1003, ¶98.

11. Claim [10]: "A method according to claim 1, wherein each of the two or more network interfaces is connected with at least one physical device."

Coss renders obvious a method according to claim 1, wherein each of the two or more network interfaces is connected with at least one physical device. EX1003, ¶99.

See [1a].

As discussed above in [1a], **Coss** discloses using the firewall facility to protect multiple domains (i.e., nodes). *E.g.*, Coss, 3:43-57 ("FIG. 1 shows four user sites 101-104, e.g., of corporations A through D, with firewall protection in their connections to the Internet 105...."), 3:58-4:3 ("FIG. 2 shows a user site 201 connected to the Internet 105 via a firewall processor 211. An administrator processor 215 and a router 216 are connected to the firewall processor 211. The router 216 is connected to additional firewall processors 212 and 213 which are internal to the user site 201. The firewall processor 212 protects a single sub-site 223, such as Human Resources (HR)...."). Coss further discloses that the domains are identified "based on the incoming or outgoing network interface[s]" and that "each domain is associated with one or more network interfaces" which are physical devices. Coss, 6:3-25 In particular, In addition, **Coss** discloses that an "incoming or outgoing network interface may be in the form of a network interface card (NIC) e.g., an Intel EtherExpress Pro 100B card available from Intel Corporation." Coss, 6:3-17. EX1003 ¶¶100-101.

12. **Claim [11]: "A method according to claim 1, wherein the set of firewall rules being dynamically self-configurable further comprises dynamically updating the set of firewall rules during runtime without operator interaction."**

Coss discloses, and at minimum renders obvious, a method according to claim 1, wherein the set of firewall rules being dynamically self-configurable further comprises dynamically updating the set of firewall rules during runtime

without operator interaction. EX1003, ¶102.

See [1D], [2]. As discussed above, **Coss** discloses a method of implementing a firewall including dynamically updating the rule set "at any time by trusted parties," including by "a trusted application" *or* a "firewall administrator." **Coss**, 8:28-40. A POSITA would have understood that, by separately disclosing a "trusted application" to a (presumably human) "administrator," **Coss** thus discloses performing the dynamic rule update "without operator interaction." EX1003, ¶103. *See also* **Coss** at, e.g., Claim 35, 2:33-46, 9:66-10:21.

13. Independent Claim 12

- (a) Claim [12PRE]: "A device for controlling data through a firewall, comprising:"

See [1]. EX1003, ¶104.

- (b) Claim [12A]: "a plurality of network interfaces, wherein each of the plurality of network interfaces is operable to utilize one or more physical devices;"

Coss discloses, and at minimum renders obvious, a plurality of network interfaces, wherein each of the plurality of network interfaces is operable to utilize one or more physical devices. EX1003, ¶105.

See [10].

As discussed above in [1a] and [10], **Coss** discloses using the firewall facility to protect multiple domains (i.e., nodes). **Coss** at, e.g., 3:36-4:3. In addition, **Coss** discloses that an "incoming or outgoing network interface may be in the form of a

network interface card (NIC) e.g., an Intel EtherExpress Pro 100B card available from Intel Corporation." Coss, 6:14-17. EX1003, ¶106.

- (c) Claim [12B]: "a first computer readable storage medium storing a set of firewall rules, wherein the set of firewall rules is dynamically self-configurable during runtime without operator interaction, wherein the set of firewall rules comprises a plurality of chains of rules forming various paths through a hierarchical structure, and wherein the hierarchical structure comprises defined places for dynamically updating the set of firewall rules during runtime without operator interaction;"

See [1B], [1D]. EX1003, ¶107.

- (d) Claim [12C]: "a data controlling computer program comprising data controlling computer program code stored on either the first computer readable storage medium or on a second computer readable storage medium, the data controlling computer program code being executable to:"

See [1PRE]. EX1003, ¶108.

- (e) Claim [12C(i)]: "define at least one node, wherein the at least one node is associated with two or more network interfaces of the plurality of network interfaces; and"

See [1A], [1PRE]. EX1003, ¶109.

- (f) Claim [12C(ii)]: "when a packet is received at one of the two or more network interfaces associated with the at least one node, accept or deny the packet based on a review of the set of firewall rules."

See [1C], [1D]. EX1003, ¶110.

14. **Claim [13]:** "A device according to claim 12, wherein the data controlling computer program code is further executable to, while the firewall is processing traffic through the at least one node, add a rule to the set of firewall rules, delete a rule from the set of firewall rules, or modify a rule in the set of firewall rules."

See [2]. EX1003, ¶111.

15. **Claim 14**

- (a) Claim [14PRE]: "A device according to claim 12, wherein the at least one node comprises a first node, and wherein the data controlling computer program code is further executable to:"

See [3]. EX1003, ¶112.

- (b) Claim [14A]: "associate a first subset of the set of firewall rules with the first node; and"

See [3]. EX1003, ¶113.

- (c) Claim [14B]: "if the packet is received at the first node, apply the first subset of the set of firewall rules associated with the first node."

See [3]. EX1003, ¶115.

16. **Claim 15**

- (a) Claim [15PRE]: "A device according to claim 14, wherein the at least one node further comprises at least two nodes including a second node, and wherein the data controlling computer program code is further executable to:"

See [4]. EX1003, ¶115.

- (b) Claim [15A]: "associate a second subset of the set of firewall rules with the second node; and"

See [4]. EX1003, ¶116.

- (c) Claim [15B]: "if the packet is received at the second node, apply the second subset of the set of firewall rules associated with the second node."

See [4]. EX1003, ¶117.

17. **Claim [16]: "A device according to claim 15, wherein the first subset of the set of firewall rules is the same as or different from the second subset of the set of firewall rules."**

See [5]. EX1003, ¶118.

18. **Claim [17]: "A device according to claim 15, wherein either the first subset of the set of firewall rules or the second subset of the set of firewall rules equals the entire set of firewall rules."**

See [6]. EX1003, ¶119.

19. **Claim [18]: "A device according to claim 12, wherein the data controlling computer program code is further executable to condition the packet based on the set of firewall rules."**

See [7]. EX1003, ¶120.

20. **Claim [19]: "A device according to claim 18, wherein conditioning the packet based on the set of firewall rules further comprises rewriting a portion of a network packet header associated with the packet."**

See [8]. EX1003, ¶121.

21. **Claim [20]: "A device according to claim 12, wherein each of the at least one node is associated with at least two network interfaces."**

See [9]. EX1003, ¶122.

22. **Claim [21]: "A device according to claim 12, wherein each of the plurality of network interfaces is connected with at least one physical device."**

See [10]. EX1003, ¶123.

23. **Claim [22]: "A device according to claim 12, wherein each of the plurality of network interfaces is physically connected to every other network interface of the plurality of network interfaces and wherein physical connection between the plurality of network interfaces comprises indirect physical connection between the plurality of network interfaces."**

Coss discloses, and at minimum renders obvious, a device according to claim 12, wherein each of the plurality of network interfaces is physically connected to every other network interface of the plurality of network interfaces and wherein physical connection between the plurality of network interfaces comprises indirect physical connection between the plurality of network interfaces. EX1003, ¶124.

Coss discloses firewall facilities designed to protect "user sites" and "sub-sites" with each site connected via direct and/or indirect connections through, for example, a router. *E.g.*, Coss, 3:43-57 ("FIG. 1 shows four user sites 101-104, e.g., of corporations A through D, with firewall protection in their connections to the Internet 105. Such protection is provided by a firewall facility, here in the form of a LAN 110, including firewall processors 111, 113 and 114, an administrator processor 115, a router 116 and a web server 117...."), 3:58-4:3 ("FIG. 2 shows a user site 201 connected to the Internet 105 via a firewall processor 211. An

administrator processor 215 and a router 216 are connected to the firewall processor 211. The router 216 is connected to additional firewall processors 212 and 213 which are internal to the user site 201. The firewall processor 212 protects a single sub-site 223, such as Human Resources (HR). The firewall processor 213 is configured for protecting two sub-sites, such as Payroll (P) and Disbursements (D), vis-a-vis the remainder of the site 201 as well as with respect to communications between sub-sites 221 and 222."). At minimum, a POSITA would have found it obvious that a firewall protecting multiple domains (each with user interfaces) must be connected directly or indirectly to each such interface and, in turn each network interface must be at least indirectly connected to each other interface in order to properly pass packets through the network. EX1003, ¶¶125-127.

- 24. Claim [23]: "A device according to claim 12, wherein the data controlling computer program code is further executable to dynamically update the set of firewall rules during runtime without operator interaction."**

See [11]. EX1003, ¶128.

25. Independent Claim 24

- (a) Claim [24PRE]: "A data controlling computer program product comprising computer instructions stored on at least one non-transitory computer readable medium, wherein the computer instructions are operable when executed by at least one processor to:"

See [1PRE]. EX1003, ¶129.

- (b) Claim [24A]: "define at least one node for controlling data through a firewall, wherein at least one of the at least one node is associated with two or more network interfaces;"

See [1A]. EX1003, ¶130.

- (c) Claim [24B]: "associate a set of firewall rules with the at least one node, wherein the set of firewall rules further comprises a plurality of chains of rules forming various paths through a hierarchical structure, and wherein the hierarchical structure comprises defined places for dynamically updating the set of firewall rules during runtime without operator interaction;"

See [1B] and [1D]. EX1003, ¶131.

- (d) Claim [24C]: "receive a packet at a first node of the at least one node; and"

See [1C]. EX1003, ¶132.

- (e) Claim [24D]: "accept or deny the packet based on a review of the set of firewall rules, wherein the set of firewall rules is dynamically self-configurable during runtime without operator interaction."

See [1D]. EX1003, ¶133.

- 26. Claim [25]: "A data controlling computer program product according to claim 24, wherein the computer instructions are further executable to dynamically update the set of firewall rules during runtime without operator interaction."**

See [2]. EX1003, ¶134.

27. **Claim [26]:** "A data controlling computer program product according to claim 25, wherein the computer instructions are further executable to, while the firewall is processing traffic through the at least one node, add a rule to the set of firewall rules, delete a rule from the set of firewall rules, or modify a rule in the set of firewall rules."

See [2]. EX1003, ¶135.

28. **Claim 27**

- (a) Claim [27PRE]: "A data controlling computer program product according to claim 26, wherein the computer instructions are further executable to:"

See [3]. EX1003, ¶136.

- (b) Claim [27A]: "associate a first subset of the set of firewall rules with the first node; and"

See [3]. EX1003, ¶137.

- (c) Claim [27B]: "if the packet is received at the first node, apply the first subset of the set of firewall rules associated with the first node."

See [3]. EX1003, ¶138.

29. **Claim 28**

- (a) Claim [28PRE]: "A data controlling computer program product according to claim 27, wherein the at least one node further comprises at least two nodes including a second node, and wherein the computer instructions are further executable to:"

See [4]. EX1003, ¶139.

- (b) Claim [28A]: "associate a second subset of the set of firewall rules with the second node; and"

See [4]. EX1003, ¶140.

- (c) Claim [28B]: "if the packet is received at the second node, apply the second subset of the set of firewall rules associated with the second node."

See [4]. EX1003, ¶141.

- 30. Claim [29]: "A data controlling computer program product according to claim 28, wherein the first subset of the set of firewall rules is the same as or different from the second subset of the set of firewall rules."**

See [5]. EX1003, ¶142.

- 31. Claim [30]: "A data controlling computer program product according to claim 28, wherein either the first subset of the set of firewall rules or the second subset of the set of firewall rules equals the entire set of firewall rules."**

See [6]. EX1003, ¶143.

- 32. Claim [31]: "A data controlling computer program product according to claim 24, wherein the computer instructions are further executable to condition the packet based on the set of firewall rules."**

See [7]. EX1003, ¶144.

- 33. Claim [32]: "A data controlling computer program product according to claim 31, wherein conditioning the packet based on the set of firewall rules further comprises rewriting a portion of a network packet header associated with the packet."**

See [8]. EX1003, ¶145.

- 34. Claim [33]: "A data controlling computer program product according to claim 24, wherein each of the at least one node comprises at least two network interfaces."**

See [9]. EX1003, ¶146.

35. **Claim [34]: "A data controlling computer program product according to claim 24, wherein each of the two or more network interfaces is connected with at least one physical device."**

See [10]. EX1003, ¶147.

36. **Claim [35]: "A data controlling computer program product according to claim 24, wherein each of the two or more network interfaces is physically connected to every other network interface of the two or more network interfaces and wherein physical connection between the two or more network interfaces comprises indirect physical connection between the two or more network interfaces."**

See [22]. EX1003, ¶148

B. Ground 3 – Claims 1-35 Are Obvious over Coss In View of Ke

As discussed above, Coss discloses and at minimum renders obvious "defining at least one node, wherein the at least one node is associated with two or more network interfaces." *Supra*, §**Error! Reference source not found.** [b]. To the extent it is argued Coss does not disclose *defining* a node with at least two interfaces requires express disclosure of such configuration steps, **Ke** discloses this limitation. **Ke discloses defining at least one node** (e.g., "configuring an Internet security system"; "VLAN"/"VPN") **wherein the at least one node is associated with two or more network interfaces** ("adding two virtual interfaces for the [] system"). EX1003, ¶¶149-150.

Specifically, like the '282 patent, **Ke** discloses a configurable firewall protecting a series of virtual local area networks (VLANs) using a packet-based

firewall model. EX1005, [0005] ("The data processing system includes a firewall engine that can receive a set of firewall policies and apply the firewall policies to a data packet...one or more virtual private networks that each have an associated destination address and policies and a controller that can detect an incoming data packet, examine the incoming data packet for a virtual private network destination address and identify the policies associated with the virtual private network destination. If the policies include firewall policies, then the controller can call the firewall engine and apply the set of firewall policies corresponding to the virtual private network destination to the data packet...."); see also [0032]-[0035]. **Ke's** network security system includes "security system resources including firewall services and a controller that can partition the security system resources into a plurality of separate security domains. Each security domain can be configurable to enforce one or more policies relating to a specific subsystem, and to allocate security system resources to the one or more security domains." EX1005, [0016]. EX1003, ¶152.

Ke's system "provides a multi-customer, multi-domain architecture" which allows administrators "to create and manage separate security domains, each domain acting as a stand alone system and having its own set of policies." *Id.*, [0031]. **Ke** accomplishes this using "Virtual Systems" (or "Virtual Local Area Networks", "VLANs"). *Id.*, [0031]-[0033]. Specifically, "[o]n the secure side of the firewall

device (210) is a Virtual Local Area Network (VLAN) trunk (220) that carries all packets to a second 100/1000 switch (225). *A VLAN is a Layer 2 multiplexing technique that allows several streams of data to share the same physical medium, such as a trunk cable, while enjoying total segregation.* The second switch (225) directs the packets on private links to the different customers' servers (230) through a 10/100 switch (235) for each customer." *Id.* [0033]. EX1003, ¶153.

In this manner, **Ke's** teachings of using "Virtual Systems" to create and manage customer domains is similar to **Coss's** disclosure of protecting a multi-domain system, where each domain may be used by different corporations. *Compare* Ke, [0031] ("multi-customer, multi-domain architecture") *with* Coss, 3:35-56 (disclosing "multiple security domains" comprising "four user sites...of corporations A through D"). To the extent **Coss** discloses a multi-domain system (and one with defined host "groups" that are dynamically modifiable), but does not provide explicit disclosure on how to configure and define such domains, **Ke** provides those implementation details. EX1003, ¶151

Ke provides express disclosure of "configuration of the Internet security system in real time or at start up with a saved configuration script." EX1005, [0058]. For example, **Ke** discloses configuring an "Internet Security System" to support multiple domains—*i.e.*, "VLAN1 = Customer A", "VLAN2 = Customer B"; etc.—by creating "virtual system[s]" (Ke, [0059]-[0082]) with corresponding security

policies for incoming and outgoing packets. Ke, [0115]-[122]. For example, Ke discloses configuring "a new virtual system named 'marketing' and configur[ing] that system." *Id.*, [0069]. This includes "adding two virtual interfaces for the 'marketing' system." *Id.* [0071] These "virtual interfaces," acting as the incoming and outgoing interfaces for the VLAN, "allows several streams of data to share the same physical medium, such as a trunk cable, while enjoying total segregation." *Id.* [0033]. EX1003, ¶153.

A POSITA would have been motivated and would have found it obvious and advantageous to implement Coss's multi-domain firewall system using Ke's teachings of implementing domains as "virtual systems" with corresponding network interfaces. Both Coss and Ke are in the same field of art as the '282 patent—computer networking and, more specifically, protecting multi-domain systems with different firewall policies—and solve similar problems. *See, e.g.*, EX1001, 1:35-37 (disclosing "network firewalls that can dynamically adapt to changing conditions and operator requirements"); Ke, [0004] (discussing "complex[ity]" and cost and downtime problems of relying heavily on physical network infrastructure requiring a "large amount of separate equipment" and, that "every time a new customer joins the Internet data center...may require network re-configuration, and be a labor intensive and costly task"). Thus, a POSITA would have understood that implementing Coss's multi-domain system using virtual domains would have saved

time, cost, and complexity. Similarly, both Coss and Ke emphasize runtime efficiency—Coss by implementing dynamic rules at runtime (Coss, 8:28-40) and Ke with its real-time reconfiguration of the network (Ke, [0058])—and a POSITA would have recognized the benefits of adding additional runtime flexibility to Coss's system as outlined in Ke. Indeed, by disclosing "host group[s]" that can allow for hosts to be dynamically added or dropped to its system without otherwise affecting the operation of the rules, Coss provides an express motivation for combining with Ke's teachings. Coss, 2:41-46. And a POSITA would have recognized the benefits in scalability and cost afforded by implementing Ke's teachings in Coss. EX1003, ¶154.

A POSITA would have had a reasonable expectation of success in implementing Coss in view of Ke, as the combination of teachings involves straightforward concepts in the networking space. Coss's domain portioning is conceptually similar to Ke's virtual domains and POSITA would have understood that implementing Coss's domains as Virtual LANs is a straightforward integration of compatible technologies. For example, Coss already recognizes the need for "tunneling" to provide secure communications between domains and a POSITA would have recognized that these could advantageously and easily be implemented in a VLAN system, as disclosed by Ke. *See, e.g.*, Coss, 6:66-7:3 (discussing address changes and "tunnel option" for packet redirection), Ke, [0052]-[0053]. Thus a

POSITA would have understood **Coss** and **Ke** to provide complementary disclosures and combining their teachings would implement well-understood principles for their intended purposes. EX1003, ¶155.

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

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

Petitioner is the real party-in-interest.

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

The '282 patent is asserted against Petitioner in *Netskope, Inc. v. Fortinet, Inc.*, No. 3:25-cv-2360 (N.D. Cal.).

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

Designated Counsel for Petitioner and service information is below:

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

Petitioner consents to service by email at the addresses above.

VII. FEES UNDER 37 C.F.R. §42.103

Petition and Post-Institution fees totaling \$73,025.00 have been paid by electronic funds transfer.

VIII. CONCLUSION

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

Respectfully Submitted,

Dated: October 10, 2025

/s/Andrew Gish

Andrew Gish (Reg. #67,562)

ATTORNEY FOR PETITIONER

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

I, the undersigned, do hereby certify that the attached petition contains 10,513 words, as measured by the Word Count function of Microsoft Word. This is less than the limit of 14,000 words as specified by 37 C.F.R. §42.24(a)(i).

Dated: October 10, 2025

/s/ Andrew Gish

Andrew Gish (Reg. # 67,562)

ATTORNEY FOR PETITIONER

CERTIFICATION OF SERVICE

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

25883 - MUNCK WILSON MANDALA L.L.P
2000 McKinney Ave Ste 1900
Dallas, TX
UNITED STATES

Dated: October 10, 2024

/s/ Andrew Gish

Andrew Gish (Reg. # 67,562)

ATTORNEY FOR PETITIONER