

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC
Petitioner

v.

K.MIZRA LLC
Patent Owner

Case No. IPR2025-01436
Patent 8,234,705

**PETITION FOR INTER PARTES REVIEW
OF U.S. PATENT NO. 8,234,705**

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. MANDATORY NOTICES PURSUANT TO 37 C.F.R. §42.8(a)(1).....	4
A. Real Party-In-Interest	4
B. Identification of Related Matters Under 37 C.F.R. §42.8(b)(2)	4
C. Lead and Backup Counsel.....	7
D. Service Information Under 37 C.F.R. §42.8(b)(4).....	7
III. FEES	7
IV. REQUIREMENTS UNDER 37 C.F.R. §42.104.....	8
A. Grounds for Standing	8
B. Identification of Challenges and Precise Relief Requested	8
C. Prior Art Qualification of Asserted References	9
V. BACKGROUND	10
A. The '705 Patent	10
B. Examination History and Prior IPR	10
C. Prior Art.....	11
1. Freund (EX1005) and Freund '611 (EX1020).....	11
2. Ball (EX1006).....	13
3. Kappes (EX1007).....	14
4. Pujare (EX1009)	14
5. Lewis (EX1013).....	14
6. Kouznetsov (EX1021)	15

D.	Person of Ordinary Skill in the Art (“POSITA”).....	16
VI.	CLAIM CONSTRUCTION	16
A.	“protected network”	16
B.	“trusted computing base”	17
C.	“trusted platform module” (TPM).....	17
D.	“valid digitally signed attestation of cleanliness”	17
E.	“includes at least one of an . . . and an . . .”	18
F.	“quarantine” or “quarantining”	18
G.	“quarantine server”	18
H.	“a remediation host configured to provide data usable to remedy the insecure condition”	18
VII.	PETITIONER HAS A REASONABLE LIKELIHOOD OF PREVAILING	20
A.	Claims 1-19 Are Obvious Over Freund (EX1005) in view of Ball (EX1006), Pujare (EX1009), and Lewis (EX1013) (Ground 1)	20
1.	Claim 1 (Preamble): “A method for protecting a network, comprising:”	20
a.	[1.1]: “detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network,”	22
b.	[1.2]: “wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host,”	24
c.	[1.3]: “receiving a response, and”	29
d.	[1.4]: “determining whether the response includes a valid digitally signed attestation of cleanliness,”	29

e.	[1.5]: “wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host;”	31
f.	[1.6]: “when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network,”	35
g.	[1.7]: “wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host,”	35
h.	[1.8]: “serving a quarantine notification page to the first host when the service request comprises a web server request, and”	36
i.	[1.9]: “in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition; and”	38
j.	[1.10]: “permitting the first host to communicate with the remediation host.”	45
2.	Claim 2: “A method as recited in claim 1, wherein detecting an insecure condition further includes at least one of the following: scanning for a vulnerability, scanning for malicious data, determining whether a	

	security software is installed, and detecting anomalous network traffic.”	46
3.	Claim 3: “A method as recited in claim 1, wherein detecting an insecure condition includes determining that the first host should be quarantined until an update to an operating system has been installed.”	48
4.	Claim 4: “A method as recited in claim 1, wherein detecting an insecure condition includes configuring an operating system to quarantine the first host upon initial startup after installation of the operating system.”	49
5.	Claim 5: “A method as recited in claim 1, wherein permitting the first host to communicate with the remediation host includes: detecting an outbound communication from the first host; and forwarding the outbound communication if it is addressed to the remediation host.”	50
6.	Claim 6: “A method as recited in claim 1, wherein preventing the first host from sending data to the one or more other hosts includes: detecting an outbound communication from the first host; and redirecting the outbound communication to a quarantine server if it comprises a request for an approved service and is not addressed to a remediation host.”	54
7.	Claim 7: “A method as recited in claim 1, wherein quarantining the first host further includes preventing the first host from receiving via the protected network data not related to remediation of the insecure condition.”	56
8.	Claim 8: “A method as recited in claim 1, performed at an Internet service provider.”	56
9.	Claim 9: “A method as recited in claim 1, wherein the software component on the first host is an operating system.”	59
10.	Claim 10: “A method as recited in claim 1, wherein determining that the response does not include a valid	

	digitally signed attestation of cleanliness includes determining that the software component on the first host is not sufficiently updated.”	60
11.	Claim 11: “A method as recited in claim 1, wherein determining that the software component on the first host is not sufficiently updated includes determining that a patch level associated with the software component on the first host is not sufficiently recent.”	62
12.	Claim 12 (Preamble): “A system for protecting a network, comprising:”	63
	a. [12.1]: “a processor configured to:”	64
	b. [12.2]: “detect an insecure condition on a first host that has connected or is attempting to connect to a protected network,”	65
	c. [12.3]: “wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host,”	65
	d. [12.4]: “receiving a response, and”	65
	e. [12.5]: “determining whether the response includes a valid digitally signed attestation of cleanliness,”	65
	f. [12.6]: “wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host; ”	65
	g. [12.7]: “when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantine the first host, including by preventing the first host from sending data to	

	one or more other hosts associated with the protected network,”.....	66
h.	[12.8]: “wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host,”	66
i.	[12.9]: “serving a quarantine notification page to the first host when the service request comprises a web server request, and”	66
j.	[12.10]: “in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition; and”	66
k.	[12.11]: “permit the first host to communicate with the remediation host; and”	67
l.	[12.12]: “a memory coupled to the processor and configured to provide instructions to the processor.”	67
13.	Claim 13: “A system as recited in claim 12, wherein the processor is configured to detect an insecure condition at least in part by performing one or more of the following: scanning for a vulnerability, scanning for malicious data, determining whether a security software is installed, and detecting anomalous network traffic.”	67
14.	Claim 14: “A system as recited in claim 12, wherein the processor is configured to detect an insecure condition at least in part by determining that an initial startup after installation of an operating system is being performed.”	67
15.	Claim 15: “A system as recited in claim 12, wherein the processor is configured to quarantine the first host at least in part by preventing the first host from receiving via the	

	protected network data not related to remediation of the insecure condition.”	68
16.	Claim 16: “A system as recited in claim 12, wherein the software component on the first host is an operating system.”	68
17.	Claim 17: “A system as recited in claim 12, wherein determining that the response does not include a valid digitally signed attestation of cleanliness includes determining that the software component on the first host is not sufficiently updated.”	68
18.	Claim 18: “A system as recited in claim 17, wherein determining that the software component on the first host is not sufficiently updated includes determining that a patch level associated with the software component on the first host is not sufficiently recent.”	68
19.	Claim 19 (Preamble): “A computer program product for protecting a network, the computer program product being embodied in a non-transitory computer readable medium and comprising computer instructions for:”	69
	a. [19.1]: “detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network,”	69
	b. [19.2]: “wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host,”	69
	c. [19.3] receiving a response, and	70
	d. [19.4] determining whether the response includes a valid digitally signed attestation of cleanliness,	70
	e. [19.5] wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation	

	that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host;.....	70
f.	[19.6] when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network,	70
g.	[19.7] wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host,.....	71
h.	[19.8] serving a quarantine notification page to the first host when the service request comprises a web server request, and	71
i.	[19.9] in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition; and	71
j.	[19.10] permitting the first host to communicate with the remediation host.	71
VIII.	OTHER CONSIDERATIONS	71
IX.	CONCLUSION.....	72
APPENDIX A - LIST OF EXHIBITS		
CERTIFICATE OF COMPLIANCE WITH 37 C.F.R. §42.24		
CERTIFICATE OF SERVICE		

TABLE OF AUTHORITIES

Cases	Page(s)
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (<i>en banc</i>)	16
 Statutes	
35 U.S.C. §102	9
35 U.S.C. §102(a)	10
35 U.S.C. §102(b)	9
35 U.S.C. §102(e)	9
35 U.S.C. §103	9
35 U.S.C. §112(f)	44
35 U.S.C. §§311-319	1
 Rules	
37 C.F.R. §42.8(a)(1)	4
37 C.F.R. §42.8(b)(1)	4
37 C.F.R. §42.8(b)(2)	4
37 C.F.R. §42.8(b)(3)	7
37 C.F.R. §42.8(b)(4)	7
37 C.F.R. §42.10(a)	7
37 C.F.R. §42.10(b)	7
37 C.F.R. §42.15(a)	7

Petition for *Inter Partes* Review of U.S. Patent No. 8,234,705

37 C.F.R. §42.100(b)16
37 C.F.R. §42.100 *et seq.*.....1
37 C.F.R. §42.1048
37 C.F.R. §42.104(a).....8
37 C.F.R. §42.104(b)(3).....19

I. INTRODUCTION

Google LLC requests *inter partes* review of Claims 1-19 (“Challenged Claims”) of U.S. Patent No. 8,234,705 (“’705 Patent”) (EX1001) under 35 U.S.C. §§311-319; 37 C.F.R. §42.100 *et seq.*

The Challenged Claims relate to computer network-based contagion isolation and inoculation, to purportedly address a problem of protecting a private network from roaming “hosts” that connect to outside networks (e.g., Internet malware) with vulnerability to infestation. The Challenged Claims are directed to protecting the private network by detecting an insecure condition on the vulnerable “host” and quarantining that host from connecting to the network via, for example, domain name system (DNS) and web server service requests. A trusted computing base verifies the host’s cleanliness. When an insecure condition is detected, trusted network security components redirect an insecure host’s network service request to a “quarantine” server which then serves a quarantine notification page to the insecure host. *See* EX1001, 3:3-32; 14:1-26; EX1002, p.21. The quarantine server’s quarantine notification page includes a notice and/or instructions to the quarantined device (e.g., to download an appropriate anti-virus software update). EX1001, 3:13-23; 16:11-43.

The Examiner allowed the Challenged Claims indicating that the cited references failed to anticipate or render obvious the feature of “serving both the

specific quarantine notification page with the DNS redirection”. EX1002, p.21. The Examiner committed material error because the prior art cited herein discloses these exact features of providing in response to a DNS query, an IP address of a quarantine server configured to serve the quarantine notification page.

The prior art cited herein discloses detecting non-compliant, vulnerable virus hosts having insecure conditions such as infiltration by viruses and/or failure to comply with network access security policies (e.g., failure to update the anti-virus software version level). DNS and web server service requests of a non-compliant device are detected via trusted network security components which redirect the host’s service requests to the IP address of a “quarantine” server, referenced in the prior art as a “sandbox server” for serving a quarantine notification page to the non-compliant host device. *See* EX1005, ¶¶[0005], [0078], [0085]-[0089], [0147]-[0150]. A non-compliant, quarantined device is restricted to communication with the sandbox server. The sandbox server redirects the non-compliant host device to a remediation host to remedy the insecure condition (e.g., download updated anti-virus software, such as a software patch). *See* EX1007, ¶¶[0008], [0029], [0037]-[0038]; *see* EX1005, ¶¶[0042], [0071]-[0078], [0148]-[0150]; Figs. 7-8. The Challenged Claims are therefore unpatentable due to clear material error that occurred during initial examination. *See* EX1005, EX1006, EX1009, EX1013 each in its entirety.

This Petition presents an exceptional case of *unsettled* expectations following a recent IPR challenge by a different party (Cisco Systems) that terminated this year by settlement with K.Mizra, who acquired the Patent in 2019. The settlement followed a Federal Circuit (“CAFC”) remand and occurred just weeks before K.Mizra sued Google on relatively recent products Google introduced to the marketplace. Clearly K.Mizra has no settled expectations given the cloud of invalidity due to the CAFC remand.

While the Final Written Decision issued in the Cisco IPR initially upheld the claims, the CAFC determined that the PTAB’s obviousness rationale “was rooted in legal error and a fact finding unsupported by substantial evidence” and “vacate[d] the ultimate determination that Cisco failed to show” unpatentability. EX1017. K.Mizra opted to quickly settle the IPR rather than resolve the material errors raised with regard to validity. Discretionary denial is unwarranted given: (1) the CAFC remand; (2) the USPTO’s material error during prosecution; and (3) the early stage of the parallel litigation. K.Mizra’s piecemeal litigation tactics involving serial infringement allegations since acquiring the patent in 2019, only to sidestep material errors associated with the ’705 Patent’s grant, warrant institution of this IPR as the fairest and judicially efficient route to resolving invalidity of the ’705 Patent.

II. MANDATORY NOTICES PURSUANT TO 37 C.F.R. §42.8(a)(1)

A. Real Party-In-Interest

Pursuant to 37 C.F.R. §42.8(b)(1), Petitioner certifies that the real party in interest is Google LLC¹.

B. Identification of Related Matters Under 37 C.F.R. §42.8(b)(2)

The following is a list of any judicial or administrative matters that would affect, or be affected by, a decision in this proceeding:

Related District Court Matters

Case	Filing Date	Status
<i>Netskope, Inc. v. K.Mizra LLC</i> , Civ. Action No. 3:25-cv-04833 (N.D. Cal.)	June 6, 2025	Pending
<i>K.Mizra LLC v Citrix Systems, Inc. et al.</i> , Civ. Action No. 0:25-cv-60803 (S.D. Fla.)	April 24, 2025	Pending
<i>K.Mizra LLC v. Google LLC</i> , Civ. Action No. 1:25-cv-00236 (W.D. Tex.) (“the ’236 Litigation”)	February 18, 2025	Pending
<i>K.Mizra LLC v. SonicWall Inc.</i> , Civ. Action No. 1:25-cv-00047 (D. Del.)	January 10, 2025	Settled
<i>Hewlett Packard Enterprise Co. v. K.Mizra LLC</i> , Appeal 23-1183 (CAFC)	November 23, 2022	Vacated

¹ Google LLC is a subsidiary of XXVI Holdings Inc. which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

Petition for *Inter Partes* Review of U.S. Patent No. 8,234,705

Case	Filing Date	Status
<i>Cisco Systems, Inc. v. K.Mizra LLC</i> , Appeal 22-2290 (CAFC)	September 29, 2022	Vacated
<i>K.Mizra LLC v. Hewlett Packard Enterprise Co. et al</i> , Civ. Action No. 2:21-cv-00305 (E.D. Tex.)	August 9, 2021	Settled
<i>K.Mizra LLC v. Forescout Technologies, Inc.</i> , Civ. Action No. 2:21-cv-00248 (E.D. Tex.)	July 8, 2021	Settled
<i>K.Mizra LLC v. Forescout, Inc.</i> , Civ. Action No. 2:21-cv-00249 (E.D. Tex.)	July 8, 2021	Settled
<i>K.Mizra LLC v. Cisco Systems, Inc.</i> , Civ. Action No. 6:20-cv-01031 (W.D. Tex.)	November 6, 2020	Settled
<i>Network Security Technologies, LLC v. Bradford Networks, Inc.</i> , Civ. Action No. 1:17-cv-01487 (D. Del.)	October 24, 2017	Settled
<i>Network Security Technologies, LLC v. ForeScout Technologies, Inc.</i> , Civ. Action No. 1:17-cv-01488 (D. Del.)	October 24, 2017	Settled
<i>Network Security Technologies, LLC v. McAfee, Inc.</i> , Civ. Action No. 1:17-cv- 01489 (D. Del.)	October 24, 2017	Settled
<i>Network Security Technologies, LLC v. Pulse Secure, LLC</i> , Civ. Action No. 1:17-cv-01490 (D. Del.)	October 24, 2017	Settled

Related PTAB Matters

Case	Filing Date	Status
<i>Citrix Systems, Inc. et al. v. K.Mizra LLC</i> , IPR2025-01468 (PTAB)	August 29, 2025	Pending

Petition for *Inter Partes* Review of U.S. Patent No. 8,234,705

Case	Filing Date	Status
<i>Netskope, Inc. v. K.Mizra LLC</i> , IPR2025-01115 (PTAB)	August 13, 2025	Pending
<i>Forescout Technologies, Inc. v. K.Mizra LLC</i> , IPR2022-00081 (PTAB)	October 22, 2021	Joined with IPR2021-00593
<i>Hewlett Packard Enterprise Co. v. K.Mizra LLC</i> , IPR2022-00084 (PTAB)	October 22, 2021	Joined with IPR2021-00593
<i>Cisco Systems, Inc. v. K.Mizra LLC</i> , IPR2021-00593 (PTAB)	March 15, 2021	Final Written Decision (all challenged claims not unpatentable) September 19, 2022; Appealed (Appeal Nos. 22-2290; 23-1183); Settled December 2024

Other Related PTAB Matters

Case	Filing Date	Status
<i>Google LLC v. K.Mizra LLC</i> , IPR2025-01437 (PTAB)	September 19, 2025	Pending

Related Applications

The '705 Patent issued from U.S. Patent Application No. 11/237,003, filed September 27, 2005, claiming priority to U.S. Provisional Application No. 60/613,909 (EX1004), filed September 27, 2004.

C. Lead and Backup Counsel

Pursuant to 37 C.F.R. §42.8(b)(3) and §42.10(a), Petitioner hereby identifies its lead and backup counsel as follows:

<p><u>Lead Counsel</u> Patrick C. Keane (Reg. No. 32,858) BUCHANAN INGERSOLL & ROONEY PC 1737 King Street, Suite 500 Alexandria, Virginia 22314 Direct Telephone (703) 838-6522 Main Facsimile (703) 836-2021 patrick.keane@bipc.com</p>	<p><u>Backup Counsel</u> Roger H. Lee (Reg. No. 46,317) BUCHANAN INGERSOLL & ROONEY PC 1737 King Street, Suite 500 Alexandria, Virginia 22314 Telephone (703) 838-6545 Facsimile (703) 836-2021 roger.lee@bipc.com</p>
<p><u>Backup Counsel</u> Andrew J. Koopman (Reg. No. 65,537) BUCHANAN INGERSOLL & ROONEY PC 2200 Renaissance Blvd., Suite 350 King of Prussia, PA 19406 Telephone (610) 993-4217 Facsimile (610) 407-0701 andrew.koopman@bipc.com</p>	<p><u>Backup Counsel</u> Samuel Harrod (Reg. No. 79,148) BUCHANAN INGERSOLL & ROONEY PC Union Trust Building 501 Grant Street Suite 200 Pittsburgh, PA 15219 Telephone (412) 562-8805 Facsimile (412) 562-1041 samuel.harrod@bipc.com</p>

A Power of Attorney is being filed concurrently herewith in accordance with 37 C.F.R. §42.10(b).

D. Service Information Under 37 C.F.R. §42.8(b)(4)

Petitioner consents to electronic service at the email addresses listed above.

III. FEES

The undersigned authorizes the Office to charge Deposit Account 02-4800 for fees required by 37 C.F.R. §42.15(a).

IV. REQUIREMENTS UNDER 37 C.F.R. §42.104

A. Grounds for Standing

Petitioner certifies this IPR, and Petitioner is not barred or estopped from requesting this IPR. 37 C.F.R. §42.104(a).

B. Identification of Challenges and Precise Relief Requested

Petitioner challenges Claims 1-19 of the '705 Patent as unpatentable as follows:

Ground	Reference(s)	Basis	Claims
1	U.S. Patent Application Publication No. 2003/0055962 to G. Freund <i>et al.</i> (“Freund”) (EX1005) in view of U.S. Patent Application Publication No. 2006/0005009 to C. Ball <i>et al.</i> (“Ball”) (EX1006), U.S. Patent Application Publication No. 2002/0083183 to Pujare <i>et al.</i> (“Pujare”) (EX1009), and U.S. Patent No. 7,533,407 to Lewis <i>et al.</i> (“Lewis”) (EX1013)	103	1-19

The Exhibit List (Appendix A) includes the Declaration and *Curriculum Vitae* of Dr. Markus Jakobsson (EX1010, ¶¶1-352; EX1011).

C. Prior Art Qualification of Asserted References

The prior art applied in Ground 1 constitutes prior art as of the earliest claimed filing date of U.S. Provisional Application No. 60/613,909.² The '705 Patent is subject to pre-AIA 35 U.S.C. §§102/103.

Freund (EX1005), published on March 20, 2003 and filed on August 30, 2001, is prior art at least under pre-AIA §102(b) and §102(e).

Ball (EX1006), filed on June 30, 2004, is prior art at least under pre-AIA §102(e).

Kappes (EX1007), filed on November 25, 2003, is prior art at least under pre-AIA §102(e).

Pujare (EX1009), published on June 27, 2002 and filed on April 5, 2001, is prior art at least under pre-AIA §102(b) and §102(e).

Lewis (EX1013), issued on May 12, 2009 and filed on April 14, 2004, is prior art at least under pre-AIA §102(e).

Freund '611 (EX1020), issued on November 16, 1999 and filed on May 6, 1997, is prior art at least under pre-AIA §102(b) and (e).

² Petitioner does not concede that any challenged claim is entitled to an effective filing date of September 27, 2004.

Kouznetsov (EX1021), issued on August 24, 2004 and filed on August 30, 2000, is prior art at least under pre-AIA §102(a) and (e).

V. BACKGROUND

A. The '705 Patent

The '705 Patent is directed to computer network-based contagion isolation and inoculation, to protect a private network from roaming “hosts” that connect to outside networks (e.g., the Internet) with attendant vulnerability to hosting a virus infestation. The Challenged Claims recite protecting a network by detecting an insecure condition on the host and quarantining the host from connecting to a network via, for example, domain name system (DNS) and web server service requests. A trusted computing base verifies the host’s cleanliness. When an insecure condition is detected, trusted network security components redirect the insecure host’s service request to a “quarantine” server. The quarantine server serves a quarantine notification page to the insecure host. *See* EX1001, 3:8-23; 14:4-12; EX1002, p.21. The quarantine notification page provides notice and/or instructions to the quarantined host device to remedy the insecure condition (e.g., download an anti-virus software update from a remediation host). EX1001, 3:13-23; 16:11-43. EX1010, ¶27.

B. Examination History and Prior IPR

Applicant amended the claims to recite specifics of the digitally signed attestation of cleanliness, including “the presence of a patch or patch level associated

with a software component”. EX1002, pp.38-45. An Examiner’s Amendment included “in the event the service request comprises a DNS query, providing in response an IP address to a quarantine server”. *Id.*, pp.25-28. The Examiner stated that Liang and Yan fail to anticipate or render obvious “serving both the specific quarantine notification page with the DNS redirection when in combination with the remaining claim limitations.” *Id.*, p.21. This Petition cites prior art which clearly discloses the allowable subject matter.

The ’705 Patent was previously challenged in IPR2021-00593 filed by Cisco that settled earlier this year. EX1016, EX1017, EX1018. While the Final Written Decision initially upheld the claims, the CAFC “vacate[d] the Board’s motivation to combine analysis, which was rooted in legal error and a fact finding unsupported by substantial evidence”, “further vacate[d] the ultimate determination that Cisco failed to show” unpatentability, and remanded to the PTAB where PO settled. EX1017. Validity was therefore left unresolved.

C. Prior Art

1. Freund (EX1005) and Freund ’611 (EX1020)

Freund discloses protecting a computer network by quarantining a non-compliant, vulnerable roaming client (i.e., host) device using a “sandbox server” to perform quarantine functions. Web server requests (e.g., HTTP) and/or DNS requests from a potentially vulnerable host device are redirected to the sandbox

server, which provides a quarantine notification page to the vulnerable device and initiates remediation. EX1005, ¶¶[0148]-[150], Figs. 7-9. Freund's system seeks to protect private networks (e.g., LANs) from viruses or other infections that may be obtained via other networks (e.g., Internet malware). *Id.*, ¶¶[0014]-[0021]. Freund's system includes a client-side security component/module, and an associated router-side security module to challenge the host device which is connected to or attempting to connect to the local LAN for ensuring compliance with network security policies. *Id.*, ¶¶[0039]-[0041]. When Freund's system detects a non-compliant device (e.g., a potentially infested host device), Freund's system redirects communication from that device to the sandbox server to quarantine the device and serve a notification page to initiate remediation measures from a remediation host. *Id.*, ¶[0042]. EX1010, ¶¶53-55.

Freund '611, incorporated by reference in Freund (EX1005, ¶[0017]), discloses a "prior ZoneAlarm™ product" related to Freund's disclosure. *Id.* Freund '611 similarly discloses "regulating access and maintaining security of individual computer systems and local area networks (LANs) connected to larger open networks (Wide Area Networks or WANs), including the Internet." EX1020, 1:27-30. Freund '611 discloses "client-based monitoring and filtering of access" (*id.*, 3:61) using filtering implemented, for example, at an Internet Service Provider (ISP). *See id.*, 21:47-22:41; 29:17-30:10.

2. Ball (EX1006)

Ball discloses host hardware can be configured with a “Trusted Platform Module” (TPM) conforming to an industry standard “Trusted Computing Group” (TCG) Specification that was well known and conventionally integrated in a vulnerable client-side host device as a client-side security component for providing security via cryptographic operations and securely stored encryption keys that securely convey attestations of the host device’s cleanliness. A “TPM comprises a passive device that is installed in a computing device, and which can accurately measure, securely store, and securely communicate information on one or more attributes of the computing device.” EX1006, ¶[0006]. EX1010, ¶¶57-58.

Ball’s solution verifies the attributes of a client-side host. EX1006, Abs. Attributes are measured using a TPM defined by the TCG Specification (EX1008) and integrated on the computing device. *Id.*, ¶[0006]. The TPM can securely and accurately measure and communicate attributes about the computing device and digitally sign (“encrypt”) the attributes using an attestation identity key (AIK). *Id.*, ¶[0033]. EX1010, ¶¶59-60.

AIKs encrypt device information and authenticate device information when it is transmitted to another device (e.g., in response to a security challenge). EX1006, ¶[0033]. AIKs ensure the device’s identity can be trusted and the information

attested to by the TPM is accurate. *Id.* Ball’s TPM hardware ensures attestation accuracy. *Id.*, ¶¶[0030], [0033]. EX1010, ¶61.

3. Kappes (EX1007)

Kappes protects a network by authenticating client devices requesting access to the network. EX1007, Abs., ¶[0059]. Potentially vulnerable host devices are quarantined and permitted limited access to restoration services. *Id.*, ¶¶[0036]-[0038], [0052], Fig. 3. The restoration services use “standard packet filtering techniques” to filter web server requests and DNS requests. *Id.*, ¶¶[0029], [0038]. EX1010, ¶¶62-64.

4. Pujare (EX1009)

Pujare discloses a “versioning table” that makes clear that application patches are merely software upgrades. EX1009, ¶[0019]. Pujare’s versioning table contains a list of root file numbers and version numbers. *Id.* The root file numbers and version numbers are used to track the application patches and upgrades. *Id.* Pujare’s disclosure demonstrates that it was well known and conventional for software version numbers to correspond to patch levels. *Id.* A client receives the versioning table and compares it with the client’s application root file number/version number to find files required for a software patch or update. *Id.* EX1010, ¶¶65-67.

5. Lewis (EX1013)

Lewis discloses an alternate method of DNS redirection of insecure client devices to the IP address of a quarantine server. *See* EX1013, 12:11-18, 14:4-24, Fig.

8B. A DNS destination address of a non-compliant host device is rerouted to a “hi-jack” DNS server that is configured to provide only the IP address of a quarantine server. EX1013, 14:20-24, 11:15-17, 14:4-24. This causes non-compliant devices to be routed to a fix-up page of the quarantine server. *Id.* EX1010, ¶¶68-69.

The earlier publication of Lewis (U.S. 2005/0131997) was cited during examination. EX1002, p.187. The Examiner relied on Lewis to reject dependent claims. *Id.*, pp.68-69, 112-113, 187-189. Applicant never disputed the Examiner’s assertions regarding Lewis and its disclosure concerning Element [1.9] (*see* VII.A.1.i. below). EX1010, ¶70.

Petitioner merely relies on Lewis as an alternative technique to the technique disclosed by Freund for providing an IP address of a quarantine server in response to a DNS query by a non-compliant host device. Freund (EX1005) discloses and/or suggests providing an IP address of a quarantine server in response to a DNS query. As detailed in this Petition, Freund discloses the ’705’s purported novel features. Further, Lewis discloses an exemplary embodiment called out in the ’705 Patent. Petitioner’s reliance on Lewis as backup support for Freund’s disclosure of a particular claimed feature does not warrant discretionary denial. EX1010, ¶71.

6. Kouznetsov (EX1021)

Kouznetsov discusses software application management and discloses that maintenance of software applications (e.g., anti-virus software) includes

implementation of incremental improvements, such as patches, updates, and versions. EX1021, 1:39-3:33. EX1010, ¶72.

D. Person of Ordinary Skill in the Art (“POSITA”)

A person of ordinary skill in the art (“POSITA”) pertinent to the ’705 Patent would have had a bachelor’s degree in electrical engineering, computer engineering, or a related discipline, knowledge of networking as of this time, and at least two years of experience working in a field involving networking and system security. A person with a different degree could still qualify if they have additional experience that compensates for the different educational backgrounds. A person with less experience working in a field involving networking and system security could still qualify if the person has additional education that compensates for their lesser experience. *Id.* EX1010, ¶19.

VI. CLAIM CONSTRUCTION

Petitioner interprets the claims “in accordance with the ordinary and customary meaning ... as understood by one of ordinary skill in the art.” 37 C.F.R. §42.100(b). *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (*en banc*).

In Related Litigation, Petitioner and PO proposed certain constructions, discussed below. EX1014, pp.2-17; EX1015, pp.1-19.

A. “protected network”

In Related Litigation, Petitioner proposes “protected network” should be construed as:

private network, distinct from public networks like the Internet

EX1014, p.2. EX1010, ¶40.

B. “trusted computing base”

In Related Litigation, Petitioner proposes “trusted computing base” should be construed as:

hardware or software within the first host that provides security to the host

EX1014, p.4. PO proposes construing this term as “hardware or software that has been designed to be a part of the mechanism that provides security to a computer system.” EX1015, p.4. EX1024, p.6. EX1010, ¶41.

C. “trusted platform module” (TPM)

In Related Litigation, Petitioner and PO agree that “trusted platform module” should be construed as:

a secure cryptoprocessor that can store cryptographic keys and that implements the Trusted Platform Module specification from the Trusted Computing Group

See EX1014, p.2; EX1015, p.1; EX1019, p.1 (WDTX Order). *See also* EX1024, p.6 (EDTX Order). EX1010, ¶42.

D. “valid digitally signed attestation of cleanliness”

In Related Litigation, Petitioner proposes this term should be construed to have its plain and ordinary meaning wherein the plain and ordinary meaning is that

the “attestation of cleanliness” is digitally signed by and received from the “trusted computing base”. EX1014, p.10. *See also* EX1024, p.7. EX1010, ¶43.

E. “includes at least one of an . . . and an . . .”

In Related Litigation, Petitioner proposes this phrase should be construed as:

includes at least one of an . . . and at least one of an . . .

EX1014, p.11. EX1010, ¶44.

F. “quarantine” or “quarantining”

In Related Litigation, Petitioner proposes “quarantine” or “quarantining” should be construed as:

isolating from the protected network

EX1014, p.13. EX1010, ¶45.

G. “quarantine server”

In Related Litigation, Petitioner proposes “quarantine server” should be construed as:

server to which a quarantined host’s network traffic is redirected

EX1014, p.16. EX1010, ¶46.

H. “a remediation host configured to provide data usable to remedy the insecure condition”

In Related Litigation, Petitioner proposes this phrase should be construed as a means-plus-function term and is indefinite for lack of corresponding structure to perform the claimed function of “provide data usable to remedy the insecure

condition”. EX1014, p.17. Merely for compliance with 37 C.F.R. §42.104(b)(3) in this IPR, Petitioner identifies the following purported corresponding structure disclosed at column 11, lines 40-56 and column 15, lines 36-50 of the '705 Patent (i.e., “servers providing security patches or advisories, for example allowing connections to a vendor’s server, or providing virus and worm disinfecting tools, such as focused removal tools, or data updates for previously installed repair tools”; “vendor sites known to provide remediation assistance, such as to Microsoft's Windows Update service, where security patches may be obtained.”; “a security update site such as Microsoft Windows Update.”; and “a remediation site, such as a security update service such as Microsoft Windows Update”). For compliance with §42.104(b)(3), a ‘remediation host’ encompasses, e.g., a site from which remediation assistance is received. EX1010, ¶47.

In Related Litigation, PO asserted plain and ordinary meaning for the terms discussed in Sections A, D, E, F, G, and H above. EX1015, pp.2, 6, 7, 11, 13, 14.

To the extent any claim terms are construed as means-plus-function in this IPR or Related Litigation, the prior art cited herein discloses the terms with at least the same level of detail including corresponding structure and claimed functions as described in the '705 Patent. EX1010, ¶48.

The prior art cited herein discloses the noted terms under Petitioner's and PO's proposed constructions and any other constructions, as demonstrated herein. EX1010, ¶¶49-52.

VII. PETITIONER HAS A REASONABLE LIKELIHOOD OF PREVAILING

The Challenged Claims are unpatentable over the prior art. EX1010, ¶73.

A. Claims 1-19 Are Obvious Over Freund (EX1005) in view of Ball (EX1006), Pujare (EX1009), and Lewis (EX1013) (Ground 1)

The following sections detail how Claims 1-19 are obvious over the prior art. EX1010, ¶74.

1. Claim 1 (Preamble): “A method for protecting a network, comprising:”

To the extent the Preamble is limiting, Freund discloses and suggests the Preamble. Freund discloses a method for protecting a network (e.g., a local area network (LAN)) from malicious code by denying connections to the network. EX1005, ¶¶[0004], [0068]. Freund aims to “protect the overall security of the network.” *Id.*, ¶[0019]. *See also id.*, ¶[0066]. Figure 3 of Freund is compared to Figure 10B of the '705 Patent:

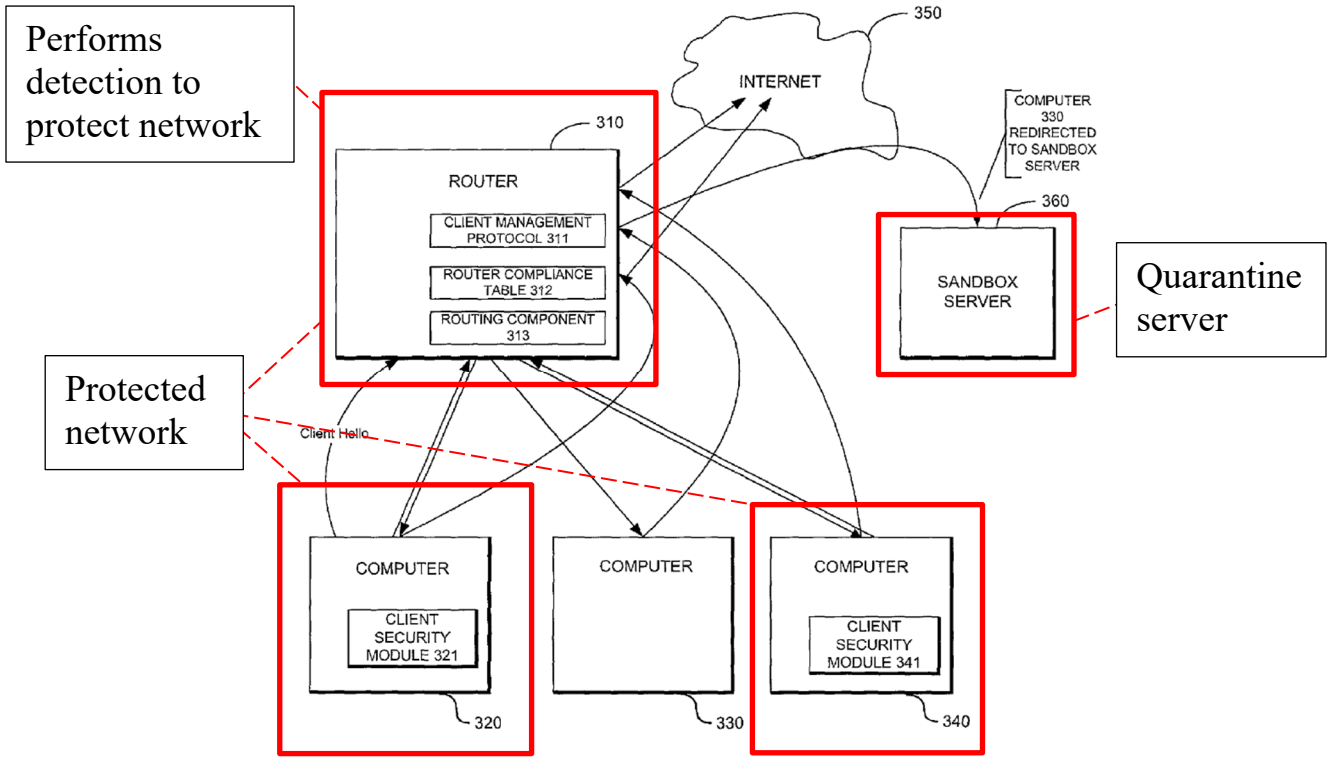


Figure 3 of Freund (Annotated)

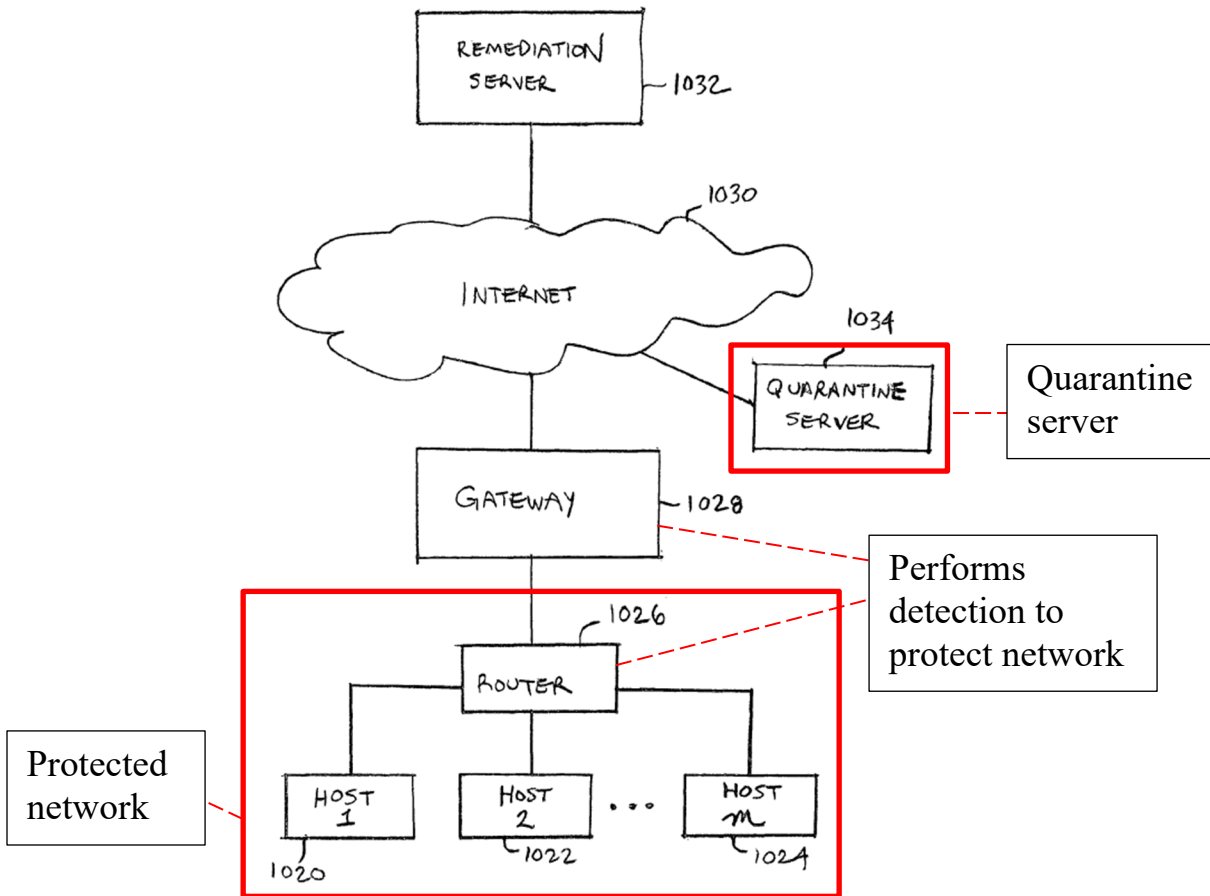


Figure 10B of '705 Patent (Annotated)

EX1010, ¶¶75-77.

- a. [1.1]: “detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network,”

Freund discloses and suggests Element [1.1]. Freund discloses detecting an insecure condition on a vulnerable first host (i.e., client-side host) that has connected or is attempting to connect to a protected network. A comparison of the above annotated Freund Figure with the annotated '705 Patent Figure 10B demonstrates that the hosts of both connect to a router device to access the network. All

communication on the network passes through the router device, including communication to other hosts. EX1010, ¶¶78-80.

Freund's Figure 7 shows a notification page served by a quarantine server after an insecure condition was detected. EX1005, Fig. 7. Freund discloses detecting a non-compliant device (e.g., running an older, outdated version of security software or having a virus) that has connected or is attempting to connect to a protected network (e.g., a LAN). EX1005, ¶¶[0071], [0078], [0088]. Freund's client monitoring protocol (CMP) evaluates responses from a host device to determine if the device is compliant or non-compliant. *Id.*, ¶¶[0084]-[0085], [0088]-[0089]. If Freund's CMP determines the device properly responded, then it is determined compliant. If a device did not properly respond, or did not respond at all, Freund's CMP determines the device has an insecure condition (e.g., insufficient anti-virus version, security module disabled, presence of virus, etc.). *Id.*, ¶[0088]. EX1010, ¶¶81-83.

Freund's disclosure refers specifically to a protected network as a "private network" which can connect to the Internet. EX1005, ¶[0028]. The protected network can be a Local Area Network (LAN) on a client's premises which can connect to "larger open networks (Wide Area Networks or WANs), including the Internet. *Id.*, ¶[0004]; [0069]-[0070]. EX1010, ¶84.

b. [1.2]: “wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host,”

Freund in view of Ball discloses and suggests Element [1.2]. Freund detects the insecure condition by contacting a trusted computing base (TCB) configured as Freund’s “client monitoring protocol software” (CMP) and/or “client-side security component/module” associated with a TPM within the first host. Freund’s CMP contacts a client-side security module (i.e., the TCB) within the host device. EX1005, ¶¶[0084]-[0089]. Freund’s CMP contacts the client-side security module of the host device by sending a router challenge. *Id.*, ¶[0084]. The client-side security module receives the router challenge and responds to the router challenge. *Id.*, ¶[0093]. *See also id.*, ¶¶[0077]-[0078], [0100]. The CMP can challenge the client-side security module to determine a version level of security software and/or to verify appropriate anti-virus software is running on the device. *Id.*, ¶¶[0127]-[0132]. Freund’s router-side CMP and client-side security module perform various security functions by handling router challenges and responding to security requirements issued by the router challenge (e.g., providing application status, anti-virus software version installation and status, and so forth). EX1005, ¶¶[0091]-[0093], [0118]-[0144]. Freund’s router-side CMP and/or client-side security module are part of the mechanism providing security to Freund’s hosts and network. Freund’s router-side CMP and/or client-side security module satisfy the claimed TCB under Petitioner’s

and PO's constructions. *See* EX1014, p.4; EX1015, p.4; EX1024, p.6. EX1010, ¶¶85-87.

Freund discloses attestations involving digital signatures to avoid being forged, and stores a private key to generate the digital signature. *See* EX1005, ¶[0127]. Freund discloses CPUs or processors including “any other suitable microprocessor or microcomputer” where program logic (e.g., for the client-side security module) is executed. EX1005, ¶¶[0057]-[0058]. Freund uses such CPUs or processors “within the first host”, i.e., they reside within the client computer. *Id.*, ¶¶[0043], [0065]. *See also id.*, ¶¶[0053]-[0061]. Freund's disclosure of the client-side security module constitutes a TCB given that the client-side security module is hardware and/or software within the first host that provides security to the host. EX1010, ¶¶88-90.

Freund's client-side security module for interfacing with a router-side security module to access a network is not expressly disclosed to be a “Trusted Platform Module (TPM)” conforming to the TCG as called out in the '705 Patent. While Freund discloses a CPU/processor as hardware for executing the TCB, Freund does not expressly disclose that this CPU/processor hardware is a TCB associated with specialized “TPM” hardware. EX1005, ¶[0056]. A POSITA would have recognized that a hardware-based TPM conforming to the TCG specification would have been an obvious, well-known hardware choice for implementing Freund's TCB security

functions. EX1010, ¶¶91-92. A POSITA would have recognized that such a substitution of known TPM hardware to implement enhanced functionality of Freund's TCB would have achieved predictable results with a reasonable expectation of success for several reasons discussed herein. *See, e.g.*, EX1006. *See also* EX1007. EX1010, ¶93.

Ball expressly discloses a TPM according to the TCG Specification to be a well-known, conventional CPU/processor for executing the client-side security module and security-based functions disclosed by Freund. Ball states that a "TPM comprises a passive device that is installed in a computing device, and which can accurately measure, securely store, and securely communicate information on one or more attributes of the computing device." EX1006, ¶[0006]. Ball's TPM and associated hardware satisfy the claimed TCB under Petitioner's and PO's constructions. EX1010, ¶¶94-96.

It therefore would have been obvious, predictable and beneficial to configure Freund's TCB as a trusted platform module (TPM) based on Ball's teachings that a TPM conforming to a TCG specification was a well-known, desirable hardware option for securely implementing client-side security functions according to industry standards. This would have allowed Freund's client-side host to verify its trustworthiness by securely communicating accurate information regarding anti-virus versions, anti-virus status, security compliance, and digitally signed

applications using cryptographic data keys to generate digital signatures and encryption/decryption. *Id.* EX1005, ¶¶[0071], [0110], [0127]. EX1010, ¶97. It would have been obvious to a POSITA to use Ball's TPM hardware to house and implement the client-side security module of Freund for securely storing cryptographic keys, hash functions, and identity information. EX1006, ¶[0006]; EX1005, ¶¶[0091]-[0093], [0110], [0127], [0144]. EX1010, ¶98.

A POSITA would have had reason to substitute at least one of the processors disclosed in Freund to execute security functions of the client-side security module using Ball's disclosed TPM (e.g., either integrated or dedicated). Ball's TPM would have provided enhanced hardware-based security for Freund's client-side security module and associated router CMP challenge functions (*see* EX1006, ¶¶[0006], [0033]-[0034], [0046]), and provided enhanced trust in the security configuration of Freund's client-side device. EX1007, ¶[0149]; EX1006, ¶¶[0006], [0033]-[0034]. A TPM would have provided enhanced hardware-based security features, such as hardware root of trust. EX1005, ¶[0110]. EX1010, ¶¶99-100.

Ball and Freund disclose similar goals of enhanced trust and security and similar mechanisms of a challenge/response protocol to determine the trustworthiness of a device. EX1005, ¶¶[0071], [0089]-[0089], [0110],[0122]-[0127]; EX1006, ¶¶[0004], [0031]-[0032]. A POSITA would have recognized that Ball's TPM would have been an appropriate hardware solution for implementing the

security measures disclosed by Freund in a manner consistent with Freund's hardware/software configuration. See EX1006, ¶¶[0006], [0031]. Ball, like Freund, discloses verifying attributes of a device, including security status of a device using a request/response protocol to determine whether a device can be trusted. EX1006, ¶¶[0002], [0030]-[0033]. Substituting Ball's TPM to provide the hardware support for Freund's security functions would have been entirely consistent with their shared goal of network security. EX1010, ¶¶101-104.

Employing a TPM in Freund's system would therefore have been obvious, predictable and beneficial for several reasons. First, such a combination would have involved the simple substitution of Ball's known TPM implementing the TCG Specification for one of the processors disclosed in Freund to execute the client-side security module. Second, such a combination would have constituted the obvious combination of prior art elements (the TPM according to the TCG Specification taught by Ball) with known methods (Freund's techniques) to yield predictable results. Third, such a combination would have involved using a known security component (a TPM according to the TCG Specification taught by Ball) as the hardware to improve the implementation of Freund's security functions thereby yielding beneficial results. Fourth, it would have been obvious to try the TPM from among the finite number of identified, predictable hardware solutions for executing

software security modules of the type disclosed by Freund with a reasonable expectation of success. EX1010, ¶¶105-109.

The combination of Freund and Ball disclose a TPM under any construction. See EX1014, p.2; EX1015, p.1. See EX1006, ¶[0006]; EX1008 in its entirety. EX1010, ¶110.

c. [1.3]: “receiving a response, and”

Freund discloses and suggests Element [1.3]. Freund’s router-based CMP interrogates the host device and receives a response from the host device’s TCB. See Element [1.2], *supra*. The CMP includes a router compliance table to determine whether the device is compliant. EX1005, ¶¶[0077]-[0078], [0084], [0086]. EX1010, ¶¶111-113.

d. [1.4]: “determining whether the response includes a valid digitally signed attestation of cleanliness,”

Freund in view of Ball discloses and suggests Element [1.4]. As discussed, Freund discloses a CMP. EX1005, ¶¶[0121]-[0122]. See Elements [1.2], [1.3], *supra*. Freund discloses determining whether the response includes a valid digitally signed attestation of cleanliness. Freund’s CMP interprets the response from the device to determine whether the response is valid. *Id.*, ¶[0144]. The table in [0144] shows “client application incorrect or old”, “AVold version”, and “AV Real-Time Monitor not running on client.” *Id.* Freund also discloses that applications (e.g., anti-virus software and versions) can be verified via digitally signed attestations using a

cryptographic hash function, to attest that the application has not been tampered with. *Id.*, ¶[0110]. The digital signature would have been obvious to include in any attestation, as part of the CMP function discussed for example at [0121], [0122]. EX1010, ¶¶114-119.

To the extent the digitally signed attestation is required to be performed by the TCB of a “TPM”, Freund and Ball in combination disclose Element [1.4]. Ball discloses the TPM transmitting a digitally signed attestation of cleanliness in the response. Freund’s system, with its security features, would have suggested to a POSITA to implement its TCB on a TPM as disclosed by Ball. *See* Element [1.2], *supra*. Ball discloses that “[t]o ensure that the quoted value is not corrupted during communication” the TPM “can generate an attestation identity key (AIK) that is used to encrypt some or all of the quoted value.” EX1006, ¶[0033]. The TCG Specification, incorporated by reference in Ball (*id.*, ¶[0006]), confirms that “[a]ttestation by the TPM is an operation that provides proof of data known to the TPM. This is done by **digitally signing** specific internal TPM data using an attestation identity key (AIK).” EX1008, p.6 (emphasis added). Freund’s system would have implemented the TPM with its client-side security module (i.e., the TCB and/or part of the TCB) to provide digitally signed attestations from the TCB associated with the TPM in the response to the CMP that a device is not infested, is security compliant, or is running anti-virus software that is sufficiently updated.

Freund and Ball in combination disclose determining whether the response includes a valid digitally signed attestation of cleanliness. EX1010, ¶¶120-126.

Freund and Ball disclose a valid digitally signed attestation of cleanliness, signed by and received from a TCB of a TPM. Freund and Ball disclose a valid digitally signed attestation of cleanliness under any construction. *See* EX1014, p.2; EX1015, p.6. *See also* EX1024, pp.7-11. EX1010, ¶¶127.

- e. [1.5]: “wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host;”

The cited art discloses and suggests Element [1.5]. Freund discloses that the valid digitally signed attestation of cleanliness indicates that the first host is not infested (e.g., a router CMP challenge verifies that the first host is running an anti-virus program, *see*, e.g., EX1005, ¶¶[0110]-[0117], [0132]-[0133]), and receives an attestation of the presence of an anti-virus version update - i.e., patch or a patch level associated with a software component of the first host (*see*, e.g., EX1005, ¶¶[0078], [0085]). EX1010, ¶¶128-129.

Freund discloses an “anti-virus challenge” option” in which “the router-side security module looks for the appropriate code to verify if the anti-virus program is running on the client machine and if both the anti-virus program and the associated

data file are up to date.” EX1005, ¶¶ [0132]-[0133]. *See also id.*, ¶ [0117] (“Anti-virus Real-Time monitoring not enabled. Informs user to activate Real-Time monitoring.”). Freund’s valid digitally signed attestation of cleanliness (e.g., a hashed, signed application preventing substitution of applications) ensures that client-side applications (including anti-virus applications) cannot be tampered with, thereby creating a level of trust in a client’s response to the router’s CMP anti-virus challenge. EX1005, ¶[0110]. Freund’s valid digitally signed attestation of cleanliness attests that the first host is not infested by attesting that the anti-virus software is running and/or that anti-virus real-time monitoring is enabled. EX1005, ¶¶[0110]-[0117]. EX1010, ¶¶130-132. Moreover, “performing a virus scan” was a well-known and conventional technique used in security and authentication processes, and it would have been obvious to perform such a virus scan and attest to same given that the purpose of such virus scan is to “ensure or restore the integrity of the content of the device”. EX1007, ¶¶[0052]-[0054], ¶[0047]. EX1010, ¶133.

The ’705 Patent discloses that a first host that is “not infested” (i.e., cleanliness) includes “a version associated with a current anti-contagion software or definition file in use, wherein a sufficiently updated software and/or scan may act as a cleanliness assertion.” EX1001, 14:16-19. The ’705 Patent discloses that “infestations”, “contagion”, and “viruses” are related concepts and anti-contagion software encompasses anti-virus software. EX1001, 11:20-22 (“Infestation herein

refers to the current presence and/or execution of contagion”), 3:30-36 (“Examples of contagion include computer worms, viruses”), 3:36-45 (“Anti-contagion software refers herein to software that prevents, impedes, or remediates contagion, such as Norton Antivirus from Symantec, or VirusScan from McAfee, which identifies and/or removes contagion from a user’s computer”). Thus, Freund’s device transmits a digitally signed attestation of cleanliness by asserting whether the anti-virus software is currently running and/or performing real-time monitoring of the client device to Freund’s CMP. EX1005, ¶¶[0110], [0144]. EX1010, ¶¶134-136.

Freund also discloses that the attestation validates the presence of an anti-virus version update level as a patch or a patch level associated with a software component on the first host. Freund’s attestation of cleanliness confirms that anti-virus software is updated with any recent updates represented by patch or patch level updates to the anti-virus software. EX1005, ¶¶[0078], [0085]. Freund discloses that “a computer running an *older version* of the security software may respond in the negative to a router challenge requesting confirmation that the computer is running a *current* version of the software”. EX1005, ¶[0078] (emphasis added). EX1010, ¶¶137-139.

Freund’s Figure 4 shows a patch or patch level for the TRENDMICRO anti-virus software PC-Cillin and ZAP with a “version” that Freund’s router inspects and enforces via the digitally signed attestation of cleanliness. EX1005, Fig. 4, ¶¶[0098]-[0099], [0132]. EX1010, ¶140. If the correct update (i.e., patch) level for these

software components is not confirmed, Freund's router will quarantine the client device and redirect its communications to the sandbox server to remediate and update the anti-virus software. EX1005, ¶¶[0114]-[0117]. *See also* EX1021, 2:34-37. *See also id.*, 2:49-52 (“many applications provide downloadable access to updates and patches”). The '705 Patent describes patches in a similar manner. EX1001, 3:16-17 (“download a software patch, update, or definition”). EX1010, ¶141.

Freund discloses that “a computer running an *older version* of the security software may respond in the negative to a router challenge requesting confirmation that the computer is running a *current* version of the software”. EX1005, ¶[0078] (emphasis added). *See also* EX1009, ¶[0019] (“A versioning table contains a list of root file numbers and version numbers. This information is used to track application patches and upgrades. Each entry in the versioning table corresponds to one patch level...”). EX1010, ¶142.

Freund discloses both: (1) an attestation that the TCB has ascertained that the first host is not infested, and (2) an attestation that the TCB has ascertained the presence of software updates via a patch or a patch level associated with a software component on the first host. Freund therefore discloses limitation [1.5] under any construction. *See* EX1014. EX1010, ¶143.

- f. [1.6]: “when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network,”

Freund discloses and suggests Element [1.6]. Freund discloses detecting an insecure condition such as outdated anti-virus software (*see* Element [1.2], *supra*) on a host by using a router-based CMP challenge/response protocol. EX1005, ¶[0041]. Non-compliant devices in Freund are redirected to the “sandbox” quarantine server. Freund’s “security module only allows the non-compliant computer to access the sandbox server to perform a defined set of tasks to address the non-compliance.” *Id.* Freund prevents the quarantined device from sending any data to any devices (including one or more other hosts associated with Freund’s network) other than the sandbox server or to a server providing updates to the out-of-date anti-virus software. *Id.*, ¶¶[0042], [0071]. EX1010, ¶¶144-148.

Freund’s quarantining includes isolating a vulnerable host device from a protected network. Freund discloses quarantining a device under any construction. *See* EX1014, p.2. EX1010, ¶149.

- g. [1.7]: “wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host,”

Freund discloses and suggests Element [1.7]. Freund discloses receiving a service request from a host device and preventing the device from sending data to

one or more other hosts. Freund discloses using service requests to request services from a network. EX1005, ¶¶[0042], [0147]. Freund’s “router receives a request for connection”. *Id.* Freund discloses its “sandbox server to which non-compliant computers are re-directed when they attempt to connect”. *Id.*, ¶[0081]. If a device is non-compliant, Freund “operates to redirect the local computer to the sandbox server instead of the address originally requested.” *Id.*, ¶[0089]. Freund prevents the device from sending data to one or more other hosts associated with the protected network upon receiving and redirecting the service request to the sandbox server. EX1010, ¶¶150-155.

Freund also discloses preventing the non-compliant host device from sending data on the network. Freund discloses a network connection being “denied in the event that the destination address is determined at step 960 not to be the DNS or DHCP server.” EX1005, ¶[0151]. Freund also denies a network connection where the service request is redirected to the sandbox server. *See id.*, Fig. 9, step 970. EX1010, ¶¶156-157.

h. [1.8]: “serving a quarantine notification page to the first host when the service request comprises a web server request, and”

Freund discloses and suggests Element [1.8]. Freund discloses receiving a service request in a quarantine-capable network. *See* Element [1.7], *supra*. Freund discloses that a service request is redirected to the sandbox server so the sandbox

server can transmit a quarantine notification page to the device when the service request is a web server request. EX1005, Fig. 7, Fig. 8, ¶¶[0049]-[0050], [0114]-[0117], [0148]-[0151]. EX1010, ¶¶158-160.

Freund discloses serving a quarantine notification page to a device in response to a service request when the device is quarantined and/or deemed non-compliant. EX1005, ¶[0149]. Freund discloses serving “an error message window 700 that is displayed to a non-compliant client computer that is redirected to the sandbox server.” *Id.*, ¶[0114]. Freund’s quarantine notification page “informs the user that he or she needs to update the virus protection software installed on the computer.” *Id.*, ¶[0114]; EX1010, ¶¶161-162.

Freund’s quarantine notification page is served to the device when the service request comprises a web server request (e.g., an HTTP request). Freund’s routing component monitors the service request and “determines whether or not the destination port is HTTP (port 80 TCP). If the destination port is HTTP, then the re-routing manager proceeds in step 951 to manipulate the destination IP address and port.” EX1005, ¶[0148]. A POSITA would have understood that a service request having a destination port of HTTP would include a web server request because it is well known that HTTP is a standard protocol for web server requests. EX1010, ¶¶163-166.

When Freund determines the service request is a web server request, Freund responds by replacing the destination Internet Protocol (“IP”) address with the IP address of the sandbox server which then serves the quarantine notification page to the device. EX1005, ¶¶[0149]. “Using this information, the sandbox server then displays a page with information enabling the client to address the specific problem that was detected.” *Id.* “In this manner, the connection request from a non-compliant client computer is patched and manipulated to reroute this packet to the sandbox server.” *Id.* Freund discloses serving a quarantine notification page to the device when the service request comprises a web server request. EX1010, ¶¶167-168.

- i. **[1.9]: “in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition; and”**

Freund discloses and suggests Element [1.9]. Element [1.9] was the stated basis for allowance during prosecution because the prior art considered at that time did not disclose providing, in response to a DNS query, an IP address of a quarantine server for serving a quarantine notification page to remedy an insecure condition. EX1002, pp.19-21. In Freund, where a service request comprises a DNS query, Freund discloses providing in response an IP address of the “sandbox” quarantine server configured to serve the notification page of Figure 7 or 8. This response is

provided when the DNS query is not associated with a remediation host. EX1010, ¶¶169-171. Freund therefore provides rerouting of an insecure client device to the IP address of the sandbox server for quarantining and remediation. EX1005, Figs. 7-9, ¶¶[0049]-[0050], [0114]-[0117], [0147]-[0151]. EX1010, ¶172.

Freund discloses a service request including a DNS query transmitted to a DNS server. EX1005, ¶[0150]. However, Freund's system prevents the host device from contacting the subject of the DNS query by providing, in response, the IP address of the "sandbox" quarantine server if the host name that is the subject of the DNS query is not associated with a remediation host. *Id.*, ¶[0149]. EX1010, ¶¶173-174. Freund replaces the destination IP address from the DNS query with the IP address of the sandbox server to redirect the request to the sandbox server. *Id. See also* EX1005, ¶[0150]. EX1010, ¶175.

Freund's providing of the sandbox server's IP address satisfies limitation [1.9] at least because the '705 Patent acknowledges that the redirect to the quarantine server "may be accomplished in numerous ways". EX1001, 15:1-12; 15:57-16:31. The Examiner's reasons for allowance focused on the claim language of serving a quarantine notification page following a DNS query not associated with remediation. EX1002, pp.17-21. Freund squarely addresses this by providing, in response to a DNS query, the IP address of the sandbox quarantine server to serve the quarantine

notification page and effect remediation. Freund discloses the stated reasons for allowance. EX1010, ¶¶176-178.

Freund discloses using a router to provide the IP address of the sandbox server in response to a DNS query (EX1005, ¶¶[0149]-[0150]. EX1010, ¶179). Freund's providing of an IP address of a quarantine server in response to a DNS query involves a rerouting of the client device that transmitted the DNS query to the sandbox server via the DNS server as per exemplary embodiments of the '705 Patent specification. EX1005, ¶[0150] ("the re-routing manager proceeds to step 960 to evaluate whether or not the destination port was DNS or DHCP and the destination IP address that of the DNS/DHCP server."). Non-compliant client devices are permitted to access the DNS server, otherwise "the browser would fail prior to being redirected to the sandbox server because it could not lookup the IP address of the sandbox server." *Id.* Freund's system therefore permits a non-complaint device to access the DNS server as part of the process for providing the IP address of the sandbox server in response to a DNS query. EX1010, ¶¶180-182. The DNS query and an HTTP request are used to access a web server and/or a website including a quarantine server. EX1010, ¶183. The IP address of the sandbox server is replaced in an HTTP request when the IP address that was the original subject of the DNS query is not associated with a remediation host. EX1010, ¶184.

Although Freund satisfies limitation [1.9], should it be argued that the claim is somehow limited to the non-limiting exemplary embodiments disclosed in the '705 specification, Freund's combination with Lewis satisfies any asserted construction of the claim language. Lewis discloses that, for a non-compliant device accessing a DNS server as per Freund, an alternate process to that disclosed by Freund for providing the IP address of a quarantine server at a router is to provide the IP address of the quarantine server at the DNS server. Lewis discloses that "[w]henever the client performs name resolution on any address [*i.e.*, a DNS query], it will receive the IP of QS [*i.e.*, a quarantine server]." EX1013, 14:22-23. Lewis' clients are redirected to a fix-up page [*i.e.*, a quarantine notification page] on the quarantine server. *Id.*, 15:24. *See also id.*, 9:1-13. Lewis thus discloses an alternate technique of DNS redirection to a quarantine server by providing the IP address of a quarantine server at a DNS server rather than at a router. EX1010, ¶¶185-186.

It would have been obvious to provide the IP address of Freund's sandbox quarantine server in response to Freund's DNS query by a DNS server as disclosed by Lewis, rather than at a router as disclosed by Freund. Such a substitution of an IP address at a DNS server rather than at a router would have been an obvious and predictable substitution for Freund's expressly disclosed providing of the sandbox server IP address. Each of Freund, Lewis, and the '705 Patent are concerned with

achieving the goal of DNS redirection of a non-compliant client device to a quarantine server. EX1010, ¶187.

Freund recognizes that non-compliant client devices must be permitted to access DNS to retrieve an IP address prior to being redirected to the sandbox server. EX1005, ¶[0150]. Freund discloses replacing an IP address from the DNS server with the IP address of the sandbox server when the IP address is not associated with remediation. *Id.*, ¶[0149]. Lewis teaches that the IP address of the sandbox server in Freund could be provided at the DNS server itself, rather than at the router. EX1010, ¶¶188-190.

Substituting Lewis’s DNS server to provide the IP address of the sandbox server in Freund would have been an obvious substitution of one known alternative for another to obtain predictable results of redirecting the non-compliant device to the IP address of the quarantine server. Lewis teaches one example of substituting the IP address of the quarantine server at the DNS server in a manner as described in an exemplary embodiment of the ’705 Patent (*see* EX1001, 15:1-5), while Freund discloses another example of replacing the IP address at a router as one of the “numerous ways” referenced by the ’705 Patent (*see* EX1001, 15:1-3; 16:1-3). EX1010, ¶191.

To provide the redirect to the sandbox server, Freund’s system determines whether the service request is associated with a remediation request by filtering the

request based on a destination address. EX1005, Fig. 9 (step 920). Freund's router keeps a list of destination addresses that are used for remediation in order to filter requests that are directed to remediation. *See also*, Freund's U.S. Provisional Application No. 60/303,653. EX1012, p.27. "If a computer is not in compliance, then the computer's access to the Internet is restricted to those activities necessary to get the computer back into compliance. This is accomplished by redirecting the attempted connection by a non-compliant computer to a designated 'sandbox' server that can facilitate appropriate corrective action, including the download of appropriate software to correct the non-compliance." EX1005, ¶¶[0071]. EX1010, ¶¶192-193.

Freund's sandbox server is configured to serve the quarantine notification page. Freund discloses a separate remediation host that is accessed by non-compliant devices via links provided on the quarantine notification page. EX1005, Figs. 7, 8, ¶¶[0114]-[0117]. A non-compliant device accesses the download link to a remediation host by transmitting a new HTTP request that is permitted by Freund's router as an exempt request from quarantined communication. EX1005, Fig. 9, 920. *See* EX1010, ¶¶194-195.

Freund's Figure 4 confirms that updates are provided via separate remediation hosts at least because the router and sandbox server require checking for specific versions of anti-virus software. If Freund's sandbox server provided remediation

updates, Freund's software and router would not require the illustrated tracking of licenses and specific anti-virus versions on the client devices. EX1010, ¶¶196-197. Thus, Freund discloses a third-party host (e.g., TRENDMICRO) from which to download anti-virus remediation updates. EX1010, ¶198. The anti-virus product "TRENDMICRO" illustrated in Freund is a separate product. Downloading updates for the TRENDMICRO antivirus product would involve contacting a TRENDMICRO host. EX1010, ¶199.

Freund's client devices would have accessed the "download" link using the "Update Now!" or "Download Now!" buttons on the notification page. EX1010, ¶200. Freund's sandbox server would have retrieved the anti-virus update from a data file, database, or server (e.g., via TCP). *See* EX1005, ¶¶[0035]-[0036]. EX1010, ¶201. The '705 Patent discloses links provided through the quarantine notification page to the same extent as Freund's sandbox server. *See* EX1001, 16:2-9 ("provides links to remediation sites appropriate to the quarantine"; "The links on the quarantine notification page may be examples of HTTP addresses that are for use in remediation."). EX1010, ¶202.

To the extent this limitation is interpreted under §112(f)/¶6, Freund discloses corresponding structure at least to the same extent as the '705 Patent. Freund discloses a quarantine server and remediation host under any construction. *See* EX1014, pp.2-3. EX1010, ¶203.

Element [1.9] served as basis for allowance during prosecution. *See* EX1002, pp.19-21. The Examiner’s reasons for allowance explained “[t]he closest prior art, Liang (US 7287278 B2) and Yan et al. (US 2005/0033987 A1) disclose detecting abnormal events on host, and redirecting to quarantine serve[r] to get remediated; however, they fail to anticipate or render serving both the specific quarantine notification page with the DNS redirection when in combination with the remaining claim limitations.” Freund, alone and/or in combination with Lewis, discloses this exact function under any construction. EX1010, ¶¶204-206.

j. [1.10]: “permitting the first host to communicate with the remediation host.”

Freund in view of Ball discloses and suggests Element [1.10]. Freund discloses permitting the device to communicate with a remediation host when the device is quarantined. Freund discloses that a non-compliant device redirected to the sandbox server is only permitted “to perform a defined set of tasks to address the non-compliance. All other Internet access by the non-compliant computer is disabled.” EX1005, ¶[0042], Fig. 9, ¶¶[0148]-[0151]; EX1010, ¶¶207-208.

Freund discloses that the non-compliant computer’s “access to the Internet is restricted to those activities necessary to get the computer back into compliance. This is accomplished by redirecting the attempted connection by a non-compliant computer to a designated ‘sandbox’ server that can facilitate appropriate corrective action, including the download of appropriate software to correct the non-

compliance.” EX1005, ¶[0071]. Freund allows a non-compliant computer to download appropriate corrective software from a remediation host or through the sandbox server itself acting as a remediation host. *See also id.*, ¶¶[0078], [0095]. Figures 7-9 of Freund also show and describe the notification page providing update and download links to remediation hosts to retrieve data to remedy the non-compliant devices. *See id.*, Figs. 7-9; ¶¶[0114]-[0115]. EX1010, ¶¶209-211.

2. **Claim 2: “A method as recited in claim 1, wherein detecting an insecure condition further includes at least one of the following: scanning for a vulnerability, scanning for malicious data, determining whether a security software is installed, and detecting anomalous network traffic.”**

Freund in view of Ball discloses and suggests claim 2. Freund discloses detecting an insecure condition at least by “scanning for a vulnerability”, “determining whether a security software is installed”, and “detecting anomalous network traffic” as claimed. Freund discloses a component that “checks to ensure that appropriate end point security software is in place on all of the computers on the LAN.” EX1005, ¶[0071]. Freund discloses verifying “that the computer has installed and is running appropriate security software, and is in compliance with other established security policies.” *Id.* A device that is not in compliance with security policies exhibits a vulnerability. EX1010, ¶¶212-215. Freund discloses and suggests scanning (e.g., checking) for a vulnerability by determining whether a device is running appropriate security software and complies with established security

policies. Freund also discloses determining whether security software is installed by determining that the device has installed and is running appropriate security software. EX1005, ¶¶[0019], [0071]. An outdated application (e.g., including an OS) and/or security software would be a vulnerability. *Id.*, ¶[0110]. EX1010, ¶¶216-218.

Freund also discloses that “detecting an insecure condition” includes “scanning for malicious data”. Freund discloses an “anti-virus challenge” that “allows the administrator to use the router for anti-virus enforcement and distribution. The router-side security module looks for the appropriate code *to verify if the anti-virus program is running* on the client machine and if both the anti-virus program and the associated data file are up to date.” EX1005, ¶[0132] (emphasis added). *See also id.*, ¶[0068] (implementing security policies that serve to avoid or reduce the impact of “attacks from malicious code”). Indeed, it was well known to “perform[] a virus scan” to detect whether an insecure condition exists, and it would have been obvious to perform such a virus scan for this purpose. EX1007, ¶[0047], ¶¶[0052]-[0054]. EX1010, ¶¶219-220.

Freund also discloses “detecting anomalous network traffic” as claimed at least because Freund discloses checking network requests for compliance by evaluating responses in a router compliance table and determining whether network traffic includes HTTP requests and/or DNS requests. EX1005, ¶¶[0147]-[0150]. By determining a network request and therefore a device is non-compliant with the

router compliance table, Freund discloses and suggests “detecting anomalous network traffic” as claimed. EX1010, ¶¶221-222.

Freund contemplates using “one or more of the security policy requirements indicated in the router challenge” and thus teaches use of any combination of the security policy requirements described above. EX1005, ¶[0078]. *See also id.*, ¶[0093] (“any optional security requirements”). EX1010, ¶223.

3. Claim 3: “A method as recited in claim 1, wherein detecting an insecure condition includes determining that the first host should be quarantined until an update to an operating system has been installed.”

Freund in view of Ball discloses and suggests claim 3. Freund discloses detecting an insecure condition including determining that the first host should be quarantined until an update to an operating system (“OS”) has been installed. Freund discloses that a device is quarantined until software on the device has been sufficiently updated to correct non-compliance. EX1005, ¶[0071], ¶¶[0117]-[0118]. Freund recognizes network security vulnerabilities associated with devices running “older software”. *Id.*, ¶¶[0088]-[0089]. It was well-known and conventional to verify the version of an OS for security and attestation purposes. *See, e.g.*, EX1006, ¶[0031] (“verifies” the “type and/or version” of an “operating system” for purposes of “attestation”). Thus, Freund’s teachings concerning the insecure condition associated with out-of-date software is not limited to anti-virus software but also

applies to out-of-date OSs—indeed, Freund specifically warns against OS “security holes”. EX1005, ¶¶[0015], [0019]. EX1010, ¶¶224-228.

4. Claim 4: “A method as recited in claim 1, wherein detecting an insecure condition includes configuring an operating system to quarantine the first host upon initial startup after installation of the operating system.”

Freund in view of Ball discloses and suggests claim 4. Freund discloses detecting an insecure condition including configuring an OS to quarantine the first host upon initial startup after installation of the OS. Freund specifically warns against OS “security holes”. EX1005, ¶¶[0015], [0019]. Freund contemplates that OSs can result in security vulnerabilities or security non-compliance such that the OS must be up-to-date to avoid such vulnerabilities. *See* Claim 3, *supra*. Freund expressly recognizes that a “user may inadvertently disable previously installed security software in the process of upgrading his or her operating system.” EX1005, ¶[0019]. Freund recognizes that security software can be disabled when installing and/or upgrading an OS. A POSITA would have been motivated to use Freund’s security compliance system to configure an OS on the client-side device to quarantine a device upon initial startup installation to solve this problem recognized by Freund and to check for disabled or altered security software upon initial startup to ensure a device remains compliant after installation of a new OS. *See also*, EX1006, ¶[0031]; EX1010, ¶¶229-234.

Freund also discloses quarantine of a device upon initial startup. *See* EX1005, ¶[0125]. Freund recognizes out-of-date OSs posing security threats, problems associated with disabling security software upon installation of an OS, and Freund contemplates quarantine upon initial startup of a device. It would have been obvious to a POSITA that Freund discloses configuring an OS to quarantine a device upon initial startup to ensure the device's OS is up-to-date and in compliance with Freund's security policies after installation. A POSITA would have had a reasonable expectation of success given that Freund informs a POSITA that installation and/or upgrade of OSs can pose a threat to a device's security compliance, and Freund is directed to ensuring networked devices remain in compliance with established security policies. *Id.*, ¶[0071]; EX1010, ¶¶235-238.

5. **Claim 5: “A method as recited in claim 1, wherein permitting the first host to communicate with the remediation host includes: detecting an outbound communication from the first host; and forwarding the outbound communication if it is addressed to the remediation host.”**

Freund in view of Ball discloses and suggests claim 5. Freund discloses receiving a service request from a device. *See* Element [1.7], *supra*. Freund discloses detecting an outbound communication from the device (i.e., as a service request). *See* Element [1.7], *supra*. Freund discloses forwarding the outbound communication if it is addressed to the remediation host to permit the device to communicate with the remediation host at least because Freund discloses that a device's activities are

restricted to activities necessary to get the device back into compliance with security policies when the device is redirected to the sandbox server. EX1005, ¶[0071]. Figure 9 of Freund discloses a process for forwarding an exempt address referenced in block 920 via block 980 (e.g., the exempt address of a remediation host such as the PC-Cillan anti-virus software referenced in Freund's Figure 4, available from TRENDMICRO referenced in EX1005, ¶[0132]. *See*, EX1005, ¶[0147]-[0151]; [0104]-[0105]. EX1010, ¶¶239-243.

Freund redirects communication to the sandbox server or permits communication if it is addressed to remediation "to address the specific problem that was detected." EX1005, ¶[0149]. *See id.*, ¶¶[0071], [0149], Fig. 9. EX1010, ¶244.

Freund discloses that requests (e.g., HTTP requests to other services) are redirected to the sandbox server for a non-compliant device. EX1005, ¶¶[0042], [0078], [00147]-[0150]. Freund's routing component monitors the service request and "determines whether or not the destination port is HTTP (port 80 TCP). If the destination port is HTTP, then the re-routing manager proceeds in step 951 to manipulate the destination IP address and port." EX1005, ¶[0148]. Freund discloses that such requests to connect to other services are "redirected to the sandbox server." *Id.*, ¶¶[0040], [0042], [0071]. Freund discloses allowing the non-compliant device to access the sandbox server to perform a defined set of tasks to address non-compliance, such as downloading appropriate software to correct non-compliance.

Id., ¶¶[0071], [0078]. Freund allows a non-compliant device to access remediation services. *See, e.g., id.*, Fig. 7. Any other communications from a non-compliant device are redirected to the sandbox server (i.e., quarantine server). *Id.*, ¶¶[0148]-[0150]. EX1010, ¶¶245-249.

Freund's system "enables the user to take the steps necessary to bring his or her computer back into compliance." EX1005, ¶[0095]. Prompts such as "Update Now" in Figure 7 and "Download Now" in Figure 8, when selected, constitute service requests for accessing virus protection software updates "associated with a remediation request" as claimed. *Id.*, Fig. 7 (prompting user to select "Update Now!" button for remediation), ¶[0114] ("The message displayed in panel 702, as shown in FIG. 7, informs the user that he or she needs to update the virus protection software installed on the computer."); Fig. 8 (prompting user to select "Download Now!" button for remediation), ¶[0116] ("error message panel 802 is displayed to the user indicating that there is a new version of the security software available and prompting the user to download the new version"). The '705 Patent works this same way. *See, e.g.,* EX1001, 16:2-7 ("An example of a quarantine notification page is a web page that provides notification that the computer is quarantined, and/or provides links to remediation sites appropriate to the quarantine, such as a link to a site that provides anti-contagion software for removing a virus that the quarantined computer is believed to contain."). EX1010, ¶¶250-251.

A non-compliant computer is “permitted **only** a limited Internet connection to the sandbox server” where “the security module only allows the non-compliant computer to access the sandbox server **to perform a defined set of tasks to address the non-compliance. All other Internet access by the non-compliant computer is disabled.** [Emphasis added.]” EX1005, ¶[0042]. For example, when the user selects the “Update Now” and “Download Now” prompts described above, Freund’s system forwards requests made by the non-compliant device to download updates (e.g., anti-virus software updates) given that such system-generated prompts are for the purpose of addressing the non-compliance. EX1010, ¶¶252-253.

Freund’s sandbox server provides a notification page including a link to a remediation service (e.g., “Update Now” and “Download Now” prompts). When a device attempts to download an anti-virus software update using the prompts, Freund’s client will generate a new outbound communication request. Freund’s router and CMP determine that the new outbound communication is an exempt request for remediation, thus forwarding the outbound communication to a remediation host providing the download and/or update for remediation. *See* EX1005, Figs. 7 and 8, ¶¶[0042], [0071], [0114]-[0116]. EX1010, ¶254.

Freund filters a remediation request based on a destination address. EX1005, Fig. 9 (step 920). *See also*, Freund’s U.S. Provisional Application No. 60/303,653; EX1012, p.27. Freund discloses determining whether a destination address (e.g., to

a remediation host) is exempt. EX1010, ¶¶255-256; EX1005; Fig. 7, ¶¶[0042], [0071], [0114]. Freund filters service requests (e.g., based on destination addresses) and forwards requests for remediation by establishing policy rules for permitting service requests to specific sites for remediation. EX1005, ¶[0110] (“For example, rules can also be established on the basis of including and/or excluding access to particular Internet sites.”). EX1010, ¶257.

The ’705 Patent describes the same process disclosed by Freund. EX1001, Fig. 14, 14:50-15:12. The ’705 Patent describes testing outbound traffic to determine whether the traffic is associated with remediation such as “contact with a verified remediation site”. *Id.*, 14:53-59. If the traffic is associated with a remediation request, then the traffic is forwarded to the destination. *Id.*, 14:59-61. Otherwise, the traffic is rerouted to the quarantine server. *Id.*, 14:59-15:5. Freund discloses detecting an outbound communication from the first host; and forwarding the outbound communication if it is addressed to the remediation host. EX1010, ¶¶258-260.

6. **Claim 6: “A method as recited in claim 1, wherein preventing the first host from sending data to the one or more other hosts includes: detecting an outbound communication from the first host; and redirecting the outbound communication to a quarantine server if it comprises a request for an approved service and is not addressed to a remediation host.”**

Freund in view of Ball discloses and suggests claim 6. Freund discloses detecting an outbound communication from the device. *See* Claim 5, *supra*. Freund discloses redirecting the outbound communication to a sandbox server (i.e.,

quarantine server) if it comprises a request for an approved service and is not addressed to a remediation server. EX1010, ¶¶261-262. Freund discloses that outbound communication from a non-compliant device “is redirected and permitted only a limited Internet connection to the sandbox server. In this situation, the security module only allows the non-compliant computer to access the sandbox server to perform a defined set of tasks to address the non-compliance.” EX1005, ¶[0042]. Freund detects the outbound communication and determines whether the destination address for the communication is exempt (e.g., for remediation). EX1005, Fig. 9. Any other outbound communications (i.e., sending data to the one or more other hosts of Freund’s network) are redirected to the sandbox server at least because the communication are not detected as exempt. EX1010, ¶¶263-264.

“The re-routing manager operates to redirect the local computer to the sandbox server instead of the address originally requested.” EX1005, ¶[0089]. *See also id.*, ¶¶[0071], [0078], [0149]. Freund discloses detecting an outbound communication from the device and redirecting the outbound communication to a quarantine server if it comprises a request for an approved service and is not addressed to remediation. EX1010, ¶¶265-266.

7. **Claim 7: “A method as recited in claim 1, wherein quarantining the first host further includes preventing the first host from receiving via the protected network data not related to remediation of the insecure condition.”**

Freund in view of Ball discloses and suggests claim 7. Freund discloses that quarantining the device at the sandbox server includes preventing the device from receiving via the network data not related to remediation. Freund discloses that a non-compliant device “is redirected and permitted only a limited Internet connection to the sandbox server. In this situation, the security module only allows the non-compliant computer to access the sandbox server to perform a defined set of tasks to address the non-compliance. All other Internet access by the non-compliant computer is disabled.” EX1005, ¶[0042] (emphasis added). *See also id.*, ¶[0095]; claim 6, *supra*. By only permitting non-compliant devices to access the sandbox server and to perform defined tasks to address non-compliance, and disabling all other Internet access, Freund discloses and/or suggests preventing the device from receiving data not related to remediation of the insecure condition. EX1010, ¶¶267-270.

8. **Claim 8: “A method as recited in claim 1, performed at an Internet service provider.”**

Freund in view of Ball discloses and suggests claim 8. Freund discloses “computers are now connected to the Internet, either directly (e.g., over a dial-up or broadband connection with an Internet Service Provider or ‘ISP’) or through a

gateway between a LAN and the Internet”. EX1005, ¶[0008]. Freund’s Figure 3 shows router 310 connecting a client computer 320 to the Internet 350 and thus suggests that Freund’s method which is implemented with a router connection and a quarantine sandbox server 360, constitutes implementation of that method at Internet connection equipment of an ISP. *Id.*, Fig. 3. The ’705 Patent contemplates an “ISP web server” as the “special quarantine server” referenced in claim 1. EX1001,14:65-15:1; Fig. 14. EX1010, ¶¶271-274.

Freund’s system for network access management addresses common problems for ISPs such as attacks from malicious devices, unauthorized access, viruses, employee abuse of network systems, and network bandwidth. U.S. Patent No. 5,987,611 (EX1020) (“Freund ’611”), incorporated by reference in Freund (EX1005, ¶[0017]), discloses an “ISP-based embodiment” that is “implemented for establishing a monitoring and filtering system” for ISPs. EX1020, 21:47-52. *See also id.*, Fig. 3B. Both Freund ’611 and Freund are aimed at “maintaining security of individual computer systems and local area networks (LANs) connected to larger open networks (Wide Area Networks or WANs), including the Internet.” *Id.*, 1:25-30. EX1005, ¶[0004]. Freund expressly incorporates Freund ’611 with reference to a “prior ZoneAlarm™ product” (EX1005, ¶[0017]). Freund is also directed to a ZoneAlarm product. *Id.*, ¶¶[0018]-[0021]. *See also id.*, Figs. 4-8. EX1010, ¶¶275-277.

Freund '611 discloses that client devices can have "Internet access restricted to the ISP 'Sandbox' Server." EX1020, 28:65-67, 27:51-30:10. Freund '611 expressly performs Freund's methods at an ISP with an ISP sandbox server. It would have been obvious to implement Freund's router and sandbox server at an ISP to redirect service requests of non-compliant devices at least because Freund '611 explains that ISPs can "offer their users a tamper-proof, safe, and managed access to the Internet and protect[] the users from many security threats." *Id.*, 21:53-55, 21:47-22:41. EX1010, ¶¶278-279. Implementing Freund's router and sandbox server at an ISP would have been a simple implementation taught and/or suggested by Freund '611 that would have led to predictable results of providing managed network access from an ISP for the types of networks Freund seeks to protect. EX1005, ¶¶[0019]-[0020]. A POSITA would have had a reasonable expectation of success given that Freund '611 describes a successful implementation of the ZoneAlarm 1.0 product. *Id.*, ¶[0017]. *See* EX1020 in its entirety. EX1010, ¶¶280-281.

Freund also discloses using "an independent piece of equipment (such as a router or DSL modem)" and a quarantine server to implement its network access method. EX1005, ¶[0039]. DSL modems and/or routers and/or servers connect networks (e.g., private networks) to receive Internet access. EX1010, ¶¶282-283. *See also* EX1020, 21:59-22:6. Freund's techniques for access management are

performed at an ISP device, such as a DSL modem or an ISP router with a quarantine server to provide security policies and quarantine services to private networks. EX1010, ¶284. Freund therefore discloses and/or renders obvious performing the disclosed method at an ISP at least because devices within a protected network disclosed by Freund will attempt to connect to the Internet via an ISP service and related equipment. EX1010, ¶285. Freund discloses performing the method of claim 1 at an ISP interface to the Internet. EX1010, ¶286.

9. Claim 9: “A method as recited in claim 1, wherein the software component on the first host is an operating system.”

Freund in view of Ball discloses and suggests claim 9. Freund discloses the software component on the first host is an OS. Freund discloses that its devices “include [] a kernel or operating system (OS) 210.” EX1005, ¶[0063]. *See id.*, Fig. 2. Freund also discloses various security issues that can arise with OSs. *See id.*, ¶¶[0015], [0019]. *See also* Claim 4, *supra*. Freund specifically warns against OS “security holes”. EX1005, ¶¶[0015], [0019]. Freund discloses detecting devices having out-of-date software components, and Freund teaches that the software component can include an OS at least because OSs being out-of-date can pose a well-known security threat and will cause a device to be out of compliance with established security policies. Freund also teaches that other security policies can be used for enforcement. *Id.*, ¶¶[0103], [0110]. A device having an out-of-date OS would be non-compliant with Freund’s disclosed security policies and other possible

security policies. *See id.*, ¶¶[0068], [0071]-[0072], [0078], [0085]-[0089]; *see also*, EX1006, ¶[0031]. EX1010, ¶¶287-292.

10. Claim 10: “A method as recited in claim 1, wherein determining that the response does not include a valid digitally signed attestation of cleanliness includes determining that the software component on the first host is not sufficiently updated.”

Freund in view of Ball discloses and suggests claim 10. Freund discloses determining an insecure condition by determining a software component on the first host is not sufficiently updated (i.e., anti-virus software out-of-date). Freund discloses determining that a device is not clean when the CMP and/or router-side security module “looks for the appropriate code to verify if the anti-virus program is running on the client machine and if both the anti-virus program and the associated data file are up to date.” EX1005, ¶[0132]. *See also id.*, ¶¶[0114], [0117], [0144] (“ZAP version outdated 33 Client application incorrect or old.”), Figs. 7-8. Freund discloses determining that the response does not include a valid digitally signed attestation of cleanliness, where cleanliness includes that a software component (i.e., anti-virus software) on the device is sufficiently updated. EX1010, ¶¶293-294.

Kappes demonstrates that a POSITA would have understood Freund in view of Ball to disclose storing a content authentication token within the first host. “The content authentication token framework can and the token scheme be implemented with a trusted program (or a set of trusted programs) running on the client device

110.” EX1007, ¶[0049]. “This secure program can participate in the challenge/response protocol for content authentication.” *Id.* Kappes demonstrates that a POSITA would have understood Freund’s router challenge/response to access a digitally signed content authentication token to detect the insecure condition. According to Kappes, “[t]he trusted program can be provided, for example, on a Smart Card, driver or run inside a secure portion of the device 110.” EX1007, ¶[0049]. *See also* EX1008, pp.3-21. EX1010, ¶¶295-296.

The TCG Specification, incorporated by reference in Ball, also supports the attestation of Freund, as the TCG Specification expressly discloses a TPM and components thereof, stating that “[i]mplementations of TPMs may be done in hardware or software.” EX1008, p.19. TCG Specification also notes that a TPM is “a building block of a trusted platform”. *Id.* A diagram and a component architecture of a TPM are also presented. *Id.*

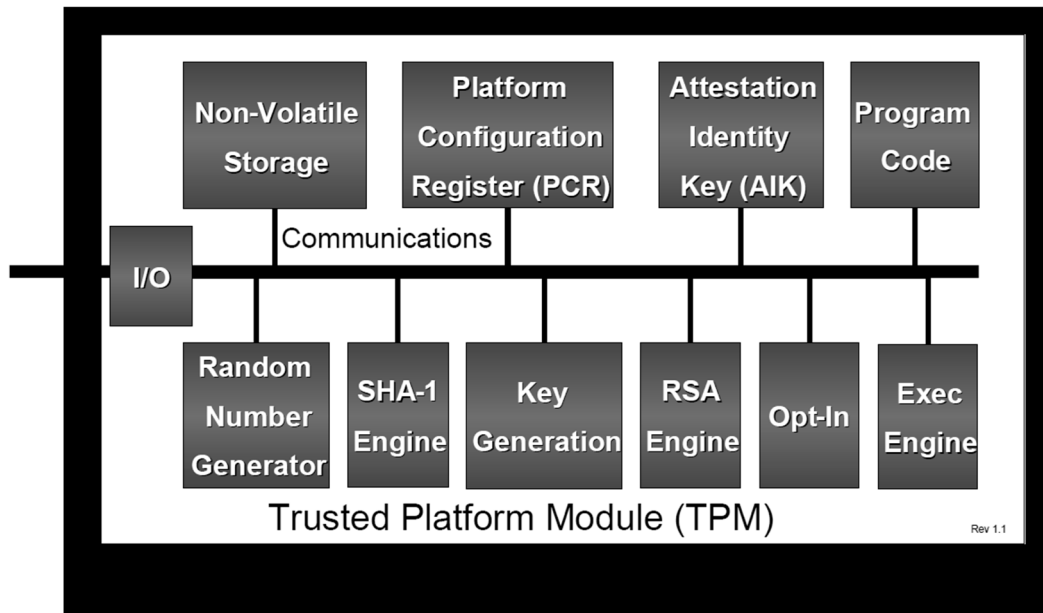


Figure 4:g – TPM Component Architecture

See also, EX1008, pp.19-26; EX1010, ¶¶297-298.

Implementing Ball’s TPM with its TCG Specification to provide the security features of Freund with a content authentication process and trusted program would have been a straightforward combination of known elements according to known methods to yield predictable results. EX1007, ¶[0049]; EX1010, ¶299.

11. **Claim 11:** “A method as recited in claim 1, wherein determining that the software component on the first host is not sufficiently updated includes determining that a patch level associated with the software component on the first host is not sufficiently recent.”

Freund in view of Ball discloses and suggests claim 11. Freund discloses determining that a software component on the device (e.g., anti-virus software) is not sufficiently updated. See Claim 10, *supra*. Freund discloses that “a computer

running an *older version* of the security software may respond in the negative to a router challenge requesting confirmation that the computer is running a *current* version of the software”. EX1005, ¶[0078] (emphasis added). Freund’s failed router challenge due to “running an older version” of the security software constitutes a disclosure and suggestion that “a patch level ... is not sufficiently recent.” *See also* EX1009, ¶[0019] (“A versioning table contains a list of root file numbers and version numbers. This information is used to track application patches and upgrades. Each entry in the versioning table corresponds to one patch level...”). EX1010, ¶¶300-301.

Freund discloses the software components of the first host include an OS. EX1005, ¶[0063] (devices “include[] a kernel or operating system (OS) 210.”), Fig. 2. *See* Claims 3/4, *supra*. Freund discloses that OSs can have “well-known security holes.” EX1005, ¶[0015]. Out-of-date anti-virus software and/or OSs including well-known security holes would violate security policies and that these software components would include a patch or patch level that is not sufficiently recent. EX1010, ¶¶302-304. Freund’s disclosure of updates to anti-virus software or OSs would encompass patches or patch levels. EX1010, ¶305.

12. Claim 12 (Preamble): “A system for protecting a network, comprising:”

To the extent the Preamble is limiting, Freund discloses and suggests the Preamble. Freund discloses a system for protecting a network. *See* Claim 1

(Preamble), *supra*. See also Fig. 3. EX1010, ¶¶306-307. Freund “provides a security system that delegates enforcement of certain security policies to software that is not running on a local computer but instead running on another piece of equipment” on the same LAN. EX1005, ¶[0068]. Freund’s Fig. 3 discloses a system including a router for monitoring client requests within the private network. *Id.*, ¶¶[0073]-[0074], Fig.3. Freund’s system includes a router executing the CMP connected to and serving multiple devices (e.g., personal computers) having the client-side security module stored thereon (where computer 330 is not running the client-side security module). *Id.* Freund’s system also includes the sandbox server residing somewhere on the Internet. *Id.* Freund’s system using the client-side security modules, the router and the CMP to monitor network traffic is configured to protect the devices on the private network from transmitting or receiving malicious or unauthorized data. *Id.*, ¶¶[0068]-[0072]. EX1010, ¶¶308-311.

a. [12.1]: “a processor configured to:”

Freund discloses and suggests Element [12.1]. Freund’s system can be “implemented on a conventional or general-purpose computer system” including “a central processing unit(s) (CPU) or processor (s)” where “any other suitable microprocessor or microcomputer may be utilized for implementing the present invention.” EX1005, ¶¶[0055]-[0056]. Freund discloses a system including a processor. EX1010, ¶¶312-314.

- b. [12.2]: “detect an insecure condition on a first host that has connected or is attempting to connect to a protected network,”**

Freund discloses and suggests Element [12.2]. *See* Element [1.1], *supra*.

EX1010, ¶315.

- c. [12.3]: “wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host,”**

Freund in view of Ball discloses and suggests Element [12.3]. *See* Element [1.2], *supra*. EX1010, ¶316.

- d. [12.4]: “receiving a response, and”**

Freund discloses and suggests Element [12.4]. *See* Element [1.3], *supra*.

EX1010, ¶317.

- e. [12.5]: “determining whether the response includes a valid digitally signed attestation of cleanliness,”**

Freund in view of Ball discloses and suggests Element [12.5]. *See* Element [1.4], *supra*. EX1010, ¶318.

- f. [12.6]: “wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host; ”**

Freund discloses and suggests Element [12.6]. *See* Element [1.5], *supra*.

EX1010, ¶319.

- g. [12.7]: “when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantine the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network,”**

Freund discloses and suggests Element [12.7]. *See* Element [1.6], *supra*.

EX1010, ¶320.

- h. [12.8]: “wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host,”**

Freund discloses and suggests Element [12.8]. *See* Element [1.7], *supra*.

EX1010, ¶321.

- i. [12.9]: “serving a quarantine notification page to the first host when the service request comprises a web server request, and”**

Freund discloses and suggests Element [12.9]. *See* Element [1.8], *supra*.

EX1010, ¶322.

- j. [12.10]: “in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition; and”**

Freund discloses and suggests Element [12.10]. *See* Element [1.9], *supra*.

EX1010, ¶323.

k. [12.11]: “permit the first host to communicate with the remediation host; and”

Freund discloses and suggests Element [12.11]. *See* Element [1.10], *supra*.

EX1010, ¶324.

l. [12.12]: “a memory coupled to the processor and configured to provide instructions to the processor.”

Freund discloses and suggests Element [12.12]. Freund discloses the system including a memory coupled to the processor and configured to provide instructions to the processor. EX1005, ¶¶[0055]-[0058]. EX1010, ¶¶325-326.

13. Claim 13: “A system as recited in claim 12, wherein the processor is configured to detect an insecure condition at least in part by performing one or more of the following: scanning for a vulnerability, scanning for malicious data, determining whether a security software is installed, and detecting anomalous network traffic.”

Freund in view of Ball discloses and suggests claim 13. *See* claim 2, *supra*.

EX1010, ¶327.

14. Claim 14: “A system as recited in claim 12, wherein the processor is configured to detect an insecure condition at least in part by determining that an initial startup after installation of an operating system is being performed.”

Freund in view of Ball discloses and suggests claim 14. *See* claim 4, *supra*.

EX1010, ¶328.

- 15. Claim 15: “A system as recited in claim 12, wherein the processor is configured to quarantine the first host at least in part by preventing the first host from receiving via the protected network data not related to remediation of the insecure condition.”**

Freund in view of Ball discloses and suggests claim 15. *See* claim 7, *supra*.

EX1010, ¶329.

- 16. Claim 16: “A system as recited in claim 12, wherein the software component on the first host is an operating system.”**

Freund in view of Ball discloses and suggests claim 16. *See* claim 9, *supra*.

EX1010, ¶330.

- 17. Claim 17: “A system as recited in claim 12, wherein determining that the response does not include a valid digitally signed attestation of cleanliness includes determining that the software component on the first host is not sufficiently updated.”**

Freund in view of Ball discloses and suggests claim 17. *See* claim 10, *supra*.

EX1010, ¶331.

- 18. Claim 18: “A system as recited in claim 17, wherein determining that the software component on the first host is not sufficiently updated includes determining that a patch level associated with the software component on the first host is not sufficiently recent.”**

Freund in view of Ball discloses and suggests claim 18. *See* claim 11, *supra*.

EX1010, ¶332.

19. Claim 19 (Preamble): “A computer program product for protecting a network, the computer program product being embodied in a non-transitory computer readable medium and comprising computer instructions for:”

To the extent the Preamble is limiting, Freund discloses and suggests the Preamble. Freund discloses a computer program product for protecting a network where the computer program product is embodied in a non-transitory computer readable medium and comprising computer instructions. EX1010, ¶¶333-334.

Freund discloses that “program logic” for its network protection system “is loaded from the storage device or mass storage 116 into the main (RAM) memory 102, for execution by the CPU 101.” EX1005, ¶[0058]. Freund’s storage device or mass storage storing program logic constitutes the computer program product embodied in a non-transitory computer readable medium. EX1010, ¶¶335-336.

a. [19.1]: “detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network,”

Freund discloses and suggests Element [19.1]. *See* Element [1.1], *supra*. EX1010, ¶337.

b. [19.2]: “wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host,”

Freund in view of Ball discloses and suggests Element [19.2]. *See* Element [1.2], *supra*. EX1010, ¶338.

c. [19.3] receiving a response, and

Freund in view of Ball discloses and suggests Element [19.3]. *See* Element [1.3], *supra*. EX1010, ¶339.

d. [19.4] determining whether the response includes a valid digitally signed attestation of cleanliness,

Freund in view of Ball discloses and suggests Element [19.4]. *See* Element [1.4], *supra*. EX1010, ¶340.

e. [19.5] wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host;

Freund discloses and suggests Element [19.5]. *See* Element [1.5], *supra*. EX1010, ¶341.

f. [19.6] when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network,

Freund discloses and suggests Element [19.6]. *See* Element [1.6], *supra*. EX1010, ¶342.

- g. [19.7] wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host,**

Freund discloses and suggests Element [19.7]. *See* Element [1.7], *supra*.

EX1010, ¶343.

- h. [19.8] serving a quarantine notification page to the first host when the service request comprises a web server request, and**

Freund discloses and suggests Element [19.8]. *See* Element [1.8], *supra*.

EX1010, ¶344.

- i. [19.9] in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition; and**

Freund discloses and suggests Element [19.9]. *See* Element [1.9], *supra*.

EX1010, ¶345.

- j. [19.10] permitting the first host to communicate with the remediation host.**

Freund discloses and suggests Element [19.10]. *See* Element [1.10], *supra*.

EX1010, ¶346.

VIII. OTHER CONSIDERATIONS

Any purported evidence of secondary considerations would be insufficient to overcome the strong evidence of obviousness. EX1010, ¶¶347-349.

IX. CONCLUSION

Petitioner has shown a reasonable likelihood of success. Therefore, this Petition should be granted and the Board should institute trial.

Date: September 19, 2025

Respectfully submitted,

/Patrick C. Keane/

Patrick C. Keane, Esq.

Registration No. 32,858

BUCHANAN INGERSOLL & ROONEY PC

1737 King Street, Suite 500

Alexandria, Virginia 22314

Direct Telephone (703) 838-6522

Main Facsimile (703) 836-2021

patrick.keane@bipc.com

Counsel for Petitioner

APPENDIX A - LIST OF EXHIBITS

Exhibit No.	Description
1001	U.S. Patent No. 8,234,705, issued July 31, 2012
1002	File History of U.S. Patent Application No. 11/237,003
1003	Not Assigned
1004	U.S. Provisional Application 60/613,909
1005	U.S. Patent Publication No. 2003/0055962, published March 20, 2003 to G. Freund <i>et al.</i> (“Freund”)
1006	U.S. Patent Application Publication No. 2006/0005009, published January 5, 2006 to C. Ball <i>et al.</i> (“Ball”)
1007	U.S. Patent Application Publication No. 2005/0111466, published May 26, 2005 to M. Kappes and P. Krishnan (“Kappes”)
1008	TCG Specification Architecture Overview
1009	U.S. Patent Application Publication No. 2002/0083183, published June 27, 2002 to S. Pujare <i>et al.</i> (“Pujare”)
1010	Declaration of Markus Jakobsson
1011	<i>Curriculum Vitae</i> of Markus Jakobsson
1012	U.S. Provisional Patent Application No. 60/303,653 filed July 6, 2001 (“Freund Provisional”)
1013	U.S. Patent No. 7,533,407 issued May 12, 2009 to Lewis <i>et al.</i> (“Lewis”)
1014	Google’s Opening Claim Construction Brief filed on August 26, 2025 in <i>K.Mizra LLC v. Google LLC</i> , Civ. Action No. 1:25-cv-00236 (W.D. Tex.)

Petition for *Inter Partes* Review of U.S. Patent No. 8,234,705

Exhibit No.	Description
1015	K.Mizra’s Responsive Claim Construction Brief filed on September 16, 2025 in <i>K.Mizra LLC v. Google LLC</i> , Civ. Action No. 1:25-cv-00236 (W.D. Tex.)
1016	Final Written Decision, <i>Cisco Systems, Inc. et al. v. K.Mizra LLC</i> , IPR2021-00593, Paper 41 (PTAB Sep. 19, 2022)
1017	Federal Circuit Decision vacating Final Written Decision issued in IPR2021-00593 and remanding to PTAB, <i>Cisco Systems, Inc. et al. v. K.Mizra LLC</i> , 2022-2290, 2023-1183 (Fed. Cir. Aug. 16, 2024)
1018	Order Terminating Due to Settlement After Institution of Trial, <i>Cisco Systems, Inc. et al. v. K.Mizra LLC</i> , IPR2021-00593, Paper 51 (PTAB Jan. 30, 2025)
1019	Claim Construction Order dated October 7, 2021, <i>K.Mizra LLC v. Cisco Systems, Inc.</i> , Civ. Action No. 6:20-cv-01031 (W.D. Tex.)
1020	U.S. Patent No. 5,987,611, issued November 16, 1999 to G. Freund
1021	U.S. Patent No. 6,782,527, issued August 24, 2004 to V. Kouznetsov <i>et al</i>
1022	Not assigned
1023	Not assigned
1024	Claim Construction Order filed November 21, 2023 in <i>K.Mizra LLC v. Hewlett Packard Enterprise Company et al.</i> , Civ. Action No. 2:21-cv-00305 (E.D. Tex.)

CERTIFICATE OF COMPLIANCE WITH 37 C.F.R. §42.24

The undersigned hereby certifies that the foregoing Petition totals 13,993 words, excluding the parts which are exempted by 37 C.F.R. §42.24(a)(1).

Date: September 19, 2025

/Patrick C. Keane/
Patrick C. Keane, Esq.
Registration No. 32,858
BUCHANAN INGERSOLL & ROONEY PC
1737 King Street, Suite 500
Alexandria, Virginia 22314
Direct Telephone (703) 838-6522
Main Facsimile (703) 836-2021
patrick.keane@bipc.com
Counsel for Petitioner

CERTIFICATE OF SERVICE

The undersigned hereby certifies that on this 19th day of September, 2025, a true and correct copy of the foregoing **PETITION FOR *INTER PARTES* REVIEW FOR U.S. PATENT NO. 8,234,705 and EXHIBITS 1001, 1002, 1004-1021, and 1024 and Petitioner's Power of Attorney** are being served upon the Patent Owner at the following correspondence address of record via FedEx:

KINNEY & LANGE, P.A.
333 S. 7TH ST., SUITE 2700
MINNEAPOLIS, MN 55402-2438

and courtesy copies are being sent to Patent Owner's litigation counsel via electronic mail as follows:

Bart A. Starr (bstarr@sheridanross.com)
Brian S. Boerman (bboerman@sheridanross.com)
Robert R. Brunelli (rbrunelli@sheridanross.com)
Claire Abernathy Henry (claire@millerfairhenry.com)
Michael Charles Smith (michael.smith@solidcounsel.com)

Date: September 19, 2025

/Patrick C. Keane/
Patrick C. Keane, Esq.
Registration No. 32,858
BUCHANAN INGERSOLL & ROONEY PC
1737 King Street, Suite 500
Alexandria, Virginia 22314
Direct Telephone (703) 838-6522
Main Facsimile (703) 836-2021
patrick.keane@bipc.com
Counsel for Petitioner