

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

FORCEPOINT LLC,
Petitioner

v.

WEBROOT, INC. AND OPEN TEXT, INC.,
Patent Owners.

IPR2023-00786
U.S. Patent No. 10,025,928

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,025,928 UNDER 37 C.F.R. § 42.101**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	RELIEF REQUESTED.....	1
A.	37 C.F.R. § 42.104(b)(1): Claims for Which Review is Requested	1
B.	37 C.F.R. § 42.104(b)(2): Statutory Grounds for Challenge	2
III.	THE '928 PATENT	3
A.	The '928 Patent Specification	3
B.	The Prosecution History	6
IV.	LEVEL OF ORDINARY SKILL.....	8
V.	CLAIM CONSTRUCTION UNDER 37 C.F.R. § 42.104(B)(3)	8
VI.	THE BOARD SHOULD NOT EXERCISE DISCRETION TO DENY INSTITUTION	8
A.	The Board Need Not Reach the <i>Fintiv</i> Factors Due to Compelling Evidence of Unpatentability.....	8
B.	The <i>Fintiv</i> Factors Favor Institution	9
C.	The <i>Becton Dickinson</i> Factors Favor Institution.....	11
VII.	CLAIMS 1-20 ARE UNPATENTABLE UNDER 37 C.F.R. § 42.104(B)(4)	12
A.	GROUND 1: DEVARAJAN RENDERS OBVIOUS CLAIMS 1-4, 6-7, 9-12, 14, and 16-20.....	13
1.	Claim 1	13
2.	Claim 2.....	37
3.	Claim 3	40

4.	Claim 4.....	41
5.	Claim 6.....	42
6.	Claim 7.....	44
7.	Claim 9.....	45
8.	Claim 10.....	47
9.	Claim 11.....	48
10.	Claim 12.....	48
11.	Claim 14.....	48
12.	Claim 16.....	48
13.	Claim 17.....	49
14.	Claim 18.....	50
15.	Claim 19.....	52
16.	Claim 20.....	52
B.	GROUND 2: DEVARAJAN IN COMBINATION WITH HUBBARD RENDERS OBVIOUS CLAIMS 1-4, 6-7, 9-12, 14, and 16-20.....	52
1.	Claim 1.....	52
2.	Claims 2-4 and 6-7.....	57
3.	Claims 9-12, 14 and 16-20.....	58
C.	GROUND 3: DEVARAJAN IN COMBINATION WITH MCAFEE RENDERS OBVIOUS CLAIMS 1-20.....	58
1.	Claim 1.....	58
2.	Claim 2.....	63
3.	Claim 3.....	64

4.	Claim 4.....	65
5.	Claim 5.....	66
6.	Claim 6.....	68
7.	Claim 7.....	68
8.	Claim 8.....	70
9.	Claim 9.....	72
10.	Claim 10.....	74
11.	Claim 11.....	74
12.	Claim 12.....	74
13.	Claim 13.....	75
14.	Claim 14.....	75
15.	Claim 15.....	75
16.	Claim 16.....	75
17.	Claim 17.....	75
18.	Claim 18.....	76
19.	Claim 19.....	76
20.	Claim 20.....	76

VIII. CERTIFICATION OF GROUNDS FOR STANDING UNDER 37	
C.F.R. §42.104	76

IX. MANDATORY NOTICES	77
A. Real Party-in-Interest Under 37 C.F.R. § 42.8(b)(1)	77
B. Related Matters Under 37 C.F.R. § 42.8(b)(2)	77
C. Lead and Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3)	78

D.	Service Information Under 37 C.F.R. § 42.8(b)(4).....	78
X.	PAYMENT OF FEES UNDER 37 C.F.R. §42.103.....	78
XI.	CONCLUSION	79

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte GmbH,</i> IPR2019-01469, Paper 6 (P.T.A.B. Feb. 13, 2020).....	11
<i>Apple Inc. v. Fintiv, Inc.,</i> IPR2020-00019, Paper 11 (P.T.A.B. Mar. 20, 2020).....	9
<i>Becton, Dickinson & Co. v. B. Braun Melsungen AG,</i> IPR2017-01586, Paper 8 (P.T.A.B. Dec. 15, 2017)	11
<i>Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.,</i> 868 F. 3d 1013 (Fed. Cir. 2017)	8
<i>Phillips v. AWH Corp.,</i> 415 F.3d 1303 (Fed. Cir. 2005) (en banc)	8
<i>Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.,</i> 200 F.3d 795 (Fed. Cir. 1999)	8
<i>Webroot Inc., et al. v. AO Kaspersky Lab, et al.,</i> No. 6:22-cv-243 (W.D. Tex.)	9, 77
<i>Webroot, Inc. et al v. Forcepoint LLC,</i> No. 6:22-cv-342 (W.D. Tex.)	9, 77
Statutes	
35 U.S.C. § 102	2
35 U.S.C. § 103	2
35 U.S.C. § 311	1
35 U.S.C. § 314	9

Other Authorities

37 C.F.R. § 1	12
37 C.F.R. § 42	1, 2, 8, 12, 77, 78

TABLE OF EXHIBITS

Exhibit	Description
1001	U.S. Patent No. 10,025,928 to Jaroch et al.
1002	Declaration of Dr. Michael T. Goodrich
1003	Patent File History of U.S. Patent No. 10,025,928
1004	U.S. Patent Publication No. 2013/0007870 (“Devarajan”)
1005	U.S. Patent Publication No. 2009/0070873 (“McAfee”)
1006	Complaint, <i>Webroot, Inc. et al v. Forcepoint LLC</i> , No. 6:22-cv-342 (W.D. Tex.) (Mar. 31, 2022)
1007	Excerpt from Transcript of Motions Hearing in <i>Webroot, Inc., et al. v. AO Kaspersky Lab</i> , No. 6:22-cv-243 (W.D. Tex.) (Dec. 9, 2022)
1008	U.S. Patent No. 9,654,495 (“Hubbard”)
1009	Curriculum Vitae of Dr. Michael T. Goodrich
1010	LexMachina Summary of Cases filed by Webroot, Inc. in the Western District of Texas

I. INTRODUCTION

Petitioner Forcepoint LLC (“Forcepoint” or “Petitioner”) requests *inter partes* review of claims 1-20 of U.S. Patent No. 10,025,928 (Ex.1001, the “’928 Patent”), purportedly assigned to Patent Owners Webroot, Inc. and Open Text, Inc. (collectively, “Patent Owners”).

The ’928 Patent describes “[a] protection module” that “operates to analyze threats, at the protocol level...by intercepting all requests that a browser engine resident in a computing device sends and receives.” Ex.1001, Abstract. The protection module “completes the requests without the help of the browser engine,” and then “analyzes and/or modifies the completed data before the browser engine has access to it.” *Id.* This analysis involves “processing, removing, and/or adding any code as needed.” *Id.* Then, “the protection module provides the HTML content to the browser engine, and the browser engine receives responses from the protection agent as if it was speaking to an actual web server.” *Id.* However, this system of intercepting and modifying browser requests to remove threats long predates the ’928 Patent’s priority date, and the prior art demonstrates this was well-known. Therefore, the challenged claims are invalid.

II. RELIEF REQUESTED

A. 37 C.F.R. § 42.104(b)(1): Claims for Which Review is Requested

Petitioner respectfully requests *inter partes* review under 35 U.S.C. § 311 of claims 1-20 of the ’928 Patent and the cancellation of those claims as unpatentable.

B. 37 C.F.R. § 42.104(b)(2): Statutory Grounds for Challenge

The '928 Patent was filed October 2, 2012, and lists Provisional Application No. 61/542,693, filed on October 3, 2011, as a related application. Ex.1001. Even assuming the '928 Patent is entitled to a priority date of October 3, 2011,¹ each asserted reference identified in the table is prior art to the '928 Patent under at least pre-AIA 35 U.S.C. § 102(a), (b), and/or (e).

Prior Art References
Devarajan (Ex.1004), U.S. Patent Publication No. 2013/0007870, filed September 29, 2011, and published January 3, 2013. (§ 102(a), (e)).
McAfee (Ex.1005), U.S. Patent Publication No. 2009/0070873, filed September 11, 2007, and published March 12, 2009. (§ 102(a), (b), (e)).
Hubbard (Ex.1008), U.S. Patent No. 9,654,495, filed February 28, 2007, published June 5, 2008, and issued May 16, 2017. (§ 102(a), (b), (e)).

Claims 1-20 of the '928 Patent are unpatentable under pre-AIA 35 U.S.C. § 103 based on the following grounds:

No.	Grounds of Unpatentability	Basis
1	Devarajan renders obvious claims 1-4, 6-7, 9-12, 14, and 16-20	§ 103
2	Devarajan in view of Hubbard renders obvious claims 1-4, 6-7, 9-12, 14, and 16-20.	§ 103
3	Devarajan in view of McAfee renders obvious claims 1-20.	§ 103

¹ In the parallel district court proceedings, Plaintiffs asserted a priority date of October 3, 2011 for the '928 Patent. Ex.1006, 22.

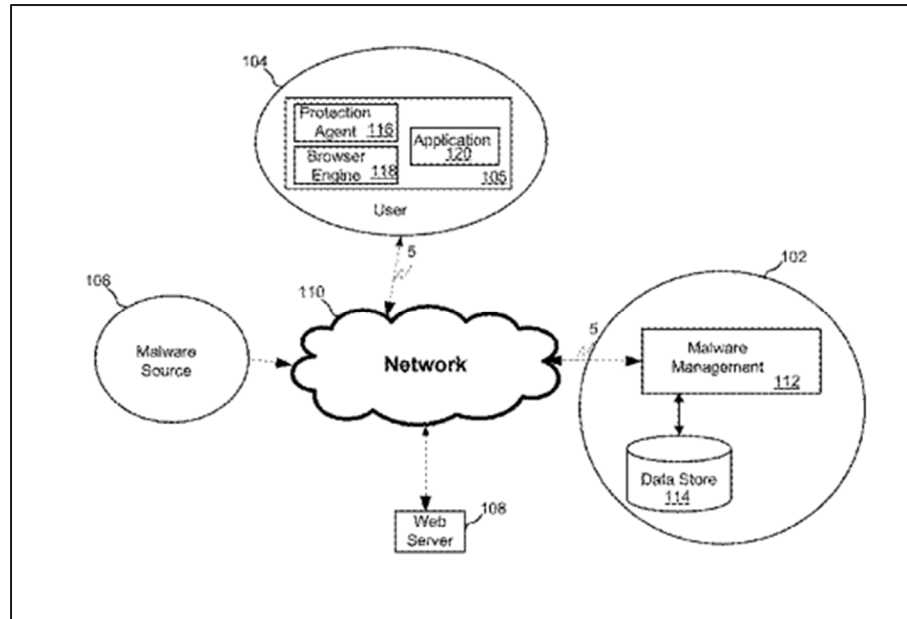
III. THE '928 PATENT

The '928 Patent, titled “PROACTIVE BROWSER CONTENT ANALYSIS” issued on July 17, 2018, from U.S. Patent Application No. 13/633,956, filed on October 3, 2012. The '928 Patent does not claim priority to any other patent or application.

A. The '928 Patent Specification

The '928 Patent includes independent claims 1 and 9, which relate to content analysis and malware prevention. The '928 Patent includes a protection agent that analyzes web content at a protocol level to detect and remove potential malware before the content is served to the user. Ex.1001, Abstract.

Figure 1 depicts “an environment in which several embodiments of the invention may be implemented.” *Id.*, 2:67-3:4. The environment includes “a security center 102 (also referred to as a “central” or “base” computer), remote user 104 operating a remote computer 105, a malware source 106, and a web server 108 [that] are all communicatively coupled through one or more networks (e.g., the Internet and/or local or wide area networks) 110 and links 5.” *Id.*, 3:4-9.

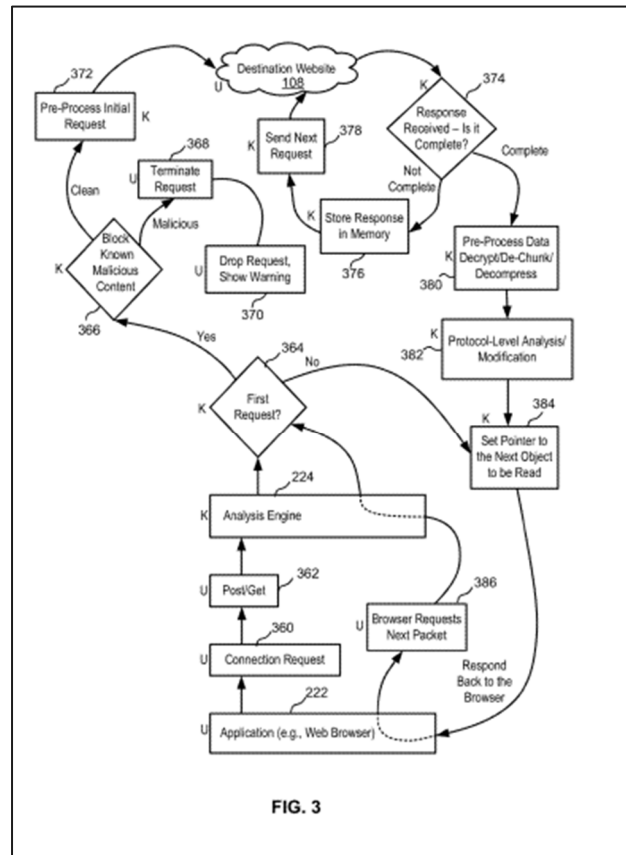


Ex.1001, Fig.1.

The protection agent, which is part of a remote computer, “operates to analyze threats, at the protocol level (e.g., at the HTML level), by intercepting all requests that the browser engine 118 sends and receives, and the protection agent 116 completes the requests without the help of the browser engine 118.” *Id.*, 4:6-11. Once complete, “the protection agent 116 analyzes and/or modifies the completed data before the browser engine 118 has access to it,” (*id.*, 4:11-13), and “if necessary, decrypts and decompresses the web content and then assembles the requested web page.” *Id.*, 4:56-61. Then, “the protection agent feeds the HTML content back to the browser engine 118, and the browser engine 118 receives responses from the protection agent 116 as if it was ‘speaking’ to an actual web server (e.g., web server

108) when in fact, it is speaking to an analysis engine of the protection agent 116.”
Id., 4:13-21.

Figure 3 “depicts exemplary aspects of the process flow that occurs in connection with” the operation of the remote computer. *Id.*, 5:25-29.



Ex.1001, Fig.3.

When a user initiates a request via the browser/application, “a connection request 360 is initiated as a POST/GET request 362 to a website (e.g., hosted by the webserver 108) and the analysis engine 224 looks at the context of the request to assess whether the request is the first request in a session (Block 364), and if the request is the first request, a determination is made as to whether the request is

associated with known, malicious content (Block 366), and if so, the request is blocked (Blocks 368, 370).” *Id.*, 5:52-62. If the request is not blocked, then “the content acquisition component 226 iteratively sends requests and receives content (Blocks 374, 376, 378) from the website 108 until the requested content has been completely received.” *Id.*, 6:17-23. After the content is received, “the browser page may be analyzed by the protocol-level analysis/modification component 382 depicted in FIG. 3 outside of the context of the browser engine 218 before the browser engine 218 has operated on the webpage.” *Id.*, 6:42-51. If malicious script is found during this analysis, “it may be commented out before the content is handed to the browser engine.” *Id.*, 6:56-58.

B. The Prosecution History

The ’928 Patent’s initial application had two independent claims that claimed general elements of receiving, analyzing, modifying, and rendering web content. Ex.1003, ’928 File History, 23-24. In the first substantive office action on July 18, 2013, the Examiner issued a non-final rejection rejecting multiple claims as indefinite, and further rejecting claims 1-20 under §102(b) as anticipated by Scott et al. (Scott), “Abstracting Application-Level Web Security.” *Id.*, 59-65. The Applicant responded on November 18, 2013, amending the claims to include “the protection agent” and a “security center” (*id.*, 80-87), and adding two new claims identifying that the malware threat can be a URL. *Id.*, 84.

The Examiner proceeded to issue repeated rejections to the claims as being obvious over Scott in view of Hallak (U.S. Patent No. 8,464,318). *Id.*, 101-05. The Examiner found that while “Scott does not explicitly state that the browser application is ‘on the remote computer,’ i.e. that the browser and the gateway (i.e. ‘protection agent’) are on the same computer,” Hallak makes up for this deficiency. *Id.*, 102-03. The Applicant responded by making a series of amendments to claims 1 and 11, including “intercepting, by a protection agent, a request from a web browser,” “determining, by a protection agent, whether the request is associated with known malicious content,” and “when the request is associated with known malicious content, blocking, by a protection agent, the request” (*id.*, 138-40), “modifying the web content comprises modifying the web content at protocol level to remove the malware threat from the web content” (*id.*, 256-58), the “modified version of the web content does not comprise the malware threat” (*id.*, 294-96), and the “web content comprises data for assembling a web page” (*id.*, 322-24).

The Examiner nonetheless issued a final rejection on October 6, 2017, again in view of Scott and Hallak. The Applicant responded by further amending the independent claims to require that “the modified web page is assembled using the modified web content.” *Id.*, 360-62. The ’928 Patent finally issued following this response without further comment.

IV. LEVEL OF ORDINARY SKILL

A person having ordinary skill in the art (a “POSITA”) relative to the ’928 Patent would have at least a four-year undergraduate degree in computer science or a closely related field and at least two years of experience in the field of computer and network security. Ex.1002, Declaration by Michael Goodrich, Ph.D., ¶¶26-29. Additional education could substitute for professional experience and vice versa. *Id.*

V. CLAIM CONSTRUCTION UNDER 37 C.F.R. § 42.104(B)(3)

Claims only need to be construed “to the extent necessary to resolve the controversy.” *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (citing *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)). Here, no terms need construction because the claims read on the prior art under any construction consistent with *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc).

VI. THE BOARD SHOULD NOT EXERCISE DISCRETION TO DENY INSTITUTION

A. The Board Need Not Reach the *Fintiv* Factors Due to Compelling Evidence of Unpatentability

The USPTO Director stated that the Board “will not rely on the *Fintiv* factors to discretionarily deny institution ... where a petition presents compelling evidence of unpatentability,” which “would plainly lead to a conclusion that one or more claims are unpatentable by a preponderance of the evidence.” USPTO Memorandum

(June

21,

2022),

https://www.uspto.gov/sites/default/files/documents/interim_proc_discretionary_denials_aia_parallel_district_court_litigation_memo_20220621_.pdf (“*Fintiv* Memorandum”), 2, 4. This Petition meets that criteria.

B. The *Fintiv* Factors Favor Institution

To the extent the Board finds it necessary to analyze the *Fintiv* factors, the relevant *Fintiv* factors demonstrate the Board should not exercise discretion to deny institution under 35 U.S.C. § 314(a) and *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, 2020 WL 2126495 (P.T.A.B. Mar. 20, 2020) (precedential).

Factor 1 (existence or likelihood of a stay):

The '928 Patent was asserted against Petitioner in *Webroot, Inc. et al v. Forcepoint LLC*, No. 6:22-cv-342 (W.D. Tex.) (filed Mar. 31, 2022) (the “related litigation”).² Ex.1006, ¶¶14-16. Where a stay on the basis of parallel IPR proceedings has been neither requested nor granted, “[t]his factor does not weigh for or against discretionary denial.” *Fintiv*, at *12. As a stay was neither requested nor granted in the related litigation, Factor 1 is neutral.

Factor 2 (proximity of trial date to statutory IPR deadline):

Pursuant to the District Court’s amended scheduling order on January 22, 2023 (Dkt. 160), the trial in this case will not begin until August 19, 2024. Based

² This case has been consolidated with four others as *Webroot Inc. and Open Text, Inc. v. AO Kaspersky Lab*, 6:22-cv-00243 (W.D. Tex.).

on an 18-month IPR schedule, a Final Written Decision (“FWD”) arising from this petition would issue around October 2024. However, “scheduled trial dates are unreliable and often change.” *Fintiv* Memorandum, 8. “A court’s scheduled trial date, therefore, is not by itself a good indicator of whether the district court trial will occur before the statutory deadline for a [FWD].” *Id.* This is particularly true here, as the consolidated case consists of multiple patents asserted against multiple defendants (fifteen patents and five defendants). The District Court stated this case is “about as complicated as you’re going to get with patent cases,” Ex.1007, 53:7-8, and said it would be “amenable to adjusting the schedule” by “granting more time to address things,” *id.*, 53:22-23. The case against Forcepoint was filed last of the five cases, so any trial relating to this patent would likely occur after other defendants. Ex.1010, Summary of Webroot WDTX Cases. Thus, it is unlikely the District Court will resolve the dispute related to the ’928 Patent before a FWD arising from this petition. Thus, this factor weighs against discretionary denial.

Factor 3 (investment in parallel proceeding):

The District Court proceeding is still in the early stages, as fact discovery has not yet begun. Petitioner has committed significant resources to this IPR proceeding, which balances any resources invested in the related litigation, which would be lost without consideration on the merits. In addition, institution of this Petition would reduce the substantial expenses that would result from the related litigation. Given

Petitioners' diligence and significant investment in filing this Petition, especially this early in the litigation, this factor weighs against a discretionary denial.

Factor 4 (overlap between issues):

There should be no overlap of invalidity challenges. An FWD is likely to issue before the Texas trial (*see* Factor 2) and Petitioners will be estopped from arguing grounds raised or that reasonably could have been raised in this IPR. Thus, Factor 4 weighs against discretionary denial.

Factor 5 (same parties):

Petitioner is a defendant in the related litigation. But in view of the other *Fintiv* factors—which heavily weigh against the Board's exercise of §314(a) discretion—the similarity of the parties is of limited weight.

Factor 6 (other circumstances impacting the Board's exercise of discretion):

Factor 6 favors institution because this Petition presents compelling evidence of unpatentability. Accordingly, in light of the guidance set forth in the *Fintiv* Memorandum at 2-4, this IPR should be instituted.

C. The *Becton Dickinson* Factors Favor Institution

The factors in *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, 2017 WL 6405100, Paper 8 at *17-18 (P.T.A.B. Dec. 15, 2017) (precedential as to § III.C.5, first para) strongly favor institution. *See also Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte GmbH*, IPR2019-01469, 2020 WL 740292,

Paper 6 (P.T.A.B. Feb. 13, 2020) (precedential). Devarajan, McAfee, and Hubbard were not cited or considered during prosecution of the '928 Patent. The prosecution of the '928 Patent involved rejections over prior art references Scott and Halak. The Examiner found that Scott and Halak, in combination, taught a protection agent operating on a remote computer, but did not teach a system where “the modified web page is assembled using the modified web content.” Ex.1003, 102-03, 360-62. The grounds presented in this Petition relying on Devarajan, McAfee, and/or Hubbard are different from the prior art discussed during prosecution and clearly disclose this claim element for the reasons described below. Accordingly, the grounds presented in this Petition are not cumulative to anything discussed during prosecution of the '928 Patent. *See* Ex.1003.

VII. CLAIMS 1-20 ARE UNPATENTABLE UNDER 37 C.F.R. § 42.104(B)(4)

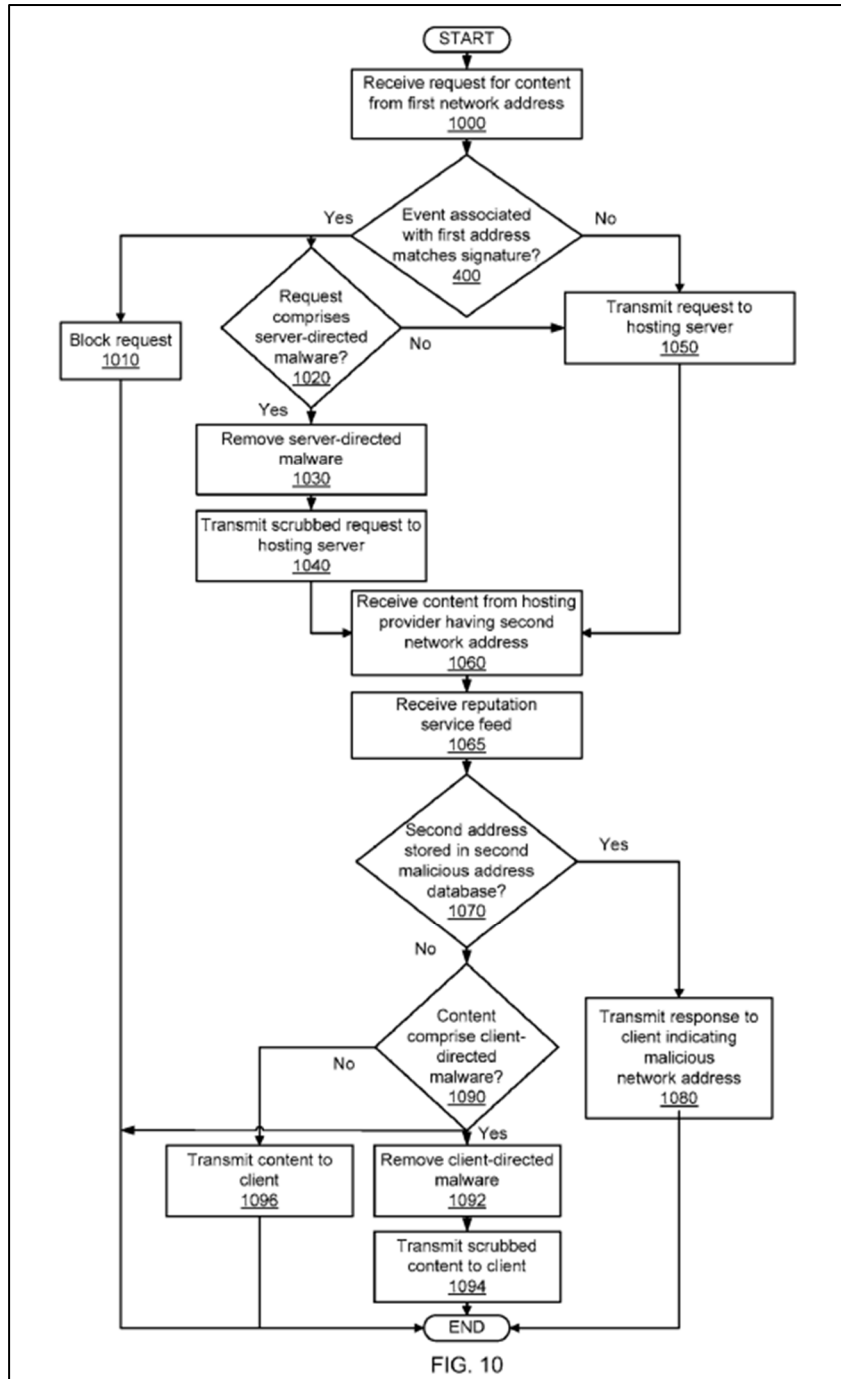
Under 37 C.F.R. § 42.104(b)(5) for evidence supporting challenge, a list of exhibits is provided at the beginning of the Petition. The relevance of this evidence and the specific portions supporting the challenge are provided herein in Section VII. Petitioners submit a declaration of Dr. Michael Goodrich (Ex.1002) in support of this Petition under 37 C.F.R. § 1.68.

A. GROUND 1: DEVARAJAN RENDERS OBVIOUS CLAIMS 1-4, 6-7, 9-12, 14, and 16-20

1. Claim 1

- a. [1P] A content analysis and malware prevention method comprising:

To the extent the preamble is limiting, Devarajan discloses “*a content analysis and malware prevention method.*” Devarajan teaches a system for “bi-directional network traffic malware detection and removal.” Ex.1004, Abstract. Figure 10 depicts its claimed method for analyzing content and preventing malware. Ex.1002, ¶53.

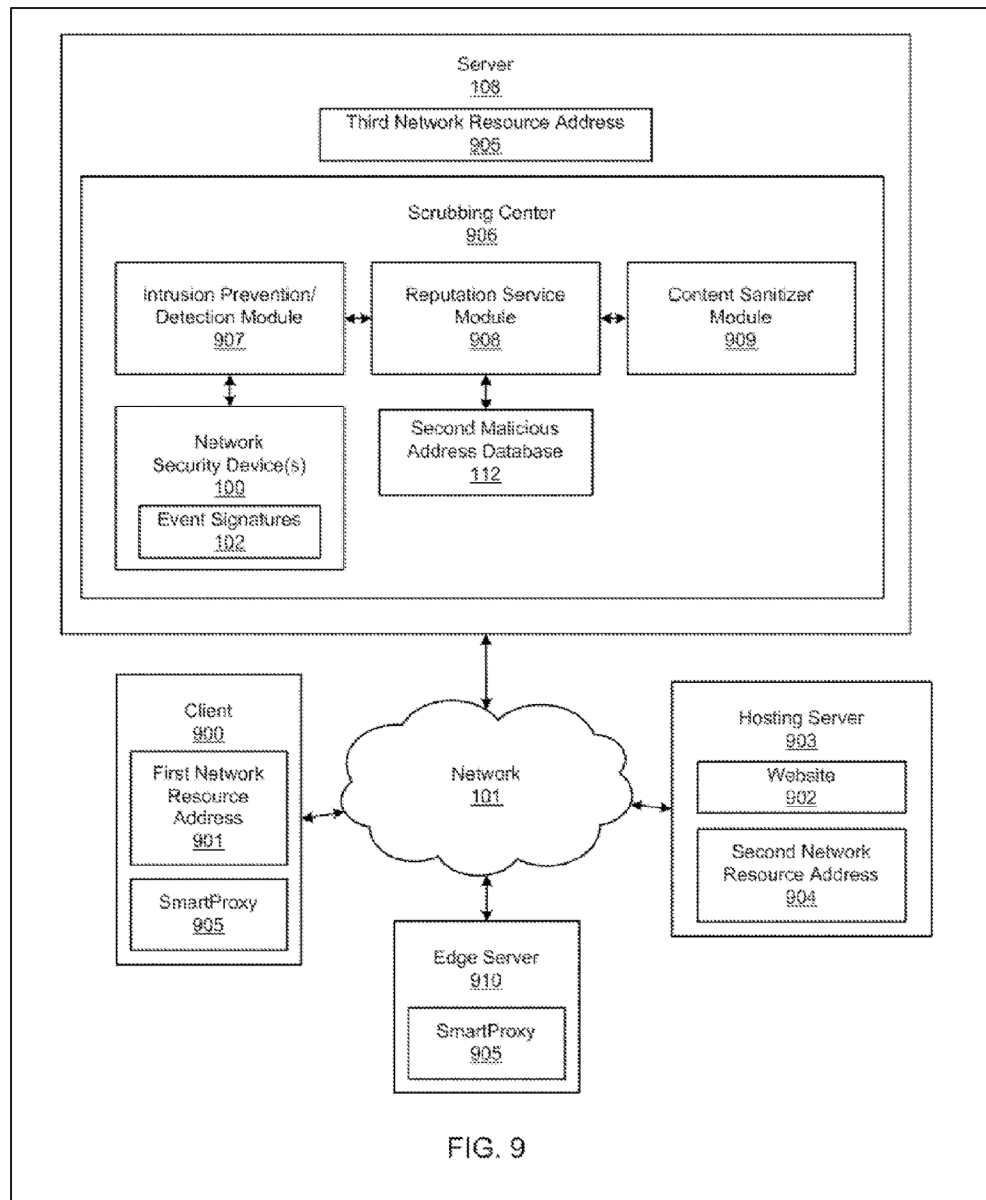


Ex.1004, Fig.10.

- b. [1A] intercepting, by a protection agent, a request from a web browser;

Devarajan discloses or renders obvious “*intercepting, by a protection agent, a request from a web browser.*”

The '928 Patent describes a protection agent as a module that analyzes threats at a protocol level and modifies data received from web servers before the user's browser engine has access to it. Ex.1001, 1:43-57. A POSITA would understand the server 108 in Devarajan teaches “*a protection agent.*” Ex.1004, [0053], [0059], [0095]; Ex.1002, ¶60. Devarajan discloses an exemplary system architecture in Figure 9:



Ex.1004, Fig.9.

Devarajan explains that “Fig. 9 illustrates an embodiment of a bi-directional network traffic malware detection and removal system that may comprise one or more server 108 (having a third network resource address 905) communicatively coupled to a network 101.” *Id.*, [0088]; Ex.1002, ¶55. Server 108 also includes a scrubbing center 906 in Figure 9 that “may comprise a plurality of software modules

running on the one or more server 108, and may comprise an intrusion prevention and detection module 907, a reputation service module 908, and/or a content sanitizer module 909.” Ex.1004, [0095]; Ex.1002, ¶55. Website content received by Devarajan’s system is “redirected to the scrubbing center 906 to ensure that...any client-directed malware is removed.” Ex.1004, [0107]; Ex.1002, ¶59.

Devarajan’s “*protection agent*” can be implemented either on a remote computer or within the client itself. Ex.1004, [0053]; Ex.1002, ¶¶58-59. Server 108 **“could be any computer or program** that provides services to other computers, programs, or users **either in the same computer or over a computer network 101.”** Ex.1004, [0053]³; *see also id.* (“the one or more server 108 could be...**software**”); Ex.1002, ¶58. A POSITA would thus understand that server 108 could be software executing, *e.g.*, on a desktop computer, such as a client 900, and providing services to other software programs running on that computer. Ex.1002, ¶58; *see also* Ex.1004, [0092] (disclosing that client 900 may be, *e.g.*, a desktop or laptop).

A POSITA would find it obvious to implement server 108 as a software server executing on the same computer (client 900) as the web browser. Ex.1002, ¶59. For example, a POSITA would understand this reduces the communication time between

³ All emphasis added unless otherwise noted.

a client and the server 108 because such communications would occur locally rather than going over a network, thereby improving latency for the overall system in Devarajan. *Id.* Also, a POSITA would understand that placing the server 108 as a software server on the client 900, as described, *e.g.*, in Ex.1004, [0053], would reduce the amount of hardware needed to implement the system of Devarajan, thereby reducing costs. Ex.1002, ¶59.

Moreover, Devarajan discloses a SmartProxy 905 that “may act as an intermediary between the client 900 or edge server 910 and the hosting server 903,” and shows that the SmartProxy 905 is a component of client 900. Ex.1004, [0105], Figs.9-10; Ex.1002, ¶59. A POSITA would understand Devarajan’s description of the functionality of the SmartProxy 905 to include communicating directly with the scrubbing center 906 (disclosed as a software module in server 108). Ex.1002, ¶59. Devarajan discloses that the **“SmartProxy 905 may comprise a proxy server application, software, or script that...may communicate directly to the scrubbing center 906.”** Ex.1004, [0105]; Ex.1002, ¶59. Devarajan further teaches that the “SmartProxy 905 may act as an intermediary between the client 900 or edge server 910 and the hosting server 903.” Ex.1004, [0105]; Ex.1002, ¶59. Devarajan’s SmartProxy “ensures all traffic from and/or to the client 900 is routed to, and filtered through, the scrubbing center 906.” Ex.1004, [0105]; Ex.1002, ¶59.

Accordingly, Devarajan discloses an embodiment where a SmartProxy 905 and server 108 are software executing on the client 900 computer (*e.g.*, a desktop or laptop) to provide the functionality of the “*protection agent*.” *See, e.g.*, Ex.1002, ¶¶59-60; Ex.1004, [0095]. A POSITA would understand this embodiment in Devarajan to disclose or render obvious “*a protection agent*,” as required by claim 1 of the ’928 Patent. Ex.1002, ¶¶54-60.

Devarajan also teaches that server 108—a “*protection agent*”—may receive “*a request from a web browser*.” For example, in Figure 9, Devarajan teaches server 108 (*e.g.*, the “*protection agent*”) is “configured to receive **a request for website content**.” Ex.1004, [0006]; *see also id.*, [0109], Fig.10 (“Fig. 10 illustrates an embodiment of a method for bi-directional detection and removal of network traffic malware”); Ex.1002, ¶55. The steps of Figure 10 include “receiving, from a client 900 having a first network resource address 901, **a request for content from a website** 902 hosted on a hosting server 903 having a second network resource address 904 and resolving from a URL such as domain name, wherein the URL (*e.g.*, domain name) is pointed in the DNS to a third network resource address 905 for one or more server 108 running a scrubbing center 906 (Step 1000).” Ex.1004, [0109]; Ex.1002, ¶57. Devarajan further explains that “the content request may be received (Step 1000)” and “the content request may be received via HTTP protocol, the request perhaps being initiated by the client’s 900 browser.” Ex.1004, [0110].

Thus, a POSITA would understand server 108 (“*protection agent*”) could receive a content “request from a browser.” Ex.1002, ¶¶61-65.

Devarajan teaches server 108 “*intercepting*” a request from a web browser. The ’928 Patent describes the protection agent “intercepting” a request from a web browser by receiving and/or stopping the request before it arrives at its intended destination. *See, e.g.*, Ex.1001, 4:6-11, 6:17-26; Ex.1002, ¶62. Likewise, in the context of an electronic communication, a POSITA would understand an interception, under its plain and ordinary meaning, to be interrupting an electronic transmission before it reaches the intended recipient. Ex.1002, ¶62.

Devarajan teaches that the server 108 receives electronic communications from a client’s web browser **before** they reach the intended recipient (*i.e.*, the hosting server 903). *Id.*, ¶63. For example, Devarajan explains server 108 may receive “a request for content from a **website 902**.” Ex.1004, [0089]; Ex.1002, ¶63. The SmartProxy 905, “may be configured (*e.g.*, programmed) to receive **website 902** content requests from the client 900 and **redirect such requests to the scrubbing center 906**.” Ex.1004, [0106]; Ex.1002, ¶59. “[T]he SmartProxy may accomplish this by storing the website 902 hosting server’s 903 second network address 904 (*e.g.*, IP address) in association with the scrubbing center 906 server’s 108 third network resource address (*e.g.*, IP address), along with instructions to route requests for the website 902 to the scrubbing center 906.” Ex.1004, [0105]-[0106]; [0101]-

[0104]; Ex.1002, ¶59. For example, Devarajan teaches that when “the client 900 initiates the content request, the returned website 902 content may **automatically be redirected to the scrubbing center 906.**” Ex.1004, [0105]-[0106]; [0101]-[0104]; Ex.1002, ¶59. The SmartProxy 905 taught by Devarajan “**may intercept and redirect the traffic to the scrubbing center 906,**” which is accomplished by “storing the scrubbing center 906 server’s 108 third network resource address (e.g., IP address), along with **instructions to route all incoming traffic to the scrubbing center 906 and request that the scrubbing center 906 return scrubbed content to the SmartProxy 905.**” Ex.1004, [0105]-[0106]; [0101]-[0104]; Ex.1002, ¶59.

Further, Devarajan teaches the content request intercepted by the server 108 may be initiated by a client 900’s web browser. Ex.1004, [0093]; Ex.1002, ¶61. A POSITA would understand Devarajan’s server 108 teaches “*a protection agent*” that “*intercept[s]...a request for content from a web browser,*” as required by claim 1. Ex.1002, ¶¶54-65.

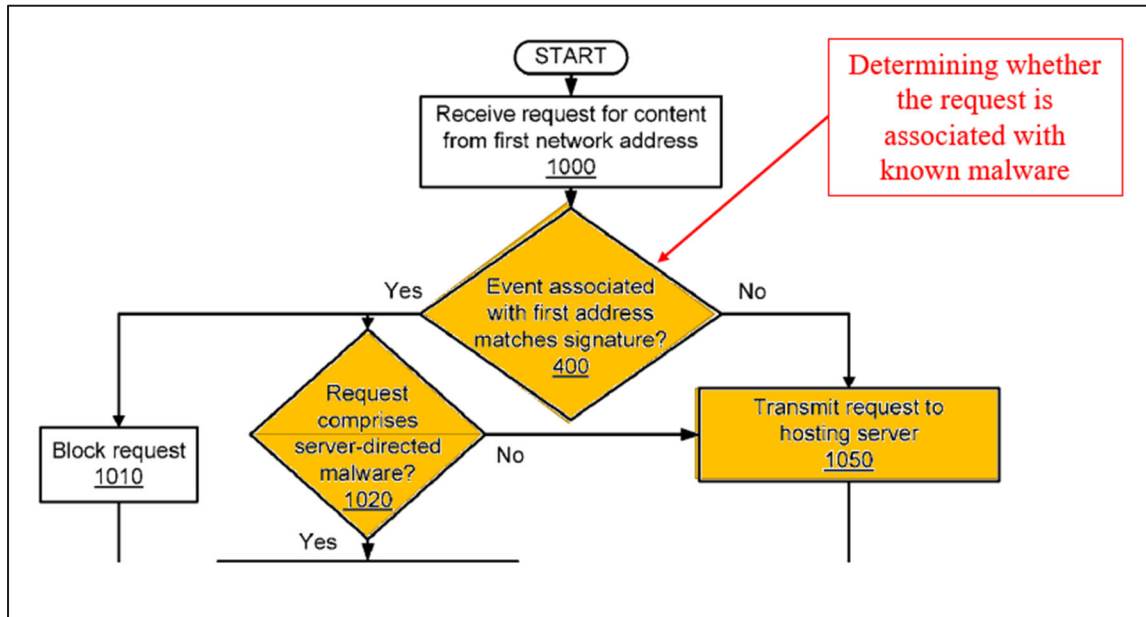
- c. [1B] determining, by the protection agent, whether the request is associated with known malicious content;

Devarajan discloses or renders obvious “*determining, by the protection agent, whether the request is associated with known malicious content.*”

Devarajan further discloses “*determining ... whether the request is associated with known malicious content.*” Ex.1002, ¶¶66-70. In Figure 10, the first step after receiving the request is step 400, which “determine[es] whether an event associated

with the client 900 or its first network resource address 901 matches one or more of a plurality of event signatures 102 in one or more network security device 100 communicatively coupled to the network 101.” Ex.1004, [0111]; Ex.1002, ¶67. Devarajan further describes that its “signature based network security device 100 may store, or otherwise have access to...a plurality of event signatures 102,” which “are simply known attack patterns.” Ex.1004, [0044]-[0045]; Ex.1002, ¶67. Devarajan explains these signatures “comprise particular strings that have been discovered in **known malicious activity**,” such as “a virus signature, a worm signature, a trojan horse signature, a rootkit signature, a backdoor signature, a spyware signature, a keystroke logger signature, or a phishing application signature.” Ex.1004, [0045]-[0046]; Ex.1002, ¶67. Further, Devarajan discloses “[a]n ‘event’ may comprise **any malicious or unwanted activity**, perhaps performed by or via a network resource 103 having a network resource address 104,” and “the network resource address may comprise an IP address, a URL, or a domain (e.g., domain name) for the network resource 103.” Ex.1004, [0048]-[0049]; Ex.1002, ¶67. Devarajan further explains that “event signatures 102” may be “based upon prior known attacks.” Ex.1002, ¶67, Ex.1004, [0044]-[0045]. Thus, a POSITA would understand after the “*protection agent*” (server 108) receives a request, step 400 determines whether the event matches the signature of known malicious

content—e.g., “determine[es] ...whether the request is associated with known malicious content.” Ex.1002, ¶¶66-69.



Ex.1004, Fig.10 (annotated).

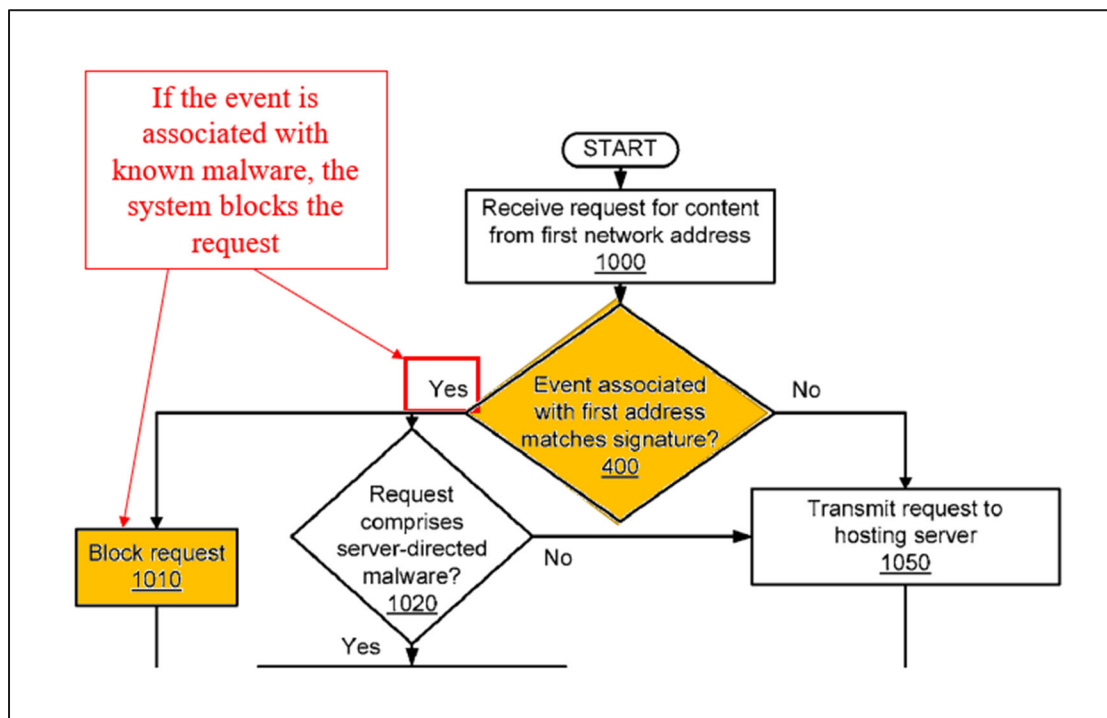
Devarajan discloses that the server 108 is “configured to” determine whether an “event matches one or more of the plurality of event signatures 102.” *Id.*, [0053]. Thus, a POSITA would understand the server 108 (“*protection agent*”), in an embodiment as a software server executing on client 900 with its scrubbing center 906, will determine whether the request from the client is associated with known malicious content. Ex.1002, ¶¶66-70.

- d. [1C] when the request is associated with known malicious content, blocking, by the protection agent, the request;

Devarajan discloses or renders obvious “*when the request is associated with known malicious content, blocking, by the protection agent, the request.*” As

described in Section VII.A.1.c, Devarajan teaches determining “*whether the request is associated with known malicious content.*”

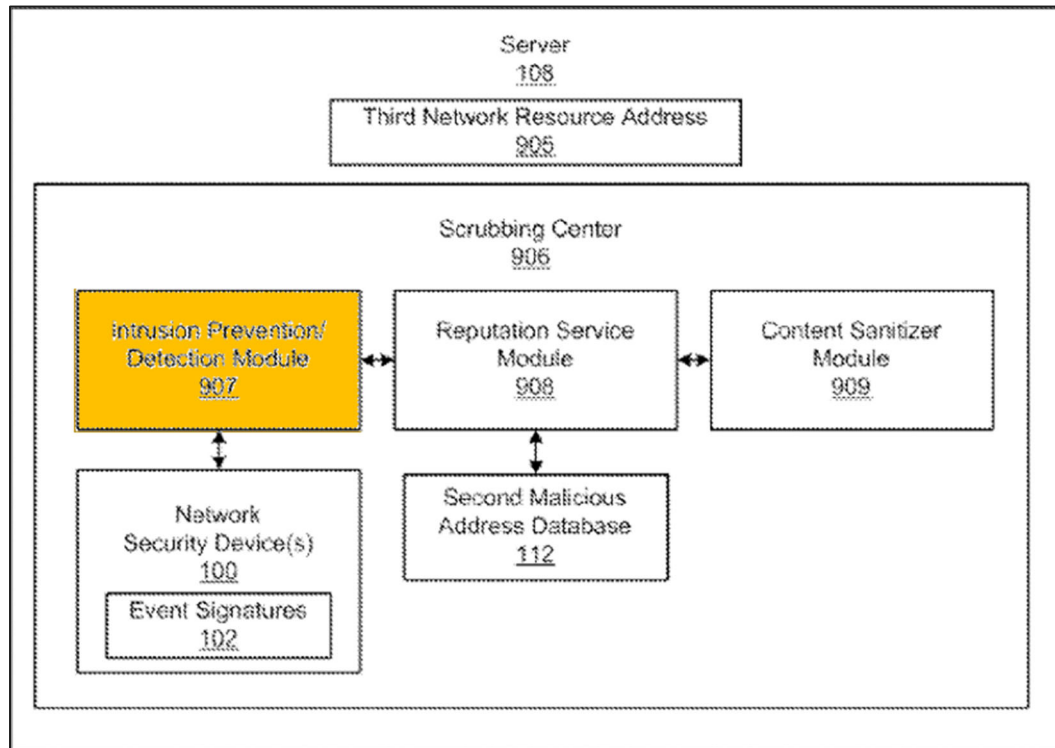
Devarajan teaches that “when” or “if the event matches one or more event signature 102, the method further **may comprise blocking the request** for content from reaching the hosting server 903 (Step 1010),” as depicted in Figure 10. Ex.1004, [0112]; Ex.1002, ¶71.



Ex.1004, Fig.10 (annotated)

Devarajan discloses that the server 108 (“*protection agent*”) will perform the blocking step 1010 (e.g., “*blocks the request*”) if the request matches known malware content. Ex.1002, ¶72. Furthermore, server 108 contains an intrusion prevention and detection module 907. See Ex.1004, Fig.9; Ex.1002, ¶72. “[T]he

intrusion prevention and detection module 907 may **block the request** for content from reaching the hosting server 903.” Ex.1004, [0098]; *see also id.*, [0101], [0112]. The “intrusion prevention and detection module” is part of the same server 108 performing step 1000 in Figure 10. Ex.1004, [0110]; Ex.1002, ¶71-73.

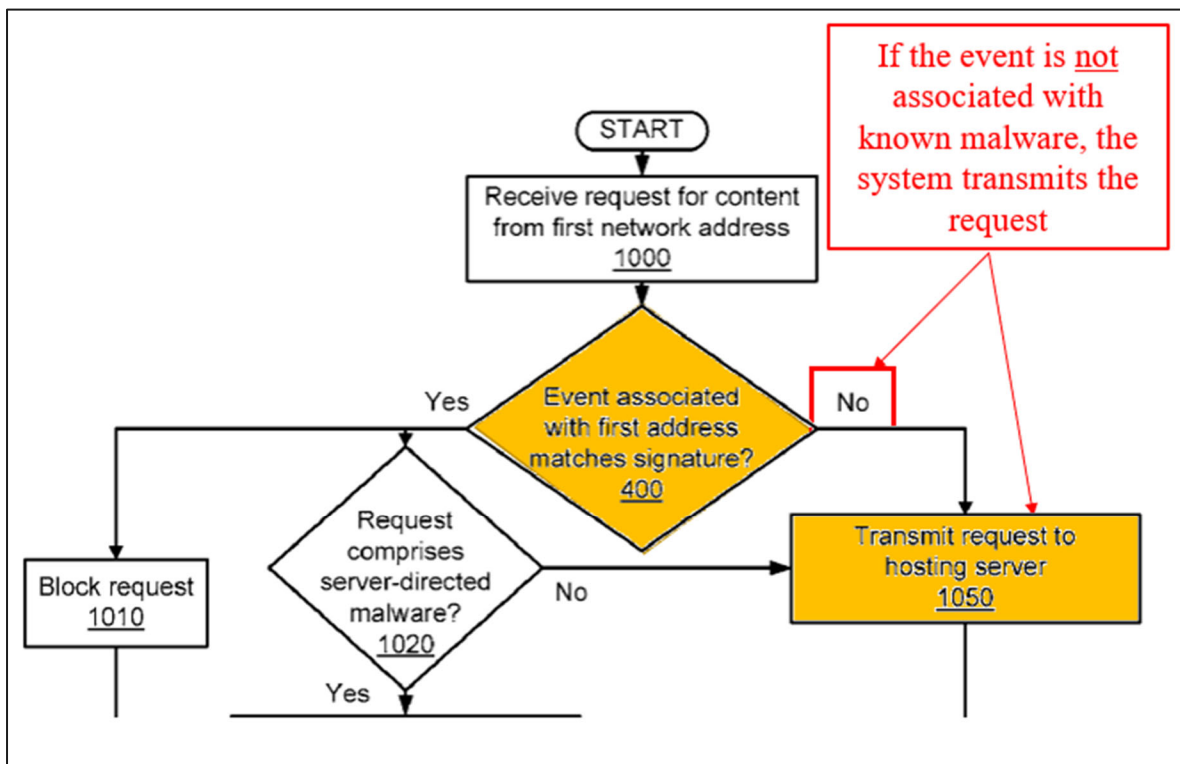


Ex.1004, Fig.9 (annotated).

- e. [1D] when the request is not associated with known malicious content: sending, by the protection agent, the request to one or more web servers

Devarajan teaches or renders obvious “*when the request is not associated with known malicious content: sending, by the protection agent, the request to one or more web servers.*” For example, as described in Section VII.A.1.b, “[t]he content request may be received” by a (software) server 108 (“*protection agent*”) as shown

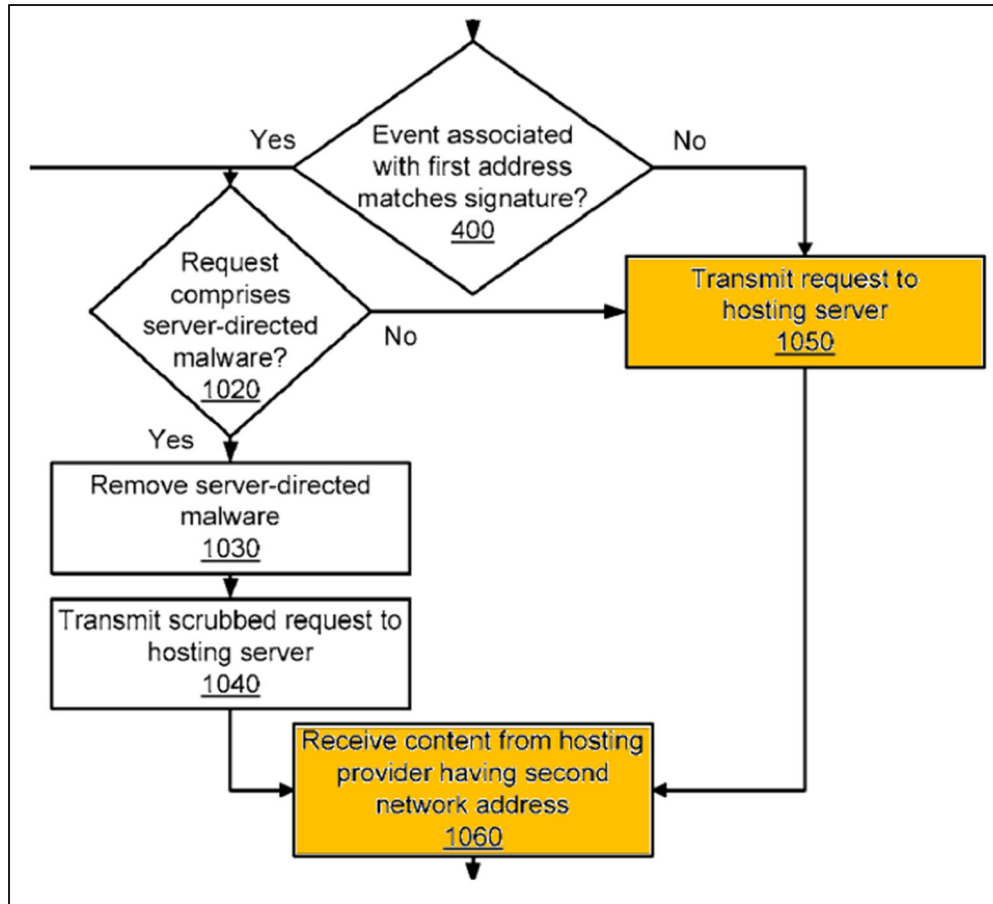
in block 1000 of Figure 10. *Id.*, [0110]. At the next block (400), the system determines whether the event matches one or more of a plurality of event signatures. *Id.*, [0111]; Ex.1002, ¶75. If at step 400 the system determines “the event does not match an event signature 102”—e.g., “*when the request is not associated with known malicious content*”—“the request for content may be transmitted ...**to the hosting server 903**” as depicted in step 1050. *Id.* The content is sent “responsive to receiving the request for content...**from the server 108**,” i.e., the “*protection agent*.” Ex.1004, [0089], [0115]; Ex.1002, ¶¶74-76. A POSITA would understand this to disclose “*sending, by the protection agent, the request to one or more web servers.*” Ex.1002, ¶¶74-75.



Ex.1004, Fig.10 (annotated).

- f. [1E] receiving, at the protection agent on a remote computer, web content from the one or more web servers, wherein web content comprises data for assembling a web page, and wherein the web content is received in response to the request;

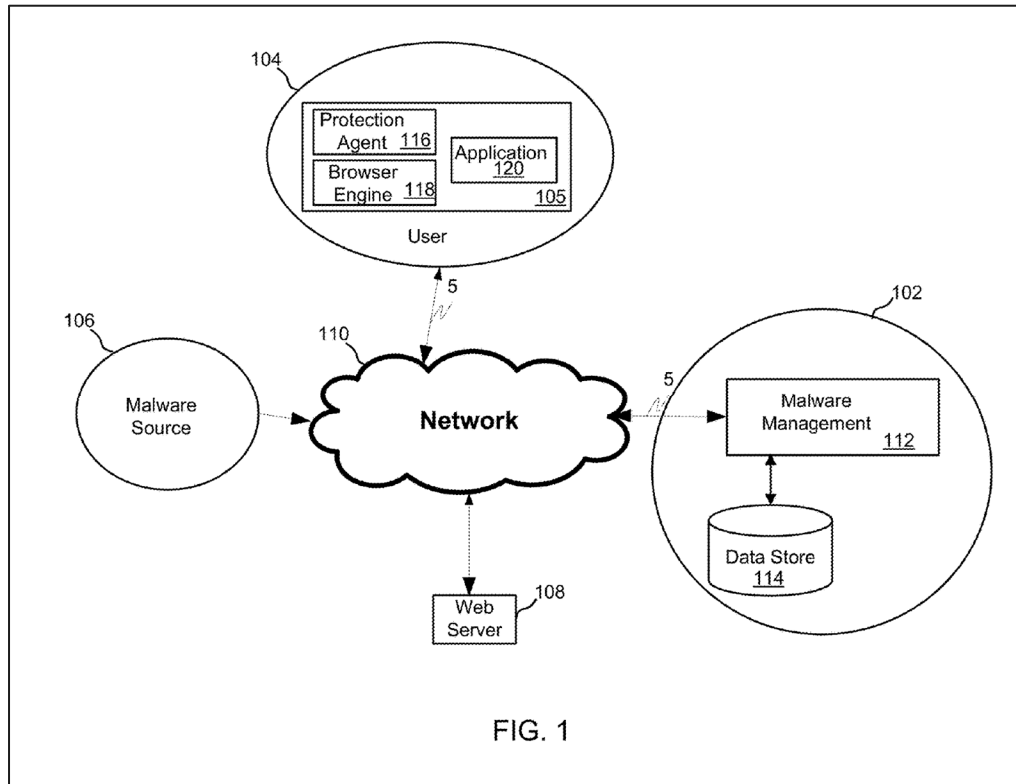
Devarajan teaches or renders obvious “*receiving, at the protection agent on a remote computer, web content from the one or more web servers, wherein web content comprises data for assembling a web page, and wherein the web content is received in response to the request.*” Ex.1002, ¶¶77-85. First, Devarajan discloses “*receiving, at the protection agent..., web content from the one or more web servers.*” *Id.*, ¶78. For example, after the request for content is transmitted to the hosting server in transmission step 1050, Devarajan’s Figure 10 depicts further receiving the content from a website 902 from the hosting server 903 as depicted in step 1060. This web content “is transmitted by the hosting server 903 responsive to receiving the request for content...from the server 109.” *Id.*, ¶78; Ex.1004, [0115]; [0119].



Ex.1004, Fig.10 (annotated).

Second, Devarajan discloses the “*protection agent*” may be on a “*remote computer*”—*i.e.*, the “receiving” step of claim 1 of the ’928 Patent may be performed by a (software) server 108 (“*protection agent*”) on the client 900’s computer (*e.g.*, a “*remote computer*”). Ex.1002, ¶79. For example, the ’928 Patent explains that the remote computer could be “a personal computer, a server of any time, a PDA, mobile phone, tablet, netbook, an interactive television, or any other device capable of loading and operating computer objects.” Ex.1001, 3:15-20; Ex.1002, ¶83. The ’928 Patent further discloses that the “remote computer” contains a protection agent

and a web browser application, as recited in claim 1 and shown in Figure 1. *See, e.g.,* Ex.1001, Fig.1 (at block 105), 3:2-20; Ex.1002, ¶83.



Ex.1001, Fig.1.

Like the protection agent in the '928 Patent, server 108 may be a software server in the same computer as the web browser. *See, e.g.,* Section VII.A.1.b; Ex.1004, [0053]; Ex.1002, ¶80. Thus, a POSITA would understand that server 108 may be software executing, *e.g.,* on a desktop, such as a client 900 and may be providing services to other software programs running on that computer. Ex.1002, ¶80; *see also* Ex.1004, [0092] (disclosing client 900 may be, *e.g.,* a desktop or laptop).

Accordingly, a POSITA would understand Devarajan to disclose an embodiment where a SmartProxy 905 and server 108 are software executing on the client 900 computer (*e.g.*, a desktop computer or laptop computer) to provide the functionality disclosed by the '928 Patent's claimed "*protection agent*." *See, e.g.*, Ex.1004, [0095]; Ex.1002, ¶81. A POSITA would understand this embodiment to disclose or render obvious "*a protection agent*," that comprises a (software) server 108 that is executed on the client computer ("*remote computer*"). Ex.1002, ¶81; *see also* Section VII.A.1.b.

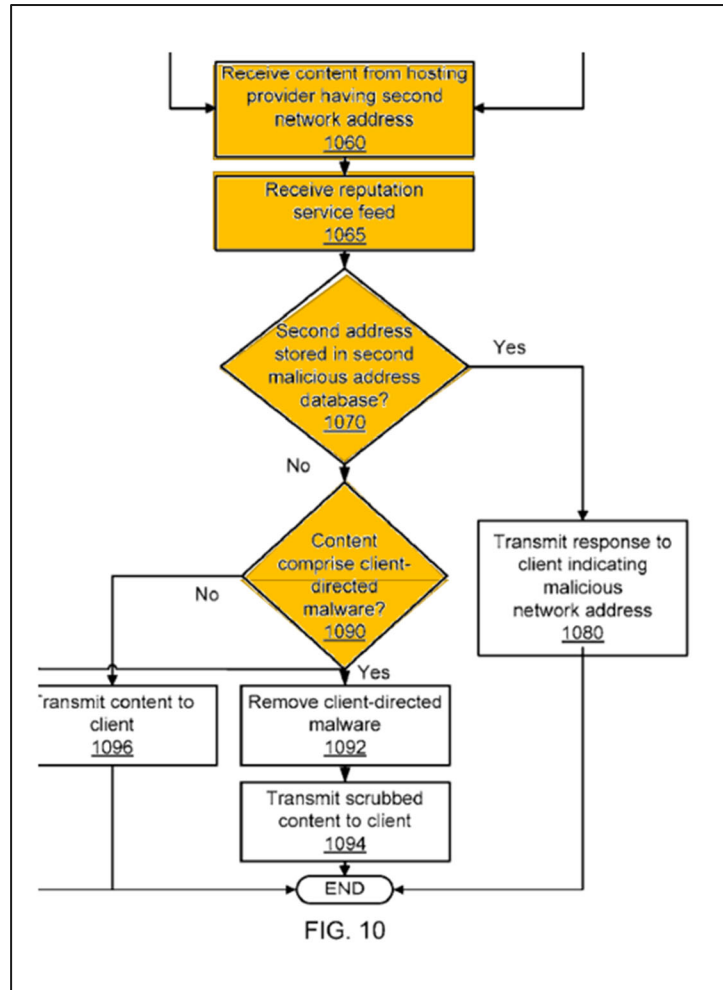
In addition, Devarajan discloses that the received "*web content comprises data for assembling a web page*." For example, Devarajan discloses the website 902 "*may comprise **any collection of data** and/or files accessible to a client 900 or server 108 communicatively coupled to the network 101. As a non-limiting example, the website 902 may comprise a single webpage or multiple interconnected and related webpages, perhaps resolving from a domain name, each of which may provide access to static, dynamic, multimedia, or any other content, perhaps by accessing files (*e.g.*, text, audio, video, graphics, executable, HTML, eXtensible Markup Language (XML). Active Server Pages (ASP), Hypertext Preprocessor (PHP), Flash files, server-side scripting, etc.) that enable the website 902 to display when rendered by a browser on a client 900.*" Ex.1004, [0090], [0023]; Ex.1002, ¶84. Accordingly, a POSITA would understand the web content 902 intercepted by the server 108,

from the host server 903 in response to a query to the host server 903 originating from the client 900 (as explained in Section VII.A.1.b) comprises “*data for assembling a web page.*” Ex.1002, ¶84. Devarajan also discloses “*the web content is received in response to the request,*” as the content is received at server 108 (“*protection agent*”) from the hosting server “responsive to receiving the request for content...from the server 108.” Ex.1004, [0115], [0119]; Ex.1002, ¶78, 82.

- g. [1F] identifying, by the protection agent, a malware threat within the web content;

Devarajan teaches or renders obvious “*identifying, by the protection agent, a malware threat within the web content.*” Once the server 108 receives the web content from the hosting server 903 (step 1060), Figure 10 shows a series of steps to “identify[...]...a malware threat within the web content.” Ex.1004, [0107], [0115]; Ex.1002, ¶¶86-89.

For example, at step 1060, Devarajan’s system receives the requested “content from hosting server 903.” Ex.1004, [0115]. Then, at step 1065, “a reputation feed may be received...from a...provider...having a second malicious network resource address database.” *Id.*, [0116]; Ex.1002, ¶87. At the next step (1070), the system determines whether the “hosting server’s 903 network resource address 904 is stored in the second malicious network resource address database.” Ex.1004, [0117]; Ex.1002, ¶87.



Ex.1004, Fig.10 (annotated).

If the system determines the network address is not stored in the database of known malware, it moves on to step 1090, which comprises “determining whether the content comprises a client-directed malware (e.g., a virus, a worm, a trojan horse, a rootkit, a backdoor, a spyware, a keystroke logger, a phishing application, a script injection, or an iFrame injection).” *Id.*, [0118]; Ex.1002, ¶88. If the content comprises client-directed malware, then the system moves on to step 1092, which involves “removing the client-directed malware from the content...and transmitting

a scrubbed content (e.g., website content having the client-directed malware removed) to the client.” Ex.1004, [0118]; Ex.1002, ¶88. As such, once the method at step 1090 determines there is client-directed malware—*i.e.*, the “yes” branch stemming from step 1090—Devarajan has “*identif[ie]d]...a malware threat within the web content.*” Ex.1002, ¶88. Moreover, Devarajan discloses this method may be performed by a (software) server 108 (“*protection agent*”) on client 900. *Id.*, ¶¶86-89.

- h. [1G] in response to identifying the malware threat, modifying, by the protection agent, the web content, wherein modifying the web content comprises modifying the web content at protocol level to remove the malware threat from the web content;

Devarajan teaches or renders obvious “*in response to identifying the malware threat, modifying, by the protection agent, the web content, wherein modifying the web content comprises modifying the web content at protocol level to remove the malware threat from the web content.*”

For example, when Devarajan determines the network address is not stored in the database of known malware, Figure 10 moves to step 1090, which “determine[es] whether the content comprises a client-directed malware.” Ex.1004, [0118]; Ex.1002, ¶91. If the content comprises client-directed malware (“*in response to identifying the malware threat*”), then the system moves to step 1092, which involves “removing the client-directed malware from the content...and transmitting

a scrubbed content (e.g., website content having the client-directed malware removed) to the client”—e.g., “*modifying...the web content.*” Ex.1004, [0118]; Ex.1002, ¶91. Moreover, Devarajan discloses this method may be performed by a (software) server 108 (“*protection agent*”) on client 900. See Section VII.A.1.b; Ex.1002, ¶¶89-91.

Devarajan also discloses “*wherein modifying the web content comprises modifying the web content at protocol level to remove the malware threat from the web content.*” For example, the ’928 Patent explains that modifying an HTML document is an example of a “protocol-level” modification. Ex.1001, 1:43-48, 4:6-11; Ex.1002, ¶92. A POSITA would understand HTML is HyperText Markup Language used for web pages and includes text and/or **links to other web content**. See, e.g., Ex.1001, 4:56-60; Ex.1002, ¶92. Similarly, Devarajan discloses, “[i]n another example, if the intrusion detection and prevention module 907 identifies **a link to a known malware website in the content**, the content sanitizer module 909 may either block the content or **remove the link from the content**. After the malware has been removed, the content sanitizer module 909 may transmit a ‘scrubbed’ content (e.g., the content having the client-directed malware removed) to the client 900.” Ex.1004, [0104]; Ex.1002, ¶92. A POSITA would therefore understand this disclosure regarding removing a link from web content in Devarajan to be performing a modification of web content at the protocol level. Ex.1002, ¶92.

Therefore, Devarajan teaches “*modifying the web content comprises modifying the web content at protocol level to remove the malware threat from the web content.*”

Ex.1002, ¶¶90-93.

- i. [1H] assembling, by the protection agent, a modified version of the web page, wherein the modified web page is assembled using the modified web content, and wherein the modified version of the web page does not comprise the malware threat; and

Devarajan teaches or renders obvious “*assembling, by the protection agent, a modified version of the web page, wherein the modified web page is assembled using the modified web content, and wherein the modified version of the web page does not comprise the malware threat.*” Devarajan teaches modifying the web content and transmitting the scrubbed content sans malware to the user. *See* Section VII.A.1.h; *see also* Ex.1004, [0104] (“After the malware has been removed, the content sanitizer module 909 may transmit a ‘scrubbed’ content (e.g., the content having the client-directed malware removed) to the client 900.”).

Moreover, a POSITA would understand the modifications described, for example, in *id.*, [0104], comprise “*assembling...a modified version of a web page, wherein the modified web page is assembled using the modified web content.*” Ex.1002, ¶94. There, removing the links from the web content, demonstrates modification, and sending “‘scrubbed’ content (e.g., the content having the client-directed malware removed) to the client 900” indicates the content has been

reassembled without the malware—*e.g.*, “*wherein the modified version of the web page does not comprise the malware threat.*” *Id.*, ¶¶94-96.

Devarajan discloses this method may be performed by a (software) server 108 (“*protection agent*”) on the client 900. *See* Section VII.A.1.b.

- j. [1I] providing, by the protection agent, the modified version of the web page to the web browser application on the remote computer for rendering and display.

Devarajan teaches or renders obvious “*providing, by the protection agent, the modified version of the web page to the web browser application on the remote computer for rendering and display.*” Devarajan discloses the server 108 (“*protection agent*”) modifying content from website 902 using the content sanitizer module and “after the malware has been removed” transmitting the “‘scrubbed’ content (*e.g.*, the content having the client-directed malware removed) to the client 900” wherein client 900 is the remote computer. Ex.1004, [0118]; Ex.1002, ¶97. Thus, Devarajan “*provid[es], by the protection agent, the modified version of the web page to...the remote computer.*”

Devarajan further teaches “[t]he website 902 may comprise any collection of data and/or files accessible to a client 900 or server 108 communicatively coupled to the network 101. As a non-limiting example, the website 902 may comprise a single webpage or multiple interconnected and related webpages, perhaps resolving from a domain name, each of which may provide access to static, dynamic,

multimedia, or any other content, perhaps by accessing files (e.g., text, audio, video, graphics, executable, HTML, eXtensible Markup Language (XML). Active Server Pages (ASP), Hypertext Preprocessor (PHP), Flash files, server-side scripting, etc.) that enable the website 902 to display when **rendered by a browser on a client 900.**” Ex.1004, [0090]; Ex.1002, ¶97. In other words, “*the modified version of the web page [is provided to] the web browser application on the remote computer for rendering and display.*” Ex.1002, ¶¶97-99.

Therefore, Devarajan renders claim 1 obvious.

2. Claim 2

- a. The method of claim 1, further comprising supplementing, by the protection agent, the web content with indicators regarding a vulnerability of one or more links within the web content.

Devarajan discloses or renders obvious “[t]he method of claim 1, further comprising supplementing, by the protection agent, the web content with indicators regarding a vulnerability of one or more links within the web content.” Ex.1002, ¶¶100-07.

Devarajan teaches that web content may include “*indicators regarding a vulnerability of one or more links within the web content*” through the use of its reputation feed at step 1065 of Figure 10. *See* Ex.1004, [0116]. The reputation feed “may be implemented” via the “network resource address reputation service.” *Id.*; Ex.1002, ¶101. The reputation service “generat[es] a reputation score for the

network resource address.” Ex.1004, [0074]; Ex.1002, ¶105. “The reputation score may comprise any score indicating the reputation for the network resource address,” including a score that “a range from 0 to 1, 1 to 10, 0% to 100%, [] A+ to F-,” and/or a color rating system, *e.g.*, with “red” indicating a “poor reputation.” Ex.1004, [0074]. This reputation score (*e.g.*, “*an indicator*”) provides a metric for the “*vulnerability of one or more links within the web content.*” Ex.1002, ¶¶105-06. For example, Devarajan explains that the reputation score may be modified “to indicate a more negative reputation for the network resource address.” Ex.1004, [0076]; Ex.1002, ¶106. “The modified reputation then may be stored...in the second malicious network resource address database 112.” *Id.*

Devarajan further teaches “*supplementing...the web content with indicators.*” For example, after generating a reputation score step 1065 and storing that score in the second malicious network resource address database at step 1070, “a response may be transmitted to the client 900 indicating that the second network resource address 904 is stored in the second malicious network resource address database 112 (Step 1080).” Ex.1004, [0076], [0117]; Ex.1002, ¶102-106. In short, Devarajan’s Figure 10 includes a step of transmitting a message (*e.g.*, “*an indicator*”) to the client 900 that the hosting server or its address have been associated with a malicious network address. Ex.1002, ¶102.

Devarajan also teaches “*supplementing...the web content with indicators regarding a vulnerability of one or more links within the web content,*” wherein Devarajan “identifies a link to a known malware website in the content” (e.g., identifies “*a vulnerability of one or more links within the web content*”) and “either block(s) the content or remove(s) the link from the content” (e.g., “*supplementing...the web content with indicators*”). Ex.1004, [0104]; Ex.1002, ¶106. The blocking or removal of the links are examples of “indicators.” Ex.1004, [0104]; Ex.1002, ¶106.

Moreover, a POSITA would understand in the likely instances where web content includes links to content that is also hosted at hosting server 903, the server 108 would supplement the web content being returned to the client 900 with indicators regarding a vulnerability of one or more links within the web content. Ex.1002, ¶103. For example, Devarajan teaches removing links to known malware sites (*see* Section VII.A.1) and teaches supplementing links with reputation scores (such as numbers, letter grades, or colors), such that a POSITA would find it obvious from Devarajan to supplement the web content (e.g., for links with poor reputation but not necessarily known malware) with link reputation scores as taught in Devarajan. *Id.*, ¶¶105-06.

Devarajan further teaches that the “*supplementing*” may be carried out “*by the protection agent.*” For example, Devarajan explains that “generating a reputation

score for the network address...may be accomplished by a server 108”—*i.e.*, by the “*protection agent.*” Ex.1004, [0074]; Ex.1002, ¶¶101-02.

3. Claim 3

- a. The method of claim 1, wherein an analysis to identify a malware threat within the web content includes one or more of a URL (Uniform Resource Locator) analysis, an IP (Internet Protocol) analysis, an image analysis, and a script/HTML analysis.

Devarajan discloses or renders obvious “[*t*]he method of claim 1, wherein an analysis to identify a malware threat within the web content includes one or more of a URL (Uniform Resource Locator) analysis, an IP (Internet Protocol) analysis, an image analysis, and a script/HTML analysis.” Ex.1002, ¶¶108-11.

Devarajan’s “*analysis to identify a malware threat within the web content*” includes at least “*an IP (Internet Protocol) analysis.*” *Id.*, ¶109. For example, Devarajan discloses transmitting “a response...to the client 900... indicating that the hosting server 903 or **its IP address** (i.e., second network resource address 904) have been **associated with a malicious network address.**” Ex.1004, [0117]; Ex.1002, ¶109. A POSITA would understand this to mean that the malware threat analysis includes “an IP (Internet Protocol) analysis” to determine the presence of malware. Ex.1002, ¶¶104, 109.

Devarajan’s “*analysis to identify a malware threat within the web content*” also includes at least “*a script/HTML analysis.*” For example, “the intrusion

detection and prevention module 907 **identifies a link to a known malware website in the content**”—e.g., script and/or HTML. Ex.1004, [0104]; Ex.1002, ¶¶108-111. Devarajan also “determine[es] whether the content comprises a client-directed malware (e.g., ...**a script injection** . . .” at step 1090 of Figure 10. Ex.1004, [0118].

4. Claim 4

- a. The method of claim 1, further comprising determining, by the protection agent, whether a request for the web content from the web browser application is a first request.

Devarajan discloses or renders obvious “[t]he method of claim 1, further comprising determining, by the protection agent, whether a request for the web content from the web browser application is a first request.” Ex.1002, ¶¶112-15.

Devarajan “determin[es]...whether a request for the web content from the web browser application is a first request” by first determining if a request for web content is stored in a malicious resource address database 112. Ex.1004, [0117] (“determin[ing] that the hosting server’s 903 network resource address 904 is stored in the second malicious network resource address database 112”); Ex.1002, ¶113. If the request is not stored to this database, Devarajan determines whether to add the address to a blacklist of blocked addresses. Ex.1004, [0077] (if “the network resource addresses 104 reputation score exceeds a predetermined value (Step 500)...adding the network resource address 104 to a blacklist (Step 510)”); Ex.1002, ¶114. A POSITA would understand Devarajan is disclosing that a first request for

web content from a web browser to a network resource address with a poor reputation score will, in a first instance, be considered, and the system will add that network resource address to a blacklist, thereby precluding future communication with that network resource address. Ex.1002, ¶115. So, depending on whether a network resource address for a host server with a poor reputation score is in the blacklist or not, the server 108 (“*protection agent*”) determines whether this request for content is a first request. *Id.*

5. Claim 6

- a. The method of claim 1, wherein a protocol-level analysis is utilized to identify the malware threat within the web content.

Devarajan discloses or renders obvious “[t]he method of claim 1, wherein a protocol-level analysis is utilized to identify the malware threat within the web content.” Devarajan discloses “modifying the web content comprises modifying the web content at protocol level to remove the malware threat from the web content.” See Section VII.A.1.h.

Not only does Devarajan teach “modifying” at the protocol-level, Devarajan also teaches “a protocol-level analysis is utilized to identify the malware threat within the web content.” Ex.1002, ¶¶117-18. For example, Step 1090 of Figure 10 “determine[es] whether the content comprises a client-directed malware (e.g., a virus, a worm, a trojan horse, a rootkit, a backdoor, a spyware, a keystroke logger, a

phishing application, a script injection, or an iFrame injection).” Ex.1004, [0118], Ex.1002, ¶117. If the content comprises client-directed malware, then the system moves on to step 1092, which involves “removing the client-directed malware from the content...and transmitting a scrubbed content (e.g., website content having the client-directed malware removed) to the client.” Ex.1004, [0118]; Ex.1002, ¶117.

The '928 Patent explains that analyzing an HTML document qualifies as “protocol-level” analysis. Ex.1001, 1:43-48, 4:6-11; Ex.1002, ¶118. A POSITA would understand HTML is HyperText Markup Language used for web pages and includes text and/or links to other web content. *See, e.g.*, Ex.1001, 4:56-64; Ex.1002, ¶118. Likewise, in Devarajan, “if the intrusion detection and prevention module 907 identifies a link to a known malware website in the content, the content sanitizer module 909 may either block the content or remove the link from the content. After the malware has been removed, the content sanitizer module 909 may transmit a ‘scrubbed’ content (e.g., the content having the client-directed malware removed) to the client 900.” Ex.1004, [0104]; Ex.1002, ¶118. A POSITA, therefore, would understand that identifying a link to a known malware website in web content in Devarajan to be performing an analysis of web content at the protocol level identified in the '928 Patent. Ex.1002, ¶¶116-19.

Moreover, software server 108 (“*protection agent*”) executes these steps on client 900 (“*remote computer*”). *Id.*, ¶117; *see* Section VII.A.1.b.

6. Claim 7

- a. The method of claim 3, further comprising performing cloud verification of the web content, wherein cloud verification determines one or more malware vulnerabilities based on an aggregation of the analyses.

Devarajan discloses or renders obvious “[t]he method of claim 3, further comprising performing cloud verification of the web content, wherein cloud verification determines one or more malware vulnerabilities based on an aggregation of the analyses.” Ex.1002, ¶120-23.

As described in Section VII.A.1, Devarajan teaches “determin[ing] one or more malware vulnerabilities based on an aggregation of the analyses.” Devarajan explains this “determining” step may be carried out in a cloud-computing environment. Ex.1004, [0054] (“The instructions 110 may...comprise Software and/or scripts stored in the computer-readable media 109...accessible via the network 101, perhaps via a grid or **cloud-computing environment**”); Ex.1002, ¶121. A POSITA would understand carrying out the instructions/analysis in a “cloud-computing environment” to imply “cloud verification of web content.” Ex.1002, ¶121.

Further, Devarajan discloses server 108 determines web content “wherein cloud verification determines one or more malware vulnerabilities based on an aggregation of the analyses,” including “determining whether the content comprises a client-directed malware (e.g., a virus, a worm, a trojan horse, a rootkit, a backdoor,

a spyware, a keystroke logger, a phishing application, a script injection, or an iFrame injection) (Step 1090).” Ex.1004, [0118]; Ex.1002, ¶122; *see also* Section VII.A.3.

7. Claim 9

- a. [9P] A content analysis and malware prevention system comprising:

Element [9P] is disclosed by Devarajan. *See* Section VII.A.1.a. Ex.1002, ¶124.

- b. [9A] at least one processor; and

Devarajan discloses a “*processor*.” For example, Devarajan states in Figure 1, “the server 108 may comprise a computer-readable storage media 109 storing instructions 110 that, when executed by a **microprocessor** 111, cause the server 108 to perform the steps for which it is configured.” Ex.1004, [0054]; Ex.1002, ¶125.

- c. [9B] memory encoding computer executable instructions that, when executed by the at least one processor, perform a method comprising:

Devarajan discloses “*memory encoding computer executable instructions that, when executed by the at least one processor, perform a method comprising*.” For example, Devarajan states, “[a]s further illustrated in FIG. 1, the server 108 may comprise a **computer-readable storage media 109 storing instructions 110 that, when executed by a microprocessor 111**, cause the server 108 to perform the steps for which it is configured.” Ex.1004, [0054]; Ex.1002, ¶126.

- d. [9C] intercepting a request from a web browser;

Element [9C] is rendered obvious by Devarajan. *See* Section VII.A.1.b.
Ex.1002, ¶127.

- e. [9D] determining whether the request is associated with known malicious content;

Element [9D] is rendered obvious by Devarajan. *See* Section VII.A.1.c.
Ex.1002, ¶128.

- f. [9E] when the request is associated with known malicious content, blocking, by a protection agent, the request;

Element [9E] is rendered obvious by Devarajan. *See* Section VII.A.1.d.
Ex.1002, ¶129.

- g. [9F] when the request is not associated with known malicious content: sending the request to one or more web servers;

Element [9F] is rendered obvious by Devarajan. *See* Section VII.A.1.e.
Ex.1002, ¶130.

- h. [9G] receiving web content from one or more web servers, wherein web content comprises data for assembling a web page;

Element [9G] is rendered obvious by Devarajan. *See* Section VII.A.1.f.
Ex.1002, ¶131.

- i. [9H] identifying a malware threat within the web content;

Element [9H] is rendered obvious by Devarajan. *See* Section VII.A.1.g.
Ex.1002, ¶132.

- j. [9I] in response to identifying the malware threat, modifying the web content, wherein modifying the web content comprises modifying the web at protocol level to remove the malware threat from the web content;

Element [9I] is rendered obvious by Devarajan. *See* Section VII.A.1.h.

Ex.1002, ¶133.

- k. [9J] assembling a modified version of the web page, wherein the modified web page is assembled using the modified web content, and wherein the modified version of the web page does not comprise the malware threat; and

Element [9J] is rendered obvious by Devarajan. *See* Section VII.A.1.i.

Ex.1002, ¶134.

- l. [9K] providing the modified version of the web page to a web browser application on the remote computer for rendering and display.

Element [9K] is rendered obvious by Devarajan. *See* Section VII.A.1.j.

Ex.1002, ¶135.

Therefore, Devarajan renders claim 9 obvious.

8. Claim 10

- a. The system of claim 9, wherein the method further comprises supplementing the web content with indicators regarding a vulnerability of one or more links within the web content.

Claim 10 is rendered obvious by Devarajan. *See* Section VII.A.2. Ex.1002,

¶136.

9. Claim 11

- a. The system of claim 9, wherein the method further comprises identifying the malware threat within the web content by performing one or more of a URL (Uniform Resource Locator) analysis, an IP (Internet Protocol) analysis, an image analysis, and a script/HTML analysis.

Claim 11 is rendered obvious by Devarajan. *See* Section VII.A.3. Ex.1002,

¶137.

10. Claim 12

- a. The system of claim 9, wherein the method further comprises determining whether a request for the web content from the web browser application is a first request.

Claim 12 is rendered obvious by Devarajan. *See* Section VII.A.4. Ex.1002,

¶138.

11. Claim 14

- a. The system of claim 9, wherein the method further comprises identifying the malware threat utilizing a protocol-level analysis.

Claim 14 is rendered obvious by Devarajan. *See* Section VII.A.5. Ex.1002,

¶149.

12. Claim 16

- a. The method of claim 1, wherein the malware threat comprises a Uniform Resource Locator (URL).

Devarajan discloses or renders obvious “[t]he method of claim 1, wherein the malware threat comprises a Uniform Resource Locator (URL).” *See* Section VII.A.1.

For example, Devarajan teaches an intrusion detection and prevention module, which analyzes the content on a page if it “identifies a link to a known malware website in the content, the content sanitizer module 909 may either block the content or remove the link from the content.” Ex.1004, [0104]; Ex.1002, ¶141. A POSITA would understand a “link” to be inclusive of a URL. Ex.1002, ¶141. Devarajan further teaches, “[a]n external feed 107 may comprise a malware domain list feed, a malware URL list feed, an emerging threat feed, an intrusion detection feed, a botnet tracking feed, a phishing tracking feed, a spam tracking feed, or a compromised network feed.” Ex.1004, [0052]; Ex.1002, ¶141. In addition, Devarajan teaches, “[a]s non-limiting examples, the network resource address may comprise an IP address, a URL, or a domain (e.g., domain name) for the network resource 103.” Ex.1004, [0049]; Ex.1002, ¶141. Given the teachings in Devarajan, (*see, e.g.*, Ex.1004, [0049], [0052], [0104]) a POSITA would understand this analysis comprises a URL. Ex.1002, ¶¶140-41.

13. Claim 17

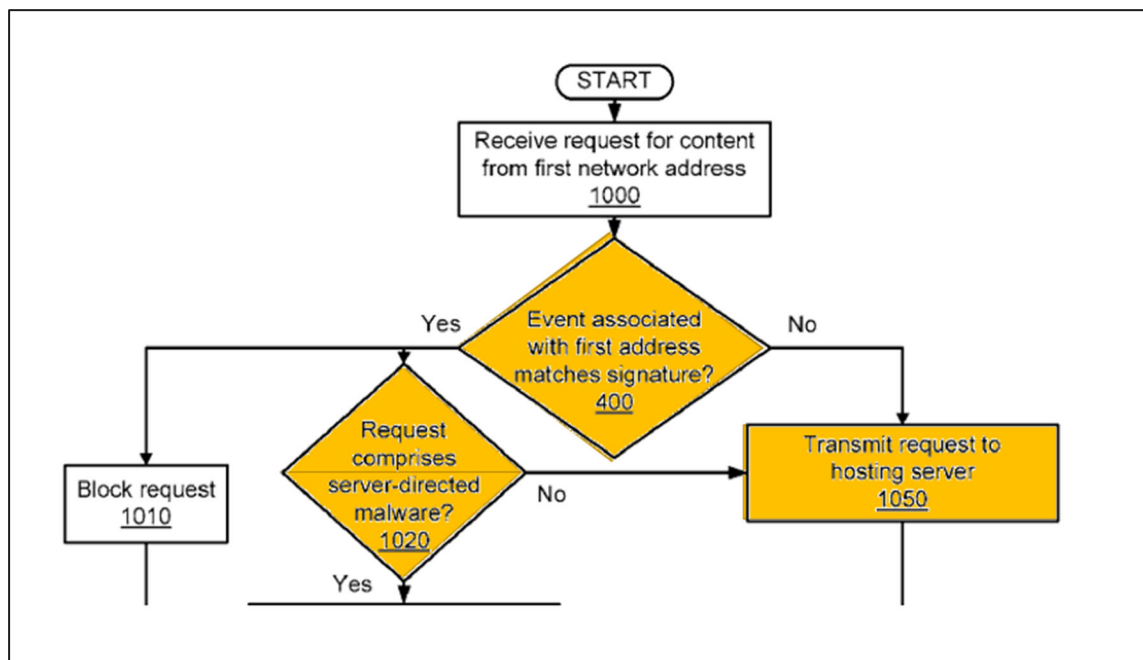
- a. The system of claim 9, wherein the method further comprises determining if a malware threat comprises a Uniform Resource Locator (URL).

Claim 17 is rendered obvious by Devarajan. *See* Section VII.A.12. Ex.1002, ¶142.

14. Claim 18

- a. The method of claim 1, wherein sending the request to one or more web servers comprises iteratively sending a plurality of requests to the one or more web servers.

Devarajan discloses “[t]he method of claim 1, wherein sending the request to one or more web servers comprises iteratively sending a plurality of requests to the one or more web servers.” See Section VII.A.1. For example, Figure 10 illustrates a “method for bi-directional detection and removal of network traffic malware.” Ex.1004, [0109]; Ex.1002, ¶143.



Ex.1004, Fig.10 (annotated).

In Figure 10, the process begins when “the content request [is] received by...[the] system,” as shown in block 1000. Ex.1004, [0110]; Ex.1002, ¶144. At the next block (400), the system determines “whether an event associated with the

client 900 or its first network resource address 901 matches one or more of a plurality of event signatures 102.” Ex.1004, [0111]; Ex.1002, ¶144. These “signatures 102” are “simply known attack patterns” and “comprise a plurality of malware signatures including, but not limited to a virus signature, a worm signature, a trojan horse signature, a rootkit signature, a backdoor signature, a spyware signature, a keystroke logger signature, or a phishing application signature.” Ex.1004, [0045]-[0046]; Ex.1002, ¶144. If at step 400 the system determines “the event does not match an event signature 102, the request for content may be transmitted...to the hosting server 903” as depicted in step 1050. Ex.1004, [0111]; Ex.1002, ¶144. Moreover, Devarajan discloses that “the SmartProxy 905 may be configured (e.g., programmed) to receive website 902 content **requests** from the client 900 and redirect such **requests** to the scrubbing center 906.” Ex.1004, [0106]; *see also* claims 17 and 21; Ex.1002, ¶144.

A POSITA would understand the above disclosures to mean a request for content may be sent iteratively by the server 108 to comprise a plurality of requests to the hosting server 903. Ex.1002, ¶143-145.

15. Claim 19

- a. The method of claim 1, further comprising processing the web content prior to identifying the malware threat within the web content.

Devarajan discloses or renders obvious “[t]he method of claim 1, further comprising processing the web content prior to identifying the malware threat within the web content.” Devarajan discloses processing the web content by analyzing its IP address prior to determining whether or not it comprises malware. Ex.1004, [0117]; Ex.1002, ¶¶146-48.

16. Claim 20

- a. The system of claim 9, wherein sending the request to one or more web servers comprises iteratively sending a plurality of requests to the one or more web servers.

Claim 20 is rendered obvious by Devarajan. *See* Section VII.A.14. Ex.1002, ¶149.

B. GROUND 2: DEVARAJAN IN COMBINATION WITH HUBBARD RENDERS OBVIOUS CLAIMS 1-4, 6-7, 9-12, 14, and 16-20

1. Claim 1

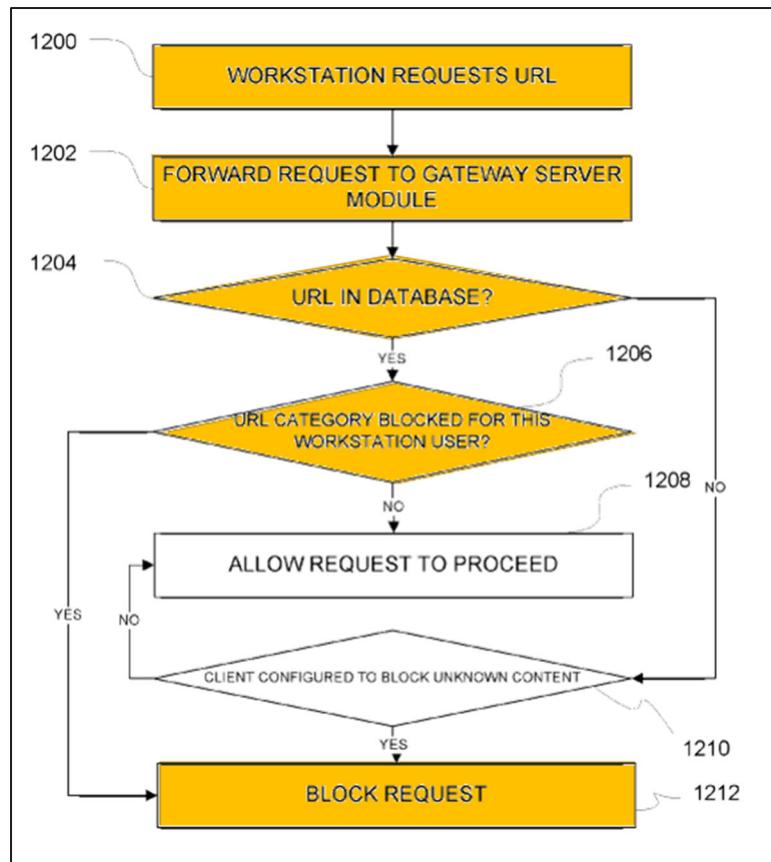
- a. [1P] A content analysis and malware prevention method comprising:

Element [1P] is rendered obvious by Devarajan in view of Hubbard. *See* Section VII.A.1.a. Ex.1002, ¶150.

- b. [1A] intercepting, by a protection agent, a request from a web browser;

Devarajan discloses or renders obvious “*intercepting, by a protection agent, a request from a web browser.*” See Section VII.A.1.b.

Hubbard also discloses “*intercepting...a request from a web browser.*” Hubbard discloses a system that comprises a “gateway server module 120,” e.g., a “software module that is installed on a server that stands as a gateway to a wider area network 112,” that is connected to “workstation 116.” Ex.1008, 4:54-58; Ex.1002, ¶152. The “gateway server module” is “in two way communication with the workstation 116,” such that requests from workstation(s) 116 “may be required to pass through the gateway server module 120 as they proceed to the Internet.” Ex.1008, 6:23-26; Ex.1002, ¶152. This request “is **intercepted** . . . and forwarded to the gateway server module 120 at block 1202.” Ex.1008, 22:54-56; Ex.1002, ¶152. At block 1206, the gateway server module determines “whether the category of the URL is one that should be blocked for the workstation user.” Ex.1008, 22:58-62; Ex.1002, ¶152.



Ex.1008, Fig. 16 (annotated).

A POSITA would be motivated to combine the “*protection agent*” of Devarajan with the “*intercepting*” methods of Hubbard because it would allow the system to further its goal of tackling malware before it reaches the user. Ex.1002, ¶¶153.

Like Devarajan, Hubbard teaches systems and methods for “controlling access to web content” in order to avoid URL links “that lead to webpages that are categorized as being ‘Malicious’ or ‘Spyware.’” Ex.1008, 2:41-42, 8:46-49; Ex.1002, ¶154. Hubbard teaches that the increased sophistication of newer web applications has created an environment where “malicious code and applications...

[can] download themselves to users' machines" without any affirmative action by the user. Ex.1008, 1:53-2:2; Ex.1002, ¶154. Hubbard notes that traditional malware blocking methods like "heightened security setting on [a] web browser" are impractical as they "impede or obstruct effect use of [various web applications]." Ex.1008, 2:8-15; Ex.1002, ¶154. Thus, Hubbard discloses a need for malware detection and potential removal without compromising the efficiency of users' web based applications. Devarajan detects malware based on known malware network addresses stored in its databases, and in the event malware is detected, Devarajan teaches removing the malware from the content request. Ex.1004, [0004], [0007]; Ex.1002, ¶154. Devarajan's malware detection system can detect malware using "signature-based, statistical anomaly-based, and stateful protocol analysis methods." Ex.1004, [0044]; Ex.1002, ¶154. Hubbard further teaches URL analysis and identification. Ex.1008, 2:41-52; Ex.1002, ¶154. Devarajan notes that a "URL may include a Hypertext Transfer Protocol (HTTP) request combined with the website's Internet address." Ex.1004, [0026]; Ex.1002, ¶154. To further enhance Devarajan's protocol analysis malware detection system, a POSITA in the field of network security would have had a reasonable expectation of success in turning to references like Hubbard, which specifically teach analyzing and categorizing URLs. Ex.1002, ¶¶153-55. A POSITA could easily adapt Hubbard's analysis methods to work with

the protection agent system taught by Devarajan, because the systems would work together. *Id.*

Thus, Devarajan in view of Hubbard renders this claim element obvious.
Ex.1002, ¶¶151-55.

- c. [1B] determining, by the protection agent, whether the request is associated with known malicious content;

Element [1B] is rendered obvious by Devarajan in view of Hubbard. *See* Section VII.A.1.c. Ex.1002, ¶156.

- d. [1C] when the request is associated with known malicious content, blocking, by the protection agent, the request;

Element [1C] is rendered obvious by Devarajan in view of Hubbard. *See* Section VII.A.1.d. Ex.1002, ¶157.

- e. [1D] when the request is not associated with known malicious content: sending, by the protection agent, the request to one or more web servers

Element [1D] is rendered obvious by Devarajan in view of Hubbard. *See* Section VII.A.1.e. Ex.1002, ¶158.

- f. [1E] receiving, at the protection agent on a remote computer, web content from the one or more web servers, wherein web content comprises data for assembling a web page, and wherein the web content is received in response to the request;

Element [1E] is rendered obvious by Devarajan in view of Hubbard. *See* Section VII.A.1.f. Ex.1002, ¶159.

- g. [1F] identifying, by the protection agent, a malware threat within the web content;

Element [1F] is rendered obvious by Devarajan in view of Hubbard. *See* Section VII.A.1.g. Ex.1002, ¶160.

- h. [1G] in response to identifying the malware threat, modifying, by the protection agent, the web content, wherein modifying the web content comprises modifying the web content at protocol level to remove the malware threat from the web content;

Element [1G] is rendered obvious by Devarajan in view of Hubbard. *See* Section VII.A.1.h. Ex.1002, ¶161.

- i. [1H] assembling, by the protection agent, a modified version of the web page, wherein the modified web page is assembled using the modified web content, and wherein the modified version of the web page does not comprise the malware threat; and

Element [1H] is rendered obvious by Devarajan in view of Hubbard. *See* Section VII.A.1.i. Ex.1002, ¶162.

- j. [1I] providing, by the protection agent, the modified version of the web page to the web browser application on the remote computer for rendering and display.

Element [1I] is rendered obvious by Devarajan in view of Hubbard. *See* Section VII.A.1.j. Ex.1002, ¶163.

2. Claims 2-4 and 6-7

Claims 2-4 and 6-7 are rendered obvious by Devarajan in view of Hubbard. *See* Sections VII.A.2-VII.A.6. Ex.1002, ¶¶164-168.

3. Claims 9-12, 14 and 16-20

Claims 9-12, 14 and 16-20 are rendered obvious by Devarajan in view of Hubbard. *See* Sections VII.B.1, VII.A.7-VII.A.16. Ex.1002, ¶¶181-189.

C. GROUND 3: DEVARAJAN IN COMBINATION WITH MCAFEE RENDERS OBVIOUS CLAIMS 1-20

1. Claim 1

- a. [1P] A content analysis and malware prevention method comprising:

Element [1P] is disclosed or rendered obvious by Devarajan. *See* Section VII.A.1.a; Ex.1002, ¶190.

- b. [1A] intercepting, by a protection agent, a request from a web browser;

Element [1A] is disclosed or rendered obvious by Devarajan. *See* Section VII.A.1.b.

McAfee also discloses “*intercepting...a request from a web browser.*” McAfee discloses a “service provider” that “**redirect[s]** the...web traffic of the user” (e.g., “*requests from a web browser*”) through its servers, thus acting as an “intermediary” in order to “scan the page for malware” and “process the page” before it is served to the user. Ex.1005, [0035]; Ex.1002, ¶195. A POSITA would understand that by acting as an “intermediary” and “redirect[ing]” “*requests from a web browser,*” McAfee teaches intercepting a user’s requests in order to evaluate

the results for potential malware before it is served to the user in a manner akin to the claimed invention in the '928 Patent. Ex.1002, ¶195.

A POSITA would have been motivated to combine the “*protection agent*” of Devarajan with the “*intercepting*” methods of McAfee because it would allow the system to further its goal of tackling malware before it reaches the user. Ex.1002, ¶¶193-94.

Like Devarajan, McAfee teaches systems and methods for malware detection and prevention. Ex.1005, [0005]; Ex.1002, ¶193. McAfee teaches malware may infect a user’s computer through a variety of means such as when a user visits a web page that “installs or otherwise executes the malware on the computer” or when a user “downloads and installs software from a malicious web page.” Ex.1005, [0002]; Ex.1002, ¶194. Devarajan detects malware based on known malware network addresses stored in its databases, and in the event malware is detected, Devarajan teaches removing the malware from the content request. Ex.1004, [0004], [0007]; Ex.1002, ¶194. Devarajan’s malware detection system can detect malware using “signature-based, statistical anomaly-based, and stateful protocol analysis methods.” Ex.1004, [0044]; Ex.1002, ¶194. McAfee further teaches page processing methods that include URL analysis. Ex.1005, [0072]; Ex.1002, ¶194. A POSITA in the field of network security would understand the importance of thorough malware detection, including by analyzing individual links and data in a

web page, and would have had a reasonable expectation of success in turning to references like McAfee, which teach a variety of malware detection methods, to further build on the teachings of Devarajan. Ex.1002, ¶¶193-94. A POSITA could easily adapt McAfee’s malware detection methods to work with the protection agent system taught by Devarajan, because the systems would work together. *Id.*

Thus, Devarajan in view of McAfee renders this limitation obvious. Ex.1002, ¶¶1191-96.

- c. [1B] determining, by the protection agent, whether the request is associated with known malicious content;

Element [1B] is disclosed or rendered obvious by Devarajan. *See* Section VII.A.1.c. Ex.1002, ¶197.

- d. [1C] when the request is associated with known malicious content, blocking, by the protection agent, the request;

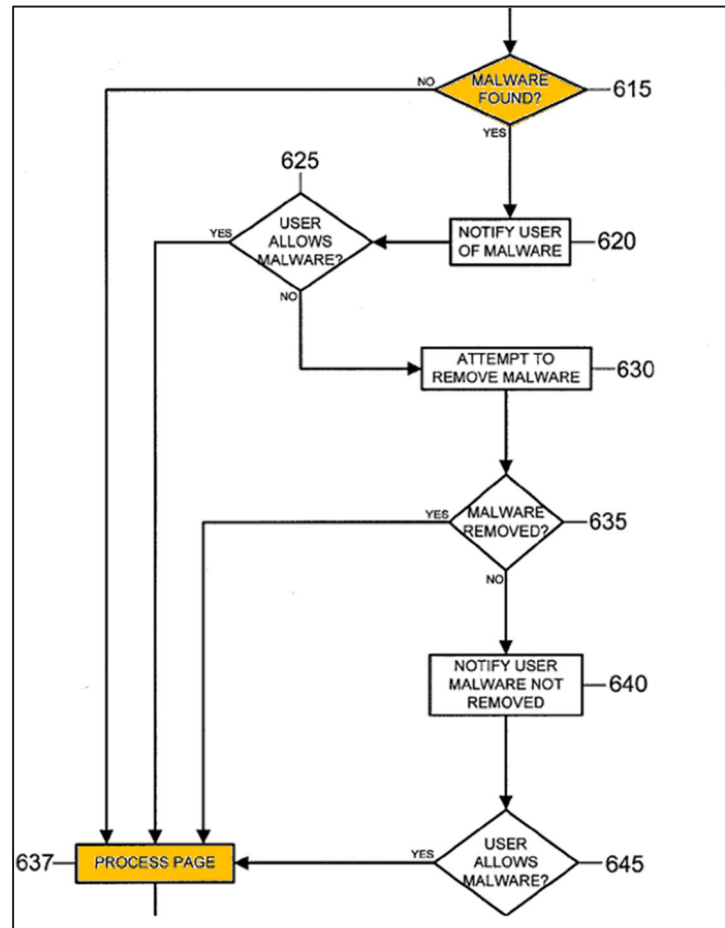
Element [1C] is disclosed or rendered obvious by Devarajan. *See* Section VII.A.1.d. Ex.1002, ¶198.

- e. [1D] when the request is not associated with known malicious content: sending, by the protection agent, the request to one or more web servers

Element [1D] is disclosed or rendered obvious by Devarajan. *See* Section VII.A.1.e.

McAfee, as depicted in Figure 6, also teaches that if no malware is found at block 615 (“*when the request is not associated with known malicious content*”), the request is moved to block 637, where “the page may be further processed by the

page processor.” Ex.1005, [0082]; Ex.1002, ¶200. McAfee teaches the page processor “may retrieve the page from the content provider server,” which constitutes “*sending...the request to one or more web server*” in order to retrieve the page’s content. Ex.1005, [0077]; *see also id.*, [0005], [0082]; Ex.1002, ¶201.



Ex.1005, Fig. 6 (annotated).

For at least the reasons explained above in claim element [1A], *see* Section VII.B.1.b, above, a POSITA would be motivated to combine the “*protection agent*” of Devarajan with the method of “*when the request is not associated with known*

malicious content: sending...the request to one or more web servers” in McAfee.

Ex.1002, ¶¶199-202.

- f. [1E] receiving, at the protection agent on a remote computer, web content from the one or more web servers, wherein web content comprises data for assembling a web page, and wherein the web content is received in response to the request;

Element [1E] is disclosed or rendered obvious by Devarajan. *See* Section VII.A.1.f. Ex.1002, ¶203.

- g. [1F] identifying, by the protection agent, a malware threat within the web content;

Element [1F] is disclosed or rendered obvious by Devarajan. *See* Section VII.A.1.g. Ex.1002, ¶204.

- h. [1G] in response to identifying the malware threat, modifying, by the protection agent, the web content, wherein modifying the web content comprises modifying the web content at protocol level to remove the malware threat from the web content;

Element [1G] is disclosed or rendered obvious by Devarajan. *See* Section VII.A.1.h. Ex.1002, ¶205.

- i. [1H] assembling, by the protection agent, a modified version of the web page, wherein the modified web page is assembled using the modified web content, and wherein the modified version of the web page does not comprise the malware threat; and

Element [1H] is disclosed or rendered obvious by Devarajan. *See* Section VII.A.1.i. Ex.1002, ¶206.

- j. [1I] providing, by the protection agent, the modified version of the web page to the web browser application on the remote computer for rendering and display.

Element [1I] is disclosed or rendered obvious by Devarajan. *See* Section VII.A.1.j. Ex.1002, ¶¶207-209.

2. Claim 2

Claim 2 is rendered obvious by Devarajan in view of McAfee. *See* Section VII.A.2; Ex.1002, ¶¶210-14.

McAfee, for example, describes “*indicators regarding a vulnerability of one or more links within the web content*” that are included within the web page itself. Ex.1005, [0082]; Ex.1002, ¶211. At block 615 of Figure 6, “if the page processor 312 finds malware on the page,” the system moves to block 620. Ex.1005, [0082]; Ex.1002, ¶212. At block 620, “the service provider server 240 may notify the user A 120A that the page contains malware” including “the type of malware, and generally any information that may describe the malware.” Ex.1005, [0082]; Ex.1002, ¶212. McAfee also teaches, “[w]hen the user performs a search on a supported search engine the software may, for example, consult a ‘hot list’ of web sites known to promulgate malware to advise the user as to whether the web pages referenced in the search results may contain malware.” Ex.1005, [0003]; Ex.1002, ¶212.

McAfee also teaches “*supplementing...the web content with [the] indicators*” in the web page returned to a user. Ex.1005, [0036]; Ex.1002, ¶213. McAfee notifies the user of the malware, and provides the user with the option of “browsing the page with the malware intact.” Ex.1005, [0036].

A POSITA would be motivated to combine the “*protection agent*” of Devarajan with method of “*supplementing...the web content with indicators regarding a vulnerability of one or more links within the web content*” at least for the reasons described in Section VII.B.1, above.

3. Claim 3

Claim 3 is rendered obvious by Devarajan in view of McAfee. *See* Section VII.A.3; Ex.1002, ¶¶215-17.

McAfee’s “*identification of]...a malware threat within the web content includes one or more of a URL (Uniform Resource Locator) analysis...and a script/HTML analysis.*” For example, McAfee includes a “URL processor” that is “operative to encode and decode the URLs on a page.” Ex.1005, [0063]; Ex.1002, ¶216. These URLs may be “encoded by the URL processor 314 to ensure each of the URLs redirects the user A120A through the service provider server 240.” Ex.1005, [0063]; Ex.1002, ¶216. McAfee further discloses a “safe search” mode that provides the user with safe web based interactions. Ex.1005, [0087]; Ex.1002, ¶216. This system teaches “exemplary HTML code for displaying safe search

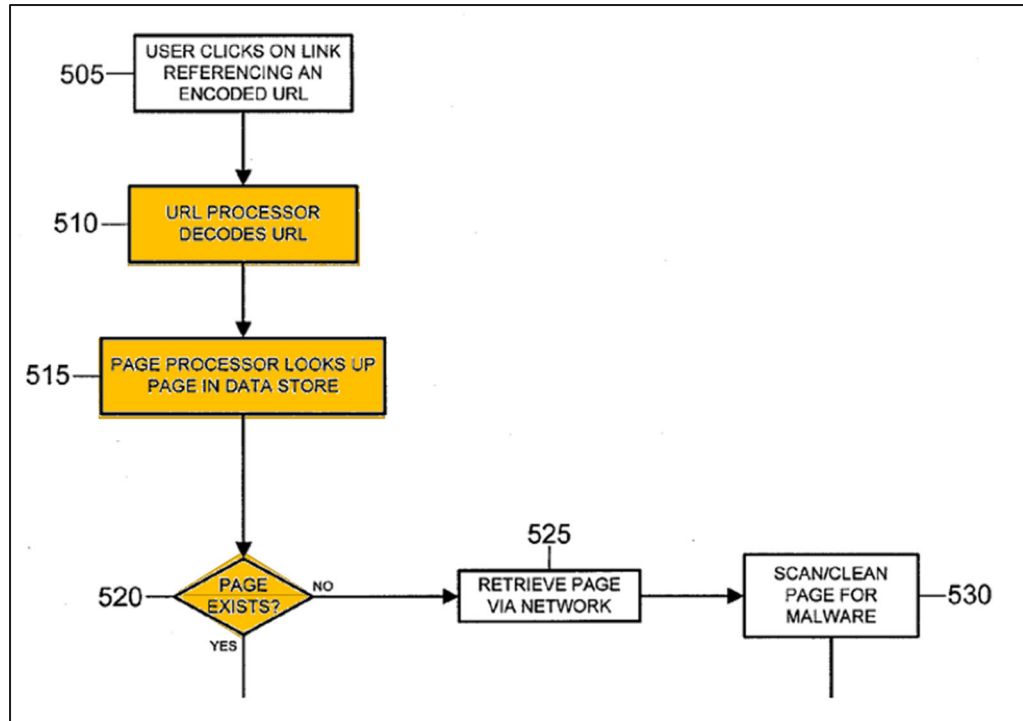
results,” where the HTML code “may utilize invisible frame, or an HTML iframe element, to implement the safe browsing system.” Ex.1005, [0087]; Ex.1002, ¶216. A POSITA would understand McAfee’s system performs HTML and URL analysis. Ex.1002, ¶216.

A POSITA would be motivated to combine the “*URL analysis*” and “*a script/HTML analysis*” of McAfee with the other types of “*analysis to identify a malware threat[s]*” in Devarajan at least for the reasons described in Section VII.C.1.

4. Claim 4

Claim 4 is rendered obvious by Devarajan in view of McAfee. *See* Section VII.A.4; Ex.1002, ¶¶ 218-20.

Figure 5 of McAfee depicts a URL processor. The process begins when a user “click[s] on a link representing a page of the content provider.” Ex.1005, [0073]; Ex.1002, ¶219. At the next block, 510, the “service provider server 240 may receive the request and may communicate [it] to the URL processor,” which decodes the URL. Ex.1005, [0073]; Ex.1002, ¶219. Then, the page processor looks “up the page referenced by the actual URL in the page data store 320” to determine whether the request had been made before and if “a clean copy of the page is cached in the page data store” (Ex.1005, [0074])—*e.g.*, “*determining...whether a request for the web content from the web browser application is a first request.*” Ex.1002, ¶219.



Ex.1005, Fig. 5 (annotated).

Moreover, a POSITA would be motivated to add the teachings from McAfee regarding caching and a first request to the system of Devarajan, as it would improve the response time for the system, since it would replace a full analysis and scrubbing with a simple memory lookup. Ex.1002, ¶220. In addition, this would be a simple modification, since caching is a well-known technique; hence, a POSITA would have an expectation of success and would be motivated to try this change. *Id.*, ¶¶220.

5. Claim 5

Claim 5 is rendered obvious by Devarajan in view of McAfee. *See* Ex.1002, ¶¶221-23.

A POSITA would understand web content that is encrypted, chunked, and/or compressed poses a challenge with respect to the detection of malware in the content, since such content is not immediately viewable. *Id.*, ¶222. For example, a POSITA would understand when a long URL, such as https://en.wikipedia.org/wiki/URL_shortening, is replaced by a shortened URL, such as <https://w.wiki/U>, the compressed, shortened URL may be hiding a URL that is associated with malware. *Id.* Thus, a POSITA would be motivated to look to a reference such as McAfee for guidance regarding dealing with compressed URLs, e.g., to find guidance on decompressing such URLs so as to apply the Devarajan URL analysis method to the decompressed web content. *Id.*

McAfee teaches, “[t]he URL processor 314 may be operative to encode and decode the URLs on a page.” When a user clicks on a link for a webpage, McAfee’s system communicates the “encoded URL to the URL processor,” which then “decode[s] the URL by looking up the actual URL mapped to the encoded URL in the URL mapping data store 330.” Ex.1005, [0073]; *see also id.*, [0063] (“The URL mapping data store 330 may be a data structure, such as a database, a hash table, or generally any data structure capable of mapping an encoded URL to a URL.”); Ex.1002, ¶223. A POSITA would understand this decoding as decompressing the web content prior to the request being processed by the system. Ex.1002, ¶223. A POSITA would understand that this decompression of the URL is easily integrated

into Devarajan. *Id.* For example, Devarajan’s analysis can proceed as disclosed in Devarajan and McAfee’s URL analysis can be part of Devarajan’s content analysis (Fig. 10, step 1090) without further modifications being required. *Id.* A POSITA would have an expectation of success in adding the URL decoding processor of McAfee to the Devarajan system, and such a modification would add URL decompressing to the Devarajan system along with the benefits of additional malware detection that would be provided. *Id.*, ¶¶221-23.

6. Claim 6

Claim 6 is rendered obvious by Devarajan in view of McAfee. *See* Section VII.A.5; Ex.1002, ¶¶224-26.

As described in Section VII.C.3, a POSITA would understand that McAfee’s analysis “*to identify the malware threat within the web content*”—e.g., performing HTML and URL analysis—is “*protocol-level analysis.*” Ex.1002, ¶¶224-26.

7. Claim 7

Claim 7 is rendered obvious by Devarajan in view of McAfee. *See* Section VII.A.6; Ex.1002, ¶¶227-30.

McAfee describes a system with a service provider 240, as depicted in Figure 3. Ex.1005, Fig. 3; Ex.1002, ¶228. McAfee teaches, “[t]he service provider server 240 may exist on one machine or may be running in a distributed configuration on one or more machines.” Ex.1005, [0055]; Ex.1002, ¶229. A POSITA would

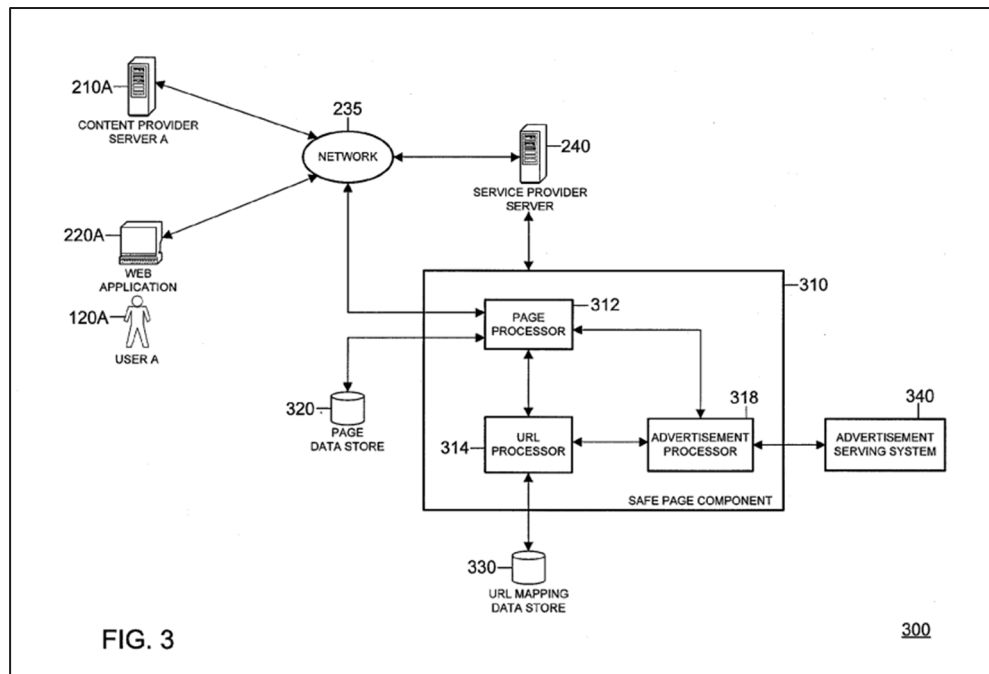
understand implementing the service provider in a distributed computing, i.e., “cloud” environment, would provide additional benefits, such as being more tolerant to computer or network failures (since the computations are distributed across multiple computers). Ex.1002, ¶229. Thus, a POSITA would be motivated to look to a reference, such as McAfee for guidance. *Id.* Given McAfee’s teaching, a POSITA would be motivated to implement the system of Devarajan in a distributed (i.e., cloud computing) environment, where, e.g., some of the components of the server 108 (such as the malicious address database) are distributed and mirrored across multiple computers in the cloud. *Id.* Further, this teaching is consistent with the teaching in Devarajan that “[t]he instructions 110 may, as non-limiting examples, comprise software and/or scripts stored in the computer-readable media 109 that may be...accessible via the network 101, perhaps via a grid or **cloud-computing environment.**” Ex.1004, [0054]; Ex.1002, ¶229. Further, given the system in Devarajan is already performing an aggregation of analyses on web content, a POSITA would understand this modification to perform the method of Devarajan in cloud to be a simple modification that would have a likelihood of success. Ex.1002, ¶229. For example, Devarajan already discloses its analysis of web content includes an aggregation for identifying “client-directed malware (e.g., a virus, a worm, a trojan horse, a rootkit, a backdoor, a spyware, a keystroke logger, a phishing application, a script injection, or an iFrame injection).” Ex.1004, [0102]; Ex.1002,

¶229. A POSITA would understand distributing this aggregated analysis to the cloud would improve robustness and availability of the system; hence, a POSITA would be motivated to make this modification of Devarajan in view of McAfee and there would be an expectation of success. Ex.1002, ¶229.

8. Claim 8

Claim 8 is rendered obvious by Devarajan in view of McAfee. Ex.1002, ¶¶231-34.

McAfee's Figure 3 illustrates the server-side components of its system. The system comprises a "a page data store 320, which may cache clean versions of web pages." Ex.1005, [0061]; Ex.1002, ¶232. A POSITA would recognize the benefits of such caching, as it allows the system to reuse previously scrubbed web content rather than redoing all the work to scrub a web page that has previously been analyzed and cleaned. Ex.1002, ¶233. For example, McAfee teaches, "[t]he pages stored, or cached, in the page data stores 320 may be immediately served to the user A 120A, and may be utilized to improve the overall performance of the system 100." Ex.1005, [0079]; Ex.1002, ¶233.



Ex.1005, Fig. 3.

A POSITA would be aware of the benefits of caching (e.g., in reducing latency) and would be motivated to look to a reference such as McAfee for guidance. Ex.1002, ¶233. A POSITA, therefore, would be motivated to incorporate this simple teaching regarding caching into the system of Devarajan, and would have an expectation of success, given the prevalence of caching in computing generally. *Id.*, ¶234. This step could easily take place during Devarajan's initial analysis at Step 400 in Figure 10 where the system determines whether the requested web content comprises malware.

9. Claim 9

- a. [9P] A content analysis and malware prevention system comprising:

Element [9P] is rendered obvious by Devarajan in view of McAfee. *See* Section VII.C.1.a. Ex.1002, ¶235.

- b. [9A] at least one processor; and

Element [9A] is rendered obvious by Devarajan in view of McAfee. *See* Section VII.A.7.b. McAfee explains its system “may be implemented with a processor such as a personal computer, personal digital assistant, mobile phone, or any other machine capable of implementing a web application.” Ex.1005, [0034]; Ex.1002, ¶236.

- c. [9B] memory encoding computer executable instructions that, when executed by the at least one processor, perform a method comprising:

Element [9B] is rendered obvious by Devarajan in view of McAfee. *See* Section VII.A.7.c. Further, McAfee explains its system may include a “programmed processor 1302 executing the instructions 1324 stored in the memory 1304.” Ex.1005, [0095]; Ex.1002, ¶237.

- d. [9C] intercepting a request from a web browser;

Element [9C] is rendered obvious by Devarajan in view of McAfee. *See* Section VII.C.1.b. Ex.1002, ¶238.

- e. [9D] determining whether the request is associated with known malicious content;

Element [9D] is rendered obvious by Devarajan in view of McAfee. *See* Section VII.C.1.c. Ex.1002, ¶239.

- f. [9E] when the request is associated with known malicious content, blocking, by a protection agent, the request;

Element [9E] is rendered obvious by Devarajan in view of McAfee. *See* Section VII.C.1.d. Ex.1002, ¶240.

- g. [9F] when the request is not associated with known malicious content: sending the request to one or more web servers;

Element [9F] is rendered obvious by Devarajan in view of McAfee. *See* Section VII.C.1.e. Ex.1002, ¶241.

- h. [9G] receiving web content from one or more web servers, wherein web content comprises data for assembling a web page;

Element [9G] is rendered obvious by Devarajan in view of McAfee. *See* Section VII.C.1.f. Ex.1002, ¶242.

- i. [9H] identifying a malware threat within the web content;

Element [9H] is rendered obvious by Devarajan in view of McAfee. *See* Section VII.C.1.g. Ex.1002, ¶243.

- j. [9I] in response to identifying the malware threat, modifying the web content, wherein modifying the web

content comprises modifying the web at protocol level to
remove the malware threat from the web content;

Element [9I] is rendered obvious by Devarajan in view of McAfee. *See*
Section VII.C.1.h. Ex.1002, ¶244.

- k. [9J] assembling a modified version of the web page,
wherein the modified web page is assembled using the
modified web content, and wherein the modified version
of the web page does not comprise the malware threat; and

Element [9J] is rendered obvious by Devarajan in view of McAfee. *See*
Section VII.C.1.i. Ex.1002, ¶245.

- l. [9K] providing the modified version of the web page to a
web browser application on the remote computer for
rendering and display.

Element [9K] is rendered obvious by Devarajan in view of McAfee. *See*
Section VII.C.1.j. Ex.1002, ¶246.

10. Claim 10

Claim 10 is rendered obvious by Devarajan in view of McAfee. *See* Section
VII.C.2. Ex.1002, ¶247.

11. Claim 11

Claim 11 is rendered obvious by Devarajan in view of McAfee. *See* Section
VII.C.3. Ex.1002, ¶248.

12. Claim 12

Claim 12 is rendered obvious by Devarajan in view of McAfee. *See* Section
VII.C.4. Ex.1002, ¶249.

13. Claim 13

Claim 13 is rendered obvious by Devarajan in view of McAfee. *See* Section VII.C.5. Ex.1002, ¶250.

14. Claim 14

Claim 14 is rendered obvious by Devarajan in view of McAfee. *See* Section VII.C.6. Ex.1002, ¶251.

15. Claim 15

Claim 15 is rendered obvious by Devarajan in view of McAfee. *See* Section VII.C.8. Ex.1002, ¶252.

16. Claim 16

Claim 16 is rendered obvious by Devarajan in view of McAfee. *See* Section VII.A.12; Ex.1002, ¶253. McAfee's system comprises a "URL processor" that is "operative to encode and decode the URLs on a page." Ex.1005, [0063]; Ex.1002, ¶253. These URLs may be "encoded by the URL processor 314 to ensure each of the URLs redirects the user A120A through the service provider server 240," to analyze whether the URL comprises malware. Ex.1005, [0063]; Ex.1002, ¶253.

17. Claim 17

Claim 17 is rendered obvious by Devarajan in view of McAfee. *See* Section VII.C.16. Ex.1002, ¶254.

18. Claim 18

Claim 18 is rendered obvious by Devarajan in view of McAfee. *See* Section VII.A.14; Ex.1002, ¶¶255-59.

McAfee also teaches that if no malware is found at block 615, then block 637 may further process the page by “the page processor.” Ex.1005, [0082]; Ex.1002, ¶257. The page processor “may retrieve the page from the content provider server,” which constitutes sending a request to the server in order to retrieve the page’s content. Ex.1005, [0077]; *see also id.*, [0005]; [0082] Ex.1002, ¶258. A POSITA would understand such requests are sent iteratively until all of the web content is received. Ex.1002, ¶258.

19. Claim 19

Claim 19 is rendered obvious by Devarajan in view of McAfee. *See* Section VII.A.15; Ex.1002, ¶260. McAfee further discloses decoding URLs on a page prior to analyzing the content linked to the URLs for malware. Ex.1005, [0063].

20. Claim 20

Claim 20 is rendered obvious by Devarajan in view of McAfee. *See* Section VII.C.18. Ex.1002, ¶261.

VIII. SECONDARY CONSIDERATIONS

There is no evidence of secondary considerations that would support a determination of non-obviousness of the claimed subject matter in the ’928 patent.

IX. CERTIFICATION OF GROUNDS FOR STANDING UNDER 37 C.F.R. §42.104

Petitioner certifies pursuant to Rule 42.104(a) that the '928 Patent is available for IPR and that Petitioner is not barred or estopped from requesting an IPR of the Challenged Claim on the grounds identified in this Petition.

X. MANDATORY NOTICES

A. Real Party-in-Interest Under 37 C.F.R. § 42.8(b)(1)

The real party-in-interest is Forcepoint LLC.

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

To the best of Petitioner's knowledge, the '928 Patent is, or has been, involved in the following district court litigations, appeals, and Board proceedings: *Webroot, Inc. et al v. Forcepoint LLC*, No. 6:22-cv-342 (W.D. Tex.) (filed March 31, 2022) (consolidated with lead case No. 6:22-cv-243).

C. Lead and Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3)

Lead Counsel	Back-Up Counsel
Kat Li Reg. No. 64,857 kat.li@kirkland.com <u>Postal and Hand-Delivery Address:</u> KIRKLAND & ELLIS LLP 401 Congress Avenue Austin, TX 78701 Telephone: (512) 678-9100 Facsimile: (512) 678-9101	Akshay S. Deoras (<i>pro hac vice</i> to be filed) akshay.deoras@kirkland.com <u>Postal and Hand-Delivery Address:</u> KIRKLAND & ELLIS LLP 555 California Street San Francisco, CA 94104 Telephone: (415) 439-1400 Facsimile: (415) 439-1500 Nadia E. Haghghatian (<i>pro hac vice</i> to be filed) nadia.haghghatian@kirkland.com <u>Postal and Hand-Delivery Address:</u> KIRKLAND & ELLIS LLP 401 Congress Avenue Austin, TX 78701 Telephone: (512) 678-9100 Facsimile: (512) 678-9101

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

Petitioner concurrently submits a Power of Attorney under 37 C.F.R. §42.10(b), and consents to electronic service to Forcepoint_Webroot_IPR@kirkland.com.

XI. PAYMENT OF FEES UNDER 37 C.F.R. §42.103

The undersigned authorizes the Office to charge the fee set forth in 37 C.F.R. §42.15(a)(1) for this Petition to Deposit Account No. 506092. Review of twenty claims is requested. The undersigned further authorizes payment for any additional

fees that may be due in connection with this Petition to be charged to the above-referenced deposit account.

XII. CONCLUSION

Petitioner requests that the Board institute *inter partes* review and cancel claims 1-20 of the '928 Patent.

Respectfully submitted,

Dated: March 31, 2023

By: /Kat Li/
Kat Li (Reg. No. 64,857)
kat.li@kirkland.com
Nadia E. Haghighatian
nadia.haghighatian@kirkland.com
KIRKLAND & ELLIS LLP
401 Congress Avenue
Austin, TX 78701
Telephone: (512) 678-9100
Facsimile: (512) 678-9101

Akshay S. Deoras
akshay.deoras@kirkland.com
KIRKLAND & ELLIS LLP
555 California Street
San Francisco, CA 94104
Telephone: (415) 439-1400
Facsimile: (415) 439-1500

**CERTIFICATE OF COMPLIANCE WITH
TYPE-VOLUME LIMITATION PURSUANT TO 37 C.F.R. § 42.24**

This brief complies with the type-volume limitation of 37 C.F.R. § 42.24(a)(1)(i).

The brief contains 13,953 words, excluding the parts of the brief exempted by 37 C.F.R. § 42.24(a).

The brief has been prepared in a proportionally spaced typeface using Microsoft Word in a 14-point Times New Roman font.

Dated: March 31, 2023

/Kat Li/
Kat Li (Reg. No. 64,857)

CERTIFICATE OF SERVICE

In compliance with 37 C.F.R. §§ 42.105, 42.6(e), the undersigned hereby certifies that a copy of the foregoing document was sent on March 31, 2023, via Federal Express® directed to POs at the correspondence address of record:

Sprinkle IP Law Group
1301 W. 25th Street, Suite 400
Austin, TX 78705

A courtesy copy was also served by electronic mail on POs' counsel of record for the following related matter:

Webroot, Inc., et al. v. Forcepoint LLC, No. 6:22-cv-342 (W.D. Tex.)

JEFFREY D. MILLS
jmills@kslaw.com
KING & SPALDING LLP
500 W. 2nd Street, Suite 1800
Austin, TX 78701
Telephone: (512) 457-2027
Telecopier: (202) 626-3737

CHRISTOPHER C. CAMPBELL
ccampbell@kslaw.com
PATRICK M. LAFFERTY
plafferty@kslaw.com
KING & SPALDING LLP
1700 Pennsylvania Avenue, NW
Washington, DC 20006
Telephone: (202) 737-0500
Telecopier: (202) 626-3737

BRIAN J. EUTERMOSER
beutermoser@kslaw.com
BRITTON F. DAVIS
bfdavis@kslaw.com
KING & SPALDING LLP
1401 Lawrence Street, Suite 1900
Denver, Colorado 80202
Telephone: (720) 535-2300
Telecopier: (720) 535-2400

MARK D. SIEGMUND
mark@cjsjlaw.com
**CHERRY JOHNSON SIEGMUND
JAMES, PLLC**
400 Austin Ave., 9th Floor
Waco, Texas 76701
Telephone: (254) 732-2242
Telecopier: (866) 627-3509

/Kat Li/

Kat Li (Reg. No. 64,857)