

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

FORCEPOINT LLC,
Petitioner,

v.

OPEN TEXT INC.,
Patent Owner.

IPR2023-00786
Patent 10,025,928 B2

Before JUSTIN T. ARBES, BARBARA A. PARVIS, and
MICHELLE N. WORMMEESTER, *Administrative Patent Judges*.

ARBES, *Administrative Patent Judge*.

DECISION
Denying Institution of *Inter Partes* Review
35 U.S.C. § 314

I. INTRODUCTION

A. *Background and Summary*

Petitioner Forcepoint LLC filed a Petition (Paper 2, “Pet.”) requesting *inter partes* review of claims 1–20 of U.S. Patent No. 10,025,928 B2 (Ex. 1001, “the ’928 patent”) pursuant to 35 U.S.C. § 311(a). Patent Owner Open Text Inc. filed a Preliminary Response (Paper 6, “Prelim. Resp.”)

pursuant to 35 U.S.C. § 313. The parties also filed a Notice of Stipulation (Paper 12) and Reply (Paper 14), pursuant to our authorization (Paper 11).

Pursuant to 35 U.S.C. § 314(a), the Director may not authorize an *inter partes* review unless the information in the petition and preliminary response “shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” *See* 37 C.F.R. § 42.4(a) (“The Board institutes the trial on behalf of the Director.”). For the reasons that follow, we do not institute an *inter partes* review.

B. Related Matters

The parties indicate that the ’928 patent is the subject of *Webroot, Inc. v. Forcepoint LLC*, No. 6:22-cv-00342 (W.D. Tex.), which has been consolidated with *Webroot, Inc. v. AO Kaspersky Lab*, No. 6:22-cv-00243 (W.D. Tex.) (“the district court case”). *See* Paper 4, 1; Paper 10, 1. Patent Owner also cites various other *inter partes* review proceedings involving unrelated patents at issue in the district court case. Paper 4, 1–2. Petitioner filed a petition challenging an unrelated patent in Case IPR2023-00784.

C. The ’928 Patent

The ’928 patent discloses systems and methods for preventing “malware,” i.e., “program[s] that [are] malicious in some way and/or collect[] and/or report[] information about a person or an organization,” or “other undesirable or unwanted applications and/or instructions.” Ex. 1001, col. 1, ll. 16–39. The ’928 patent states that prior art approaches to malware prevention “just make high-level modifications to the content after the content has been rendered and displayed” by a browser engine, which was

“problematic because it allows the browser engine to potentially execute malicious scripts or perform malicious actions while it is parsing and rendering the code” and also inefficient because “the annotation process must account for the rendering differences (e.g., differences in how and where content is displayed) that different browsers . . . exhibit.” *Id.* at col. 2, ll. 16–23, col. 7, ll. 8–20. The ’928 patent purports to overcome those problems by using a “protocol-level analysis approach.” *Id.* at col. 2, ll. 10–23.

The ’928 patent describes an exemplary computing environment comprising remote computer 105 with protection agent 116 and browser engine 118, malware source 106 that is the “source of malware that ends up or is strategically placed at the web server 108,” and security center 102, which “obtain[s] information about malware threats” and acts as “a resource for the remote computer 105.” *Id.* at col. 2, l. 66–col. 3, l. 59, Fig. 1. The ’928 patent describes the general operation of the system as follows:

In operation, the protection agent 116 in this embodiment 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. And then the protection agent 116 analyzes and/or modifies the completed data before the browser engine 118 has access to it. After performing all of its processing, removing, and/or adding any code as needed, 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. This allows the protection agent 116 to have full control over what the browser engine 118 “sees,” providing means to remove any exploits, malware, and other threats dynamically. This also enables the protection agent 116 to add content into the

browser stream at the HTML level. Stated another way, the protection agent 116 caches web content requested by a browser, analyzes and/or modifies the retrieved web content, and provides a clean or sanitized version of the web content, free of malware, to the browser.

Id. at col. 4, ll. 6–29.

Figure 3 of the '928 patent is reproduced below.

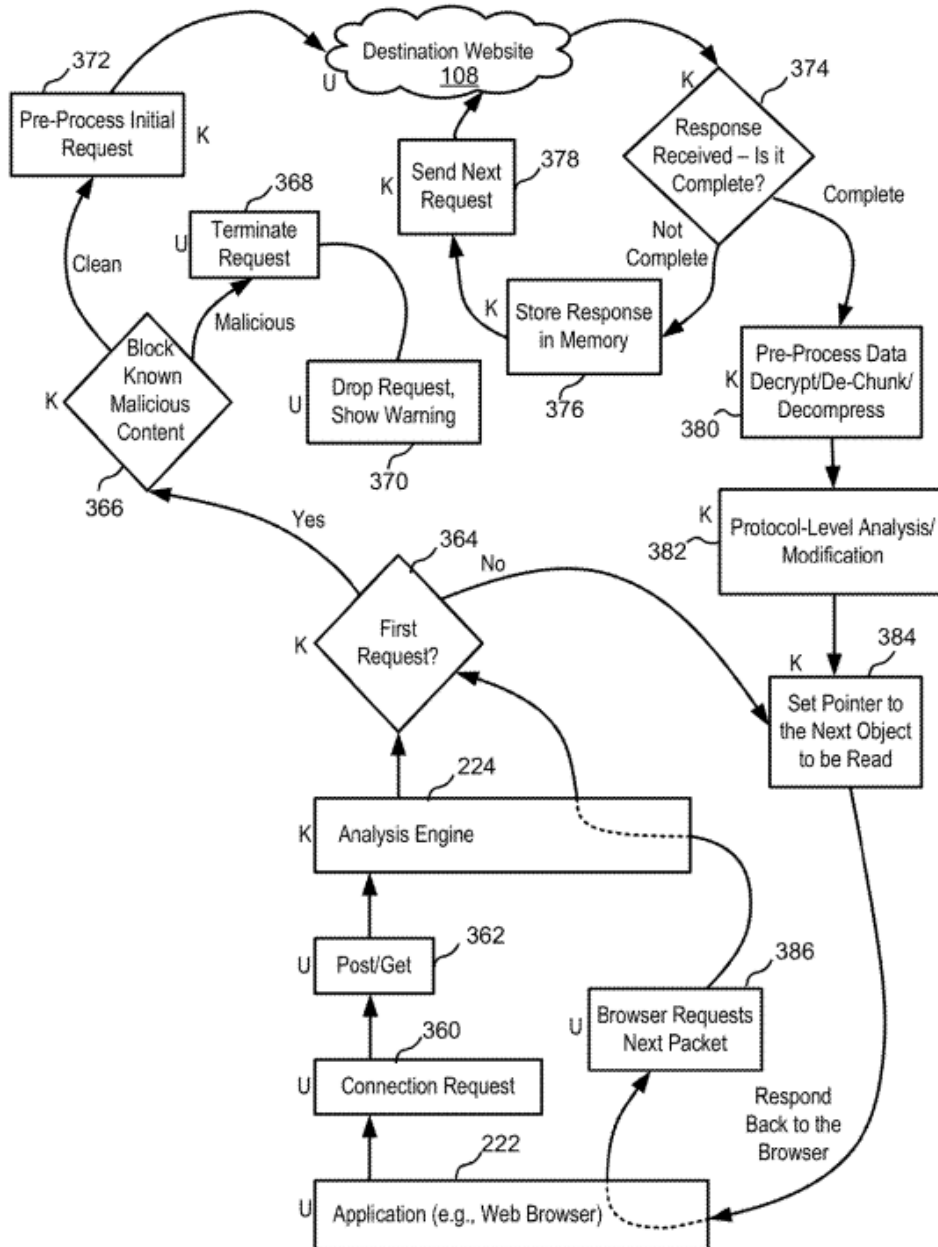


FIG. 3

Figure 3 is a flow chart of operations performed by the remote computer in the disclosed method. *Id.* at col. 2, ll. 59–60. The '928 patent discloses:

[W]hen a user initiates a request via the application 222 (e.g., web browser or other application that requests web content), 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. at col. 5, ll. 52–62. “[A]nalysis engine 224 accesses the security center 102 via the Internet and the security center 102 is utilized to facilitate whether the request is a request for known malicious content (e.g., the [Uniform Resource Locator (URL)] of the request may be compared to a black list of URLs),” or, alternatively, analysis engine 224 performs “some malware checking functionality locally.” *Id.* at col. 5, l. 63–col. 6, l. 2. If the request is not blocked, the remote computer “iteratively sends requests and receives content (Blocks 374, 376, 378) from the website 108 until the requested content has been complete[d],” processes the data (Block 380), analyzes and modifies it (if necessary) “at the protocol level (e.g., at the HTML level)” (Block 382), and finally returns the resulting web page to the browser. *Id.* at col. 6, ll. 17–col. 7, l. 6. Thus, “from the browser engine’s . . . perspective, the browser engine . . . is obtaining the content from the webserver . . . directly.” *Id.* at col. 8, ll. 1–3.

D. Illustrative Claim

Challenged claims 1 and 9 of the '928 patent are independent. Claims 2–8, 16, 18, and 19 depend, directly or indirectly, from claim 1, and

claims 10–15, 17, and 20 depend from claim 9. Claim 1 recites (with letter designations used in the Petition to refer to the various limitations):

1. A content analysis and malware prevention method comprising:

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

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

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

[1D] when the request is not associated with known malicious content:

 sending, by the protection agent, the request to one or more web servers;

 [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;

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

 [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;

 [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

 [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.

E. Evidence

Petitioner relies on the following prior art:

U.S. Patent No. 9,654,495 B2, filed Feb. 28, 2007, issued May 16, 2017 (Ex. 1008, “Hubbard”);

U.S. Patent Application Publication No. 2013/0007870 A1, filed Sept. 29, 2011, published Jan. 3, 2013 (Ex. 1004, “Devarajan”); and

U.S. Patent Application Publication No. 2009/0070873 A1, published Mar. 12, 2009 (Ex. 1005, “McAfee”).

F. Prior Art and Asserted Grounds

Petitioner asserts that claims 1–20 of the ’928 patent are unpatentable on the following grounds:

Claims Challenged	35 U.S.C. §	References/Basis
1–4, 6, 7, 9–12, 14, 16–20	103(a) ¹	Devarajan
1–4, 6, 7, 9–12, 14, 16–20	103(a)	Devarajan, Hubbard
1–20	103(a)	Devarajan, McAfee

II. ANALYSIS

A. Legal Standards

A claim is unpatentable for obviousness if, to one of ordinary skill in the pertinent art, “the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would

¹ The Leahy-Smith America Invents Act, Pub. L. No. 112-29, 125 Stat. 284 (2011) (“AIA”), amended 35 U.S.C. § 103. Because the challenged claims of the ’928 patent have an effective filing date before the effective date of the applicable AIA amendment, we refer to the pre-AIA version of 35 U.S.C. § 103. *See* Pet. 2 & n.1; Ex. 1006 ¶ 79.

have been obvious at the time the invention was made.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007) (quoting 35 U.S.C. § 103(a) (2006)). The question of obviousness is resolved on the basis of underlying factual determinations, including “the scope and content of the prior art”; “differences between the prior art and the claims at issue”; and “the level of ordinary skill in the pertinent art.” *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966). Additionally, secondary considerations, such as “commercial success, long felt but unsolved needs, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented. As indicia of obviousness or nonobviousness, these inquiries may have relevancy.” *Id.* When conducting an obviousness analysis, we consider a prior art reference “not only for what it expressly teaches, but also for what it fairly suggests.” *Bradium Techs. LLC v. Iancu*, 923 F.3d 1032, 1049 (Fed. Cir. 2019) (citation omitted).

A patent claim “is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art.” *KSR*, 550 U.S. at 418. An obviousness determination requires finding “both ‘that a skilled artisan would have been motivated to combine the teachings of the prior art references to achieve the claimed invention, and that the skilled artisan would have had a reasonable expectation of success in doing so.’” *Intelligent Bio-Sys., Inc. v. Illumina Cambridge Ltd.*, 821 F.3d 1359, 1367–68 (Fed. Cir. 2016) (citation omitted); *see KSR*, 550 U.S. at 418 (for an obviousness analysis, “it can be important to identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does”). Also, “[t]hough less common, in appropriate circumstances, a patent can be obvious in light of a single prior art reference if it would have been obvious

to modify that reference to arrive at the patented invention.” *Arendi S.A.R.L. v. Apple Inc.*, 832 F.3d 1355, 1361 (Fed. Cir. 2016).

“Although the *KSR* test is flexible, the Board ‘must still be careful not to allow hindsight reconstruction of references . . . without any explanation as to *how* or *why* the references would be combined to produce the claimed invention.” *TriVascular, Inc. v. Samuels*, 812 F.3d 1056, 1066 (Fed. Cir. 2016) (citation omitted). Further, an assertion of obviousness “cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” *KSR*, 550 U.S. at 418 (quoting *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)); accord *In re NuVasive, Inc.*, 842 F.3d 1376, 1383 (Fed. Cir. 2016) (stating that “conclusory statements” amount to an “insufficient articulation[] of motivation to combine”; “instead, the finding must be supported by a ‘reasoned explanation’” (citation omitted)); *In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016) (“To satisfy its burden of proving obviousness, a petitioner cannot employ mere conclusory statements. The petitioner must instead articulate specific reasoning, based on evidence of record, to support the legal conclusion of obviousness.”).

B. Level of Ordinary Skill in the Art

Petitioner argues that at the time of the ’928 patent (October 2011), a person of ordinary skill in the art would have had “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,” where “[a]dditional education could substitute for professional experience and vice versa.” Pet. 8 (citing Ex. 1002 ¶¶ 26–29). Patent

Owner argues that a person of ordinary skill in the art instead would have had “a bachelor’s degree in an accredited program of electrical engineering, computer engineering, or computer science, or a similar discipline, and have [had] 2–3 years of practical work experience in the general fields of electrical engineering, computer science, networking, communications, and/or device and network security,” where “[m]ore advanced degrees and/or training in a related discipline can compensate for shorter work experience.” Prelim. Resp. 28; *see* Ex. 2001 ¶¶ 49–51. The parties’ proposed definitions have only minor differences (e.g., at least two years of work experience versus 2–3 years). For purposes of this Decision, we adopt Petitioner’s proposed definition of the level of ordinary skill in the art, except that we eliminate the term “at least” to eliminate vagueness as to the amount of education and practical experience. We note, however, that our conclusions herein would be the same regardless of which party’s definition is used.

C. Claim Interpretation

We interpret the challenged claims

using the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. 282(b), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.

37 C.F.R. § 42.100(b). Petitioner does not propose interpretations of any claim terms. Pet. 8. Patent Owner states that one claim interpretation issue must be resolved due to how Petitioner applies one of the asserted prior art references, Devarajan. Prelim. Resp. 24. Specifically, claims 1 and 9 recite

“determining . . . whether the request is associated with known malicious content” (limitations [1B] and [9D]). According to Patent Owner, Petitioner reads Devarajan as teaching the limitation because it teaches “determining whether a request for web content from a client is itself malware directed to the server,” but the claimed determination instead “is whether the ‘request’ is *associated with* known malicious *content* and not whether the request, itself, is malicious.” *Id.* (citing Pet. 22–23). Patent Owner argues that the limitation should be given its “plain and ordinary meaning, which does not include determining whether the request itself is malicious.” *Id.* Based on our review of the record before us, we agree with Patent Owner.

First, the claims distinguish between a “request” for content and potentially malicious “content” that is requested. Limitations [1A]–[1C], for example, recite intercepting a “request” from a web browser, determining whether the “request” is associated with known malicious “content,” and blocking the “request” when it is so associated. Limitations [1D]–[1I] then similarly recite, when the “request” is not so associated, sending the “request” to one or more web servers, receiving web “content” “in response to [that] request,” processing the received web “content” to create a modified version of a web page, and providing that modified version of the web page to the web browser application for rendering and display. Thus, the “request” in claim 1 is a request for “content,” which is what the one or more web servers return. Dependent claims 4 and 12 likewise recite determining “whether a request for the web content from the web browser application is a first request.” Patent Owner argues, and we agree, that Petitioner’s reading of the claim language “obfuscates the difference between (i) determining whether a request from a web browser *is* server-directed malware (as disclosed in Step 400 of Devarajan) and

(ii) determining whether it is ‘*associated with known malicious content*’ (as recited in Element [1B]).” *See id.* at 41 (citing Ex. 2001 ¶ 75).

Second, the Specification of the ’928 patent consistently indicates that malicious “content” is something other than the “request” itself. For example, the Specification describes determining whether a request, such as the URL of a website, is associated with content known to be malicious (e.g., the web page(s) at that URL) by determining whether the URL is on a black list of URLs. *See* Ex. 1001, col. 5, ll. 52–62 (“a determination is made as to whether the request is associated with known, malicious content (Block 366)”), 63–67 (“In some embodiments, the analysis engine 224 accesses the security center 102 via the Internet and the security center 102 is utilized to facilitate *whether the request is a request for known malicious content (e.g., the URL of the request may be compared to a black list of URLs).*” (emphasis added)), col. 6, ll. 17–23 (“the initial request 360 by the application (e.g., browser) is intercepted and *if the request does not appear to be a request for malicious content*, the content acquisition component 226 iteratively sends *requests* and receives *content* . . . from the website 108” (emphasis added)). Other portions of the Specification likewise refer to content (which may be malicious) as distinct from a request for content. *See, e.g., id.* at col. 2, ll. 24–29 (“the protection module receives content from a web server”), col. 3, ll. 21–24 (“malware source 106 generally represents a source of malware that ends up or is strategically placed at the web server 108, which may or may not be suspected of hosting malware”), col. 4, ll. 56–61 (“protection agent 116 receives content from the web server 108”), col. 5, ll. 47–51 (“agent processes 220 enable all content that is requested by an application 222 (e.g., a browser) and received by the browser engine 218 to be intercepted”), 52–61 (“a user initiates a request via

the application 222 (e.g., web browser or other application that requests web content)"). Petitioner does not point to, and we do not find, anything in the Specification describing determining whether a request itself is malicious. The Specification thus supports Patent Owner's view that determining whether a request is "associated with known malicious content," as claimed, is different from determining whether the request itself is malicious. *See* Prelim. Resp. 26–27, 40; *Trustees of Columbia Univ. v. Symantec Corp.*, 811 F.3d 1359, 1368 (Fed. Cir. 2016) (construing the phrase "probabilistic model of normal computer system usage" to mean being "built using only attack-free data" where "[t]he specification consistently describes an implementation" involving attack-free data and "[n]othing in the specification describes any embodiment which uses attack data to build the model").

Further, as Patent Owner notes, the difference is significant because the determinations may return different results:

For example, consider a scenario in which the request is a legitimate URL (*i.e.*, the URL has not been hacked or modified to attack a server) that links to a known, malicious web page. The approach claimed in the '928 patent will compare this URL to a blacklist of URLs associated with known malicious web pages, and when it determines that the URL request is associated with known malicious content, it will block the request. In contrast, Devarajan's method/system will focus on the request itself to determine if it is malicious and, because the request does not contain a pattern or signature that indicates that the URL has been hacked or tampered with, e.g., no markers of a request containing a SQL injection or DDOS attack, determine that it is legitimate. Devarajan will, therefore, determine that the request, itself, is not malicious and allow the request to proceed.

Prelim. Resp. 27–28 (citations omitted); *see* Ex. 2001 ¶¶ 44–45.

Accordingly, we interpret “determining . . . whether the request is associated with known malicious content” to not encompass determining whether the request itself is malicious, and conclude that no further interpretation is necessary to determine whether to institute an *inter partes* review in this proceeding. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (“Because we need only construe terms ‘that are in controversy, and *only to the extent necessary* to resolve the controversy,’ we need not construe [a particular claim limitation] where the construction is not ‘material to the . . . dispute.’” (citation omitted) (emphasis added)).

D. Obviousness Ground Based on Devarajan (Claims 1–4, 6, 7, 9–12, 14, and 16–20)

Petitioner contends that claims 1–4, 6, 7, 9–12, 14, and 16–20 are unpatentable over Devarajan under 35 U.S.C. § 103(a), citing the testimony of Michael T. Goodrich, Ph.D., as support. Pet. 13–52 (citing Ex. 1002). Patent Owner responds, citing the testimony of Alessandro Orso, Ph.D., as support. Prelim. Resp. 35–44 (citing Ex. 2001). We are not persuaded that Petitioner has established a reasonable likelihood of prevailing on this asserted ground.

1. Devarajan

Devarajan discloses a “bi-directional network traffic malware detection and removal system.” Ex. 1004, code (57).

Figure 9 of Devarajan is reproduced below.

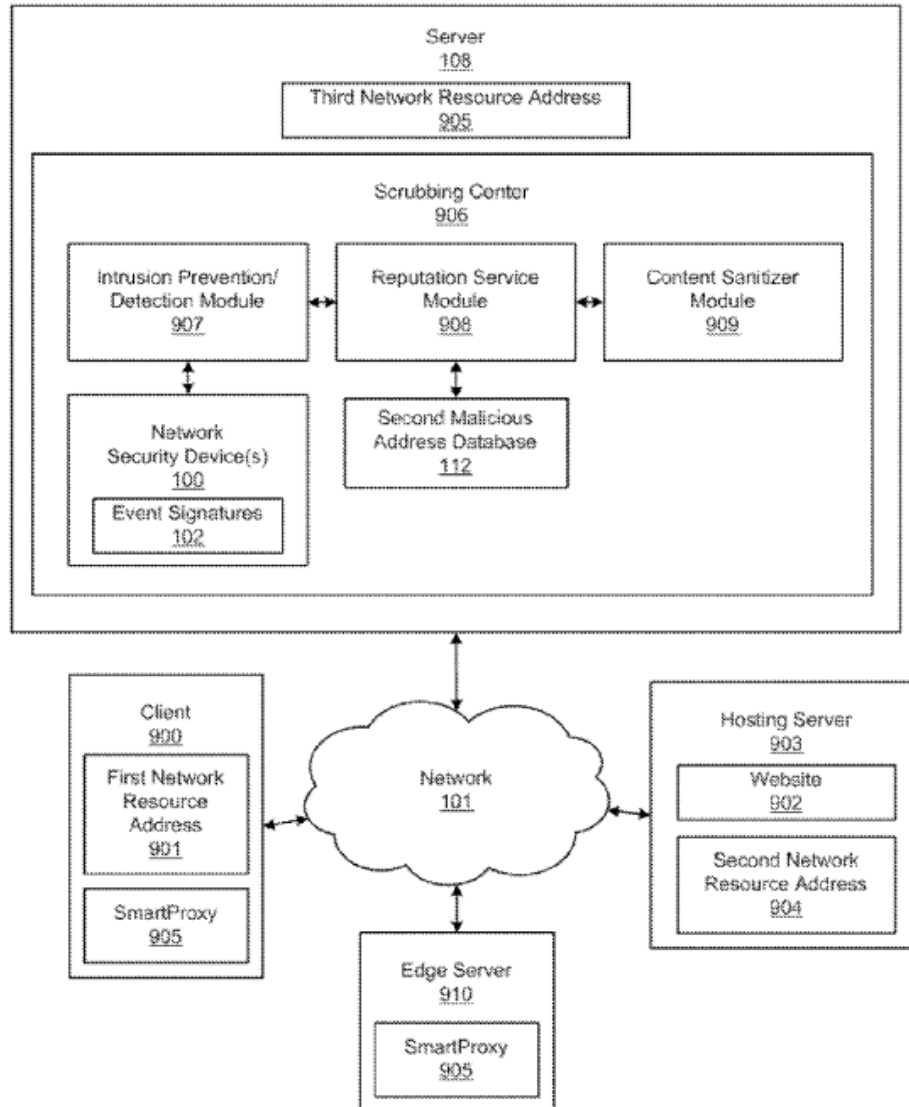


FIG. 9

Figure 9 depicts the disclosed system comprising server 108 with scrubbing center 906, client 900 having first network resource address 901, and hosting server 903 having second network resource address 904, all communicating over network 101. *Id.* ¶ 88. Communications between client 900 and hosting server 903, such as a request for content from website 902, are redirected through server 108, which performs various actions to prevent malware. *Id.* ¶¶ 89, 94, 109, 118.

Figure 10 of Devarajan is reproduced below.

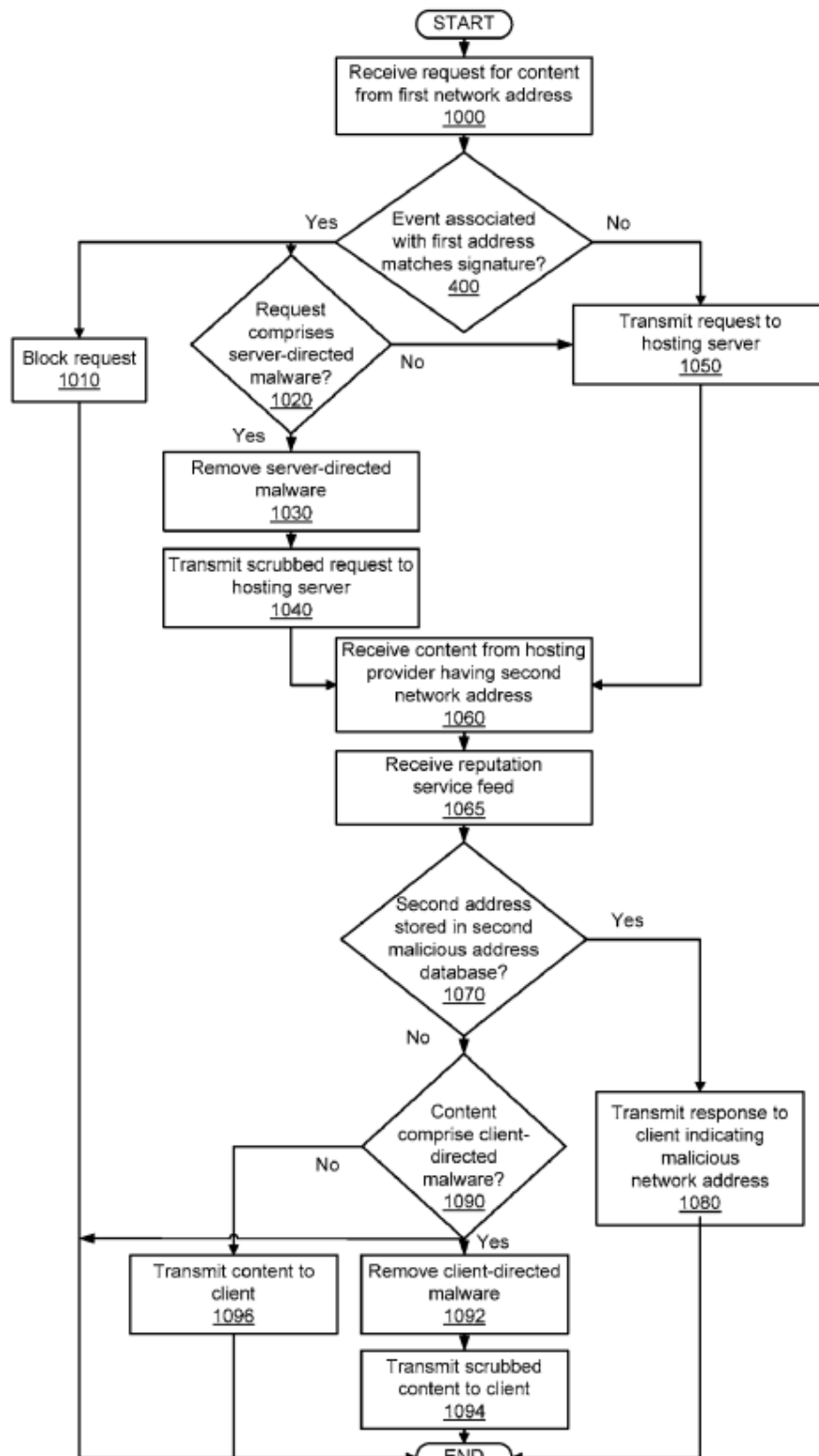


FIG. 10

Figure 10 is a flow chart showing operation of the disclosed system. *Id.*

¶ 109. At Step 1000, server 108 receives from client 900 a “request for content” from website 902 hosted on hosting server 903. *Id.* ¶ 109. At Step 400, server 108 “determin[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.” *Id.* ¶ 111. Devarajan discloses that “[a]n ‘event’ may comprise any malicious or unwanted activity” and event signatures 102 may comprise a plurality of “malware signatures” (e.g., virus signatures, worm signatures) or “attack signatures” (e.g., distributed denial of service (DDoS) attack signatures, SQL injection attack signatures), which may be “based upon prior known attacks.” *Id.* ¶¶ 45–48. Specifically, server 108 intercepts network packets, “collect[s] a stream of transmitted bytes,” and analyzes the stream to “identify strings of characters in the data, known as signatures 102, which may comprise particular strings that have been discovered in known malicious activity.” *Id.* ¶ 45.

“If the event does not match an event signature 102, the request for content may be transmitted, perhaps unaltered, to the hosting server 903 (Step 1050).” *Id.* ¶ 111. “But 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).” *Id.* ¶ 112. Alternatively, if a match exists, server 108 may “determin[e] whether the request for content comprises a server-directed malware” (e.g., a DDoS attack) (Step 1020), remove the malware from the request (Step 1030), “transmit[] a scrubbed request for content” to hosting server 903 (Step 1040), receive the content from hosting server 903 (Step 1060), check

whether the content comprises “client-directed malware” (e.g., a virus), and, if so, remove it and transmit “scrubbed content” to client 900 (Steps 1070–1094). *Id.* ¶¶ 113–118.

2. Claim 1

Petitioner argues that claim 1 would have been obvious based on the disclosure of Devarajan. Pet. 13–37.

With respect to limitation [1A], “intercepting, by a protection agent, a request from a web browser,” Petitioner relies on Step 1000 in Figure 10 above. *Id.* at 19–20. Petitioner first identifies server 108 as a “protection agent” and argues that a person of ordinary skill in the art “would find it obvious to implement server 108 as a software server executing on the same computer (client 900) as the web browser.” *Id.* at 15–17. According to Petitioner, doing so would “reduce[] the communication time between a client and the server 108, . . . thereby improving latency for the overall system,” and “reduce the amount of hardware to implement the system.” *Id.* at 17–18 (citing Ex. 1004 ¶¶ 53, 92). Petitioner also cites SmartProxy 905 on client 900, which “act[s] as an intermediary” facilitating communication between client 900 and hosting server 903, as evidence of software functionality at client 900. *Id.* at 18–19 (quoting Ex. 1004 ¶ 105). Petitioner further argues that server 108 receives “a request for content from a website 902 hosted on a hosting server 903,” initiated by client 900’s browser, at Step 1000. *Id.* at 19–20 (quoting Ex. 1004 ¶ 109) (emphasis omitted). Thus, Petitioner’s position is that server 108, modified to be implemented “as a software server executing on client 900,” constitutes a “protection agent” that “intercept[s] . . . a request from a web browser,” as recited in limitation [1A]. *Id.* at 15–21, 23 (citing Ex. 1002 ¶¶ 54–65).

With respect to limitation [1B], “determining, by the protection agent, whether the request is associated with known malicious content,” Petitioner relies on Step 400 in Figure 10. *Id.* at 21–23. Petitioner first quotes Devarajan’s description of Step 400 as “determin[ing] 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,” then quotes various disclosures pertaining to event signatures 102. *Id.* at 21–23 (quoting Ex. 1004 ¶ 111; citing Ex. 1004 ¶¶ 44–46, 48–49, 53). Petitioner then concludes that a person of ordinary skill in the art “would understand [that] after [server 108] receives a request, step 400 determines whether the event matches the signature of known malicious content.” *Id.* at 22–23 (citing Ex. 1002 ¶¶ 66–70). Thus, although not entirely clear from the Petition, Petitioner’s position appears to be that client 900’s request for content from website 902 is an “event” in Devarajan, and a match with an event signature shows that the event is “associated with known malicious content.” *Id.* at 21–23; *see* Ex. 1002 ¶ 67 (“A [person of ordinary skill in the art] would understand ‘an event associated with a client 900’ to mean a content request from the client to a website 902.”).

We are not persuaded that Petitioner has made a sufficient showing for limitation [1B]. As explained above, we interpret the limitation to not encompass determining whether the request itself is malicious. *See supra* Section II.C. Devarajan discloses two sets of actions: a first set, prior to receiving content from the hosting server, designed to protect the hosting server from server-directed malware transmitted by the client (Steps 400 and 1000–1050 in Figure 10), and a second set, after content has been received, designed to protect the client from client-directed malware transmitted by

the hosting server (Steps 1060–1094 in Figure 10). Ex. 1004 ¶¶ 7, 17, 18, 88–118 (describing the “bi-directional” detection and removal of malware), Figs. 9–10; *see* Ex. 2001 ¶¶ 58–60, 71–72. Step 400 is part of the first set. At Step 400, before transmitting the client’s request for content from the website, server 108 “determin[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 ¶¶ 96, 111. Specifically, server 108 examines the request to identify characters (i.e., signatures) that match strings indicative of malicious activity (e.g., a DDoS attack signature or an SQL injection attack signature). *Id.* ¶¶ 41, 45–48. If a match is found, the request is blocked from reaching hosting server 903 (Step 1010); otherwise, the request is transmitted (Step 1050). *Id.* ¶¶ 98, 111–112. As Patent Owner correctly points out, Devarajan is entirely silent as to what content client 900 is requesting from website 902 at this stage. *See* Prelim. Resp. 39; Ex. 2001 ¶ 72. Again, Step 400 looks only at the request from the client (i.e., “an event associated with the client 900 or its first network resource address 901” in Petitioner’s view) to determine whether the request itself is malicious as to the hosting server. *See* Ex. 1004 ¶¶ 96, 111. Step 400 does not determine whether content associated with the request might be malicious. *See id.* Accordingly, Petitioner has not shown sufficiently that Step 400 discloses limitation [1B].

Step 1020 in Figure 10 similarly involves determining whether the request itself is malicious.² Devarajan discloses, as an alternative to

² Petitioner highlights Step 1020 (along with Steps 400 and 1050) in an annotated version of Figure 10 in its Petition, but does not mention Step 1020 in its analysis of limitation [1B]. *See* Pet. 23; Ex. 1002 ¶ 67.

blocking the request if an event signature match is found, “determining [at Step 1020] whether the request for content comprises a server-directed malware” (e.g., a DDoS attack or an SQL injection attack). *Id.* ¶¶ 100, 113. If the request comprises server-directed malware, server 108 removes it and transmits a “scrubbed request” to the hosting server (Steps 1030 and 1040); otherwise, the request is transmitted as is (Step 1050). *Id.* ¶¶ 101, 114. This further demonstrates that, prior to actually transmitting the request to the hosting server, Devarajan is concerned with protecting the hosting server from server-directed malware (by determining whether the request itself is malicious as to the hosting server), not protecting the client from client-directed malware (by determining whether the request is associated with known malicious content, as claimed).

Petitioner has not shown sufficiently that Devarajan teaches or suggests limitation [1B].³ Accordingly, Petitioner has not shown a reasonable likelihood of prevailing on its assertion that claim 1 is unpatentable over Devarajan.

3. *Claims 6, 7, 9–12, 14, and 16–20*

Similar to limitation [1B], claim 9 recites a system comprising “memory encoding computer executable instructions that, when executed by the at least one processor, perform a method comprising . . . determining

³ Patent Owner also argues that Devarajan does not teach limitations [1C] and [1D] “[f]or substantially the same reasons as limitation [1B],” objective indicia demonstrate the non-obviousness of claim 1, and the Petition should be denied under 35 U.S.C. § 314(a) and *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) (precedential). Prelim. Resp. 3–18, 42–47. In light of our decision to deny institution for the reasons discussed herein, however, we need not address these arguments.

whether the request is associated with known malicious content” (limitation [9D]). Petitioner relies on its earlier analysis regarding limitation [1B].

Pet. 46. Claims 6, 7, 16, 18, and 19 depend from claim 1, and claims 10–12, 14, 17, and 20 depend from claim 9. Accordingly, Petitioner has not shown a reasonable likelihood of prevailing on its assertion that claims 6, 7, 9–12, 14, and 16–20 are unpatentable over Devarajan.

E. Obviousness Grounds Based on Devarajan and Hubbard, and Devarajan and McAfee (Claims 1–20)

Petitioner contends that claims 1–4, 6, 7, 9–12, 14, and 16–20 are unpatentable over Devarajan and Hubbard, and claims 1–20 are unpatentable over Devarajan and McAfee, under 35 U.S.C. § 103(a). Pet. 52–76.

Petitioner refers to its earlier analysis as to how Devarajan allegedly teaches limitations [1B] and [9D]. *Id.* at 56, 58, 60, 73. Petitioner does not rely on either Hubbard or McAfee for limitations [1B] and [9D]. *See id.*

Accordingly, for the reasons explained above, Petitioner has not shown a reasonable likelihood of prevailing on its assertion that claims 1–4, 6, 7, 9–12, 14, and 16–20 are unpatentable over Devarajan and Hubbard, or that claims 1–20 are unpatentable over Devarajan and McAfee. *See supra* Section II.D.

III. CONCLUSION

Based on the arguments and evidence presented by the parties, we conclude that Petitioner has not demonstrated a reasonable likelihood of prevailing with respect to at least one claim of the ’928 patent challenged in the Petition. Therefore, we do not institute an *inter partes* review.

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that the Petition is denied and no *inter partes* review is instituted.

IPR2023-00786
Patent 10,025,928 B2

FOR PETITIONER:

Kat Li
W. Todd Baker
Akshay S. Deoras
Nadia E. Haghighatian
KIRKLAND & ELLIS LLP
kat.li@kirkland.com

FOR PATENT OWNER:

Brian Eutermoser
Russell E. Blythe
Mikaela Stone
KING & SPALDING LLP
beutermoser@kslaw.com
rblythe@kslaw.com
mikaela.stone@kslaw.com