Trials@uspto.gov
571-272-7822

Paper No. 7
Entered: March 11, 2016

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SYMANTEC CORP.,
Petitioner,

v.

FINJAN, INC.,
Patent Owner.

Case IPR2015-01894
Patent 6,154,844

Before JAMES B. ARPIN, ZHENYU YANG, and
CHARLES J. BOUDREAU, *Administrative Patent Judges*.

BOUDREAU, *Administrative Patent Judge.*

DECISION
Denying Institution of *Inter Partes* Review
37 C.F.R. § 42.108

I. INTRODUCTION

Symantec ("Petitioner") filed a Petition (Paper 1, "Pet.") requesting *inter partes* review pursuant to 35 U.S.C. § 311 of claims 1, 7, 11, 15, 16, 41, and 43 of Patent No. US 6,154,844 to Touboul et al. (Ex. 1001, "the '844 patent"). Pet. 1. Finjan, Inc. ("Patent Owner") filed a Preliminary Response. Paper 6 ("Prelim. Resp."). We review the Petition under 35 U.S.C. § 314, which provides that an *inter partes* review may not be instituted "unless . . . there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition." 35 U.S.C. § 314(a).

For the reasons that follow and on this record, we are not persuaded that Petitioner demonstrates a reasonable likelihood of prevailing in showing the unpatentability of any of the challenged claims on the asserted grounds. Accordingly, we *deny* Petitioner's request to institute an *inter partes* review.

A. The '844 Patent

The '844 patent, titled "System and Method for Attaching a Downloadable Security Profile to a Downloadable," issued November 28, 2000, from U.S. Patent Application No. 08/995,648 ("the '648 application"), filed December 22, 1997. Ex. 1001, [21], [22], [45], [54].

The '844 patent is directed to systems and methods "for attaching a Downloadable security profile to a Downloadable to facilitate the protection of computers and networks from a hostile Downloadable." Ex. 1001, 1:23–27. "A 'Downloadable' is an executable application program . . . downloaded from a source computer and run on [a] destination computer." *Id.* at 1:44–47. "Examples of Downloadables include Java™ applets . . . ,

JavaScript™ scripts . . . , ActiveX™ controls . . . , and Visual Basic [scripts]." *Id.* at 1:49–55. "Downloadables may also include plugins, which add to the functionality of an already existing application program." *Id.* at 1:56–57.

According to the '844 patent, "the Internet has become a major source of many system damaging and system fatal application programs, commonly referred to as 'viruses,'" and "programmers continue to design computer and computer network security systems for blocking these viruses." *Id.* at 1:32–39. Although, "[o]n the most part, these security systems have been relatively successful," the '844 patent explains, "these security systems are not configured to recognize computer viruses which have been attached to or configured as Downloadable application programs, commonly referred to as 'Downloadables.'" *Id.* at 1:40–44. The subject matter of the '844 patent, accordingly, "provides systems for protecting a network from suspicious Downloadables." *Id.* at 1:62–63. "The network system includes an inspector for linking Downloadable security profiles to a Downloadable, and a protection engine for examining the Downloadable and Downloadable security profiles to determine whether or not to trust the Downloadable security profiles." *Id.* at 1:65–2:2. In particular, according to the '844 patent, the system and method "may examine the Downloadable code to determine whether the code contains any suspicious operations, and thus may allow or block the Downloadable accordingly." *Id.* at 2:54–3:2. Further, "because the system and method . . . link a verifiable Downloadable security profile a Downloadable, the system and method may avoid decomposing the Downloadable into the Downloadable security profile on the fly." *Id.* at 3:3–7.

In the operation of a preferred embodiment of the ’844 patent, a developer obtains or generates an uninspected Downloadable and transmits the Downloadable to the inspector for “hostility inspection,” along with a developer certificate used to authenticate the developer. Ex. 1001, 3:55–65. The inspector includes a “content inspection engine” for examining a received Downloadable (e.g., the signed Downloadable from the developer), for generating a Downloadable security profile (“DSP”) based on a rules base, and for attaching the DSP to the Downloadable. *Id.* at 3:66–4:4. The DSP “preferably includes a list of all potentially hostile or suspicious computer operations that may be attempted by the Downloadable, and may also include the respective arguments of these operations.” *Id.* at 4:4–7. Preferably, the content inspection engine performs a full-content inspection, and generating a DSP includes searching the Downloadable code for undesirable patterns and patterns suggesting the code was written by a hacker, as well as comparing a Downloadable against Downloadables known to be hostile, Downloadables known to be non-hostile, and Downloadables previously examined by the content inspection engine. *Id.* at 4:7–17. After performing content inspection, the inspector attaches an inspector certificate to the Downloadable, verifying the authenticity of the DSP attached to the Downloadable, and transmits the signed, inspected Downloadable to a web server. *Id.* at 4:65–5:5. The web server then may transmit the Downloadable via a network gateway to a computer client. *Id.* at 5:11–13.

B. Priority Date of the ’844 Patent

The Related U.S. Application Data field on the front page of the ’844 patent includes a reference to Provisional Application No. 60/030,639 (“the

’639 provisional”), filed November 8, 1996. Ex. 1001, [60]. Separately, in a section entitled “PRIORITY REFERENCE TO RELATED APPLICATIONS,” the ’844 patent states that the ’648 application “claims benefit of and . . . incorporates by reference [the ’639 provisional]; patent application Ser. No. 08/964,388, . . . filed on Nov. 6, 1997 . . . ; and patent application Ser. No. 08/790,097, . . . filed on Jan. 29, 1997.” *Id.* at 1:7–17. That statement, however, does not indicate how the ’648 application and the earlier applications are related to one another (e.g., as continuations, continuations in part, or divisionals). *See* 37 C.F.R. § 1.78(a)(2) (1997) (“Any nonprovisional application claiming the benefit of one or more prior filed copending nonprovisional applications . . . must contain or be amended to contain in the first sentence of the specification following the title a reference to each such prior application . . . indicating the relationship of the applications.”).

In March 2003, Patent Owner filed a “Petition to Amend Priority Claims Listed in Patent” (Ex. 1005, 1–3). Patent Owner’s petition acknowledges that the ’639 provisional had expired before the filing date of the ’648 application and that priority “cannot be claimed directly from this application,” but asserts that “priority is still achieved through the chain of priority” and requests that “[r]eference to the indirect claim of priority to the provisional application” be added. Ex. 1005, 2. Notably, the proposed amendment also does not indicate the relationship among the applications as required by 37 C.F.R. § 1.78(a)(2).

Petitioner points out that the Office never ruled upon Patent Owner’s petition and contends that the petition was improper, in any event, under the rules in effect at the time the ’648 application was filed. Pet. 4. Petitioner

further contends that, even assuming arguendo that the ’844 patent did include a priority claim to the referenced earlier applications, the challenged claims still would not be entitled to the benefit of earlier priority based on any of those applications, because the specifications of those earlier applications differ from the ’648 application and lack sufficient disclosure with respect to the claimed subject matter of the ’844 patent to satisfy the requirements of 35 U.S.C. § 112. Pet. 4–5. Patent Owner does not provide any substantive response to these arguments in its Preliminary Response, but merely contends that the ’844 patent claims the benefit of and incorporates the earlier applications by reference. Prelim. Resp. 3.

In light of Petitioner’s arguments and the absence of any substantive response thereto by Patent Owner, we are persuaded that the present record does not establish that the challenged claims are entitled to the benefit of any filing date earlier than December 22, 1997. *See, e.g.*, *Nintendo of Am. Inc. v. iLife Techs.*, Case IPR2015-00106, slip op. at 16 (PTAB Apr. 29, 2015) (Paper 12) (“Patent Owner must come forward with evidence and argument—either in its Preliminary Response or, if trial is instituted, in its Response—showing why the challenged claims is supported by the written description of the priority application.”); *Polaris Wireless, Inc. v. TruePosition, Inc.*, Case IPR2013-00323, slip op. at 29 (PTAB Jun. 4, 2013) (Paper 9) (explaining that there is no presumption of earlier priority where the specifications of the earlier applications are not the same).

C. Related Proceedings

The ’844 patent is the subject of a district court action between the parties, *Finjan, Inc. v. Symantec Corp.*, 3:14-cv-02998 (N.D. Cal. 2014), and also has been asserted in five other district court actions: *Finjan, Inc. v.*

FireEye, Inc., 4:13-cv-03133 (N.D. Cal. 2013), *Finjan, Inc. v. Blue Coat Systems, Inc.*, 5:13-cv-03999 (N.D. Cal. 2013), *Finjan, Inc. v. Proofpoint, Inc.*, 3:13-cv-005808 (N.D. Cal. 2013); *Finjan, Inc. v. Sophos, Inc.*, 3:14-cv-01197 (N.D. Cal. 2014), and *Finjan, Inc. v. Blue Coat Systems, Inc.*, 5:15-cv-03295 (N.D. Cal. 2015). Pet. 1; Paper 4, 1. Petitioner also has filed petitions seeking *inter partes* review of other patents involved in the above-referenced *Finjan, Inc. v. Symantec Corp.* district court action. Pet. 1. Blue Coat Systems, Inc. also has filed a petition seeking *inter partes* review of the '844 patent on the grounds as presented in the instant Petition. Case IPR2016-00498, Paper 3.

D. Illustrative Claim

Among the challenged claims, claims 1, 15, 41, and 43 are independent. Claim 1 is illustrative and is reproduced below:

> 1. A method comprising:
>
> receiving by an inspector a Downloadable;
>
> generating by the inspector a first Downloadable security profile that identifies suspicious code in the received Downloadable; and
>
> linking by the inspector the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to Web clients.

Ex. 1001, 11:13–20. Independent claims 15 (inspector system), 41 (computer-readable storage medium), and 43 (inspector system) include similar limitations. *Id.* at 11:62–12:2, 14:8–18, 14:35–42. Each of claims 7 and 11 depends directly from independent claim 1, and claim 16 depends from independent claim 15.

E. References Relied Upon

Petitioner relies on the following references:

Exhibit	Reference
1006	US 5,825,877, issued Oct. 20, 1998 (filed June 11, 1996) ("Dan")
1007	US 5,978,484, issued Nov. 2, 1999 (filed Apr. 25, 1996) ("Apperson")
1008	US 5,313,616, issued May 17, 1994 ("Cline")
1009	US 5,623,600, issued Apr. 22, 1997 (filed Sept. 26, 1995) ("Ji")
1010	Rangachari Anand et al., *A Flexible Security Model for Using Internet Content*, IEEE COMPUTER SOC'Y PROC. 16TH SYMP. ON RELIABLE DISTRIBUTED SYS. (1997) ("Anand")

Pet. 6–7. Petitioner also relies on the Declaration of Jack W. Davidson, Ph.D. (Ex. 1017).

F. Asserted Grounds of Unpatentability

Petitioner challenges the patentability of the challenged claims on the following grounds:

Reference(s)	Basis	Claims Challenged
Dan	§ 103	1, 7, 11, 15, 16, 41, and 43
Apperson, Cline, and Ji	§ 103	1, 7, 11, 15, 16, 41, and 43
Anand and Cline	§ 103	1, 7, 11, 15, 16, 41, and 43

Pet. 8.

II. DISCUSSION

A. Claim Construction

In an *inter partes* review proceeding, claims of an unexpired patent are given their broadest reasonable interpretation in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b); *In re Cuozzo Speed Techs., LLC*, 793 F.3d 1268, 1275–79 (Fed. Cir. 2015), *cert. granted sub nom. Cuozzo Speed Techs. LLC v. Lee*, 136 S. Ct. 890 (2016). Under this standard, we presume that claim terms have their ordinary and customary meaning. *See In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007) ("The ordinary and customary meaning is the meaning that the term would have to a person of ordinary skill in the art in question.") (internal quotation marks omitted). A patentee, however, may rebut this presumption by acting as his own lexicographer, providing a definition of the term in the specification with "reasonable clarity, deliberateness, and precision." *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994).

Petitioner proposes constructions for three limitations recited in means-plus-function format in independent claim 43: "means for receiving," "means for generating," and "means for linking." Pet. 14–17. Patent Owner responds to each of Petitioner's proposed constructions, offering competing constructions for the second and third of those limitations. Prelim. Resp. 5–10. Pursuant to 37 C.F.R. § 42.104(b)(3), the parties identify specific portions of the '844 patent's specification that they allege describe the structures corresponding to the claimed functions. Neither party, however, explains how the construction of these limitations is material to our decision of whether to institute a trial. On this record and for purposes of this Decision, we determine that no claim terms require express construction.

See Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc., 200 F.3d 795, 803 (Fed. Cir. 1999) (stating that claim terms need only be construed to the extent necessary to resolve the controversy).

B. Asserted Grounds of Unpatentability

1. Overview

Petitioner argues that claims 1, 7, 11, 15, 16, 41, and 43 of the '844 patent are rendered obvious by the references described above. *See supra* Sec. I.F. A patent claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the claimed subject matter and the prior art are "such that the subject matter[,] as a whole[,] would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations, including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art[1]; and (4) objective evidence of nonobviousness, i.e., secondary considerations.[2] *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966). On this record and for the reasons set forth below, we are not persuaded that Petitioner demonstrates a reasonable

[1] Petitioner proposes a definition for a person of ordinary skill in the art. Pet. 13–14; *see* Ex. 1017 ¶¶ 27–29. Patent Owner does not challenge this definition. For purposes of this Decision and to the extent necessary, we adopt Petitioner's definition.

[2] Although Patent Owner presents arguments regarding secondary considerations (Prelim. Resp. 45–48), we need not reach those arguments because we find that Petitioner's evidence and arguments regarding the obviousness of the challenged claims are deficient.

likelihood of prevailing in the challenges to claims 1, 7, 11, 15, 16, 41, and 43 of the ’844 patent.

2. Obviousness over Dan

a. Dan

Dan teaches a form of authentication in which a trusted third party, referred to in a preferred embodiment as a certification agency (“CA”), signs a certificate to identify the author of a program and to secure its integrity. Ex. 1006, 1:40–43, 49–51. The program code is associated with the certificate and an access control list (“ACL”). *Id.* at 1:43–45. The ACL describes the permissions and resources required by the code and is provided by the developer (referred to as the “code production system,” or “CPS”). *Id.* at 1:45–51, 2:20–28. The ACL also includes a Logical Resource Table (“LRT”), which “contains a row for each call to an external routine required by the code,” along with a listing of associated parameters, as well as a Physical Resources Table (“PRT”), containing the physical resources required by the code. *Id.* at 3:49–4:15. Once the CA issues the certificate, it is not possible for any party to modify either the code or the ACL without invalidating the certificate. *Id.* at 1:51–55. A client downloading the code or ACL can verify the integrity of the code and ACL, and the system can enforce the access list, such that the permissions and resources are not exceeded. *Id.* at 1:56–59.

b. Discussion

Petitioner contends that Dan renders obvious each of claims 1, 7, 11, 15, 16, 41, and 43 of the ’844 patent. Pet. 17–30. Petitioner equates the program code in Dan with the “Downloadable” recited in the challenged

claims, Dan's CA with the claimed "inspector," and the external routines identified by the LRT of Dan's ACL with "suspicious code." *Id.* at 19–23. Petitioner variously identifies the ACL, PRT, and LRT tables as the claimed DSP that identifies suspicious code in the Downloadable. *Id.* at 23–24. Petitioner contends that, "[a]lthough Dan does not expressly teach that this [DSP] is generated by the CA (*i.e.*, inspector), this would have been obvious" to a person of ordinary skill in the art. *Id.* at 24. Petitioner explains:

> Dan teaches that the CA (*i.e.*, inspector) can verify an ACL (*i.e.*, Downloadable security profile) associated with a Downloadable provided by a Code Production System (CPS). [Ex. 1006,] 4:19–25, 2:18–43. In particular, Dan teaches that this verification of the ACL by the CA may be used as an alternative to enforcement by a client. [*Id.* at] 4:23–25. Thus, a [person of ordinary skill in the art] would have understood that such verification at the CA could be performed in the same manner as the enforcement at the client using the functions associated with the ACL enforcer. [*Id.* at] 3:38–41 ("Before allowing access to any resource, the executor invokes the ACL enforcer for checking the validity of the access."), 4:59–61, 5:5–10, and 5:17–22; [Ex. 1017] ¶ 116–118 (explaining that enforcement is akin to verification).
>
> In particular, it would have been obvious to a [person of ordinary skill in the art] that these same verification/enforcement techniques (*e.g.*, static and dynamic code analyzes) could be used by the CA in Dan to generate the ACL (*i.e.*, Downloadable security profile) instead of verifying an existing ACL that was provided by a CPS. [*Id.* at] ¶ 119. A [person of ordinary skill in the art] would have been motivated to generate the Downloadable security profile at the CA for a number of reasons, including in situations where the CPS is untrusted or does not provide an ACL. [*Id.* at] ¶ 120.

Pet. 24–25.

Petitioner also asserts that, because Dan teaches that the CA links the ACL to the program code before the code is made available from a server to clients, Dan teaches "linking [by the inspector] the first DSP to the Downloadable before a web server makes the Downloadable available to web clients," as required by each of the challenged independent claims. *Id.* at 26–27.

Patent Owner responds, *inter alia*, that Petitioner has not demonstrated that Dan discloses either "[means for] generating [by the inspector] a first Downloadable security profile that identifies suspicious code in the received Downloadable" or "[means for] linking the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to web clients," as required by each of the challenged claims. Prelim. Resp. 12–20.

With respect to the "generating" step, Patent Owner raises four arguments: First, ACLs are not DSPs, and Petitioner's broad interpretation of DSP is "completely at odds" with the teachings of the '844 patent and U.S. Patent No. 6,092,194 (Ex. 1003, "the '194 patent"), incorporated by reference therein,[3] which demonstrate that ACLs not only are distinct from DSPs, but are instead what the generated DSPs are compared against. *Id.* at 13–14 (citing Ex. 1003, 6:5–19, Fig. 3). Second, ACLs, including Dan's ACL, are not DSPs, because they do not identify suspicious code in the received Downloadable. *Id.* at 14. Whereas Petitioner relies on the existence of an LRT in ACL to explain how the ACL identifies suspicious

[3] The '194 patent issued from the '388 application, which as noted above, is stated to be incorporated by reference in the '844 patent. *See* Ex. 1003, [21]; Ex. 1001, 1:12–15.

code in the received Downloadable, Patent Owner contends, Dan's LRT merely contains the logical resources "required" by the code, and nowhere does Dan state that the LRT further includes identification of suspicious code for any of the required calls. *Id.* (citing Ex. 1006, 3:49–66). Third, "nothing suggests that Dan discloses an inspector that generates the claimed DSP." *Id.* at 16. Whereas "Petitioner relies on Dan's [CA] to be the claimed inspector and the ACL to be the claimed DSP that identifies suspicious code," Patent Owner continues, "the Petition acknowledges that the Dan's [CA] never generates the Dan's ACL," and "[i]ndeed, Dan's [CA] is designed to avoid any need to ever generate an ACL as they are always received from the code producer who authored of the associated code." *Id.* (citing Pet. 20, 24; Ex. 1006, 2:25–37, Fig. 1). Finally, Patent Owner contends that Dan's "ACL enforcer" does not generate a DSP, but "only 'ensures that the permissions and resources specified in the ACL for the code are provided and no additional permissions/resources are allowed.'" *Id.* at 17–18 (citing Ex. 1006, 4:15–18).

With respect to the "linking" step, Patent Owner contends that, at least because "Dan does not disclose generating a first Downloadable security profile that identifies suspicious code in the received Downloadable," Dan also "cannot disclose ['linking the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to web clients.'" *Id.* at 20.

We are persuaded by Patent Owner's arguments that Petitioner has not demonstrated on this record that Dan teaches or suggests an inspector generating a DSP that identifies suspicious code in a received Downloadable, and that Dan, therefore, also does not teach or suggest

linking such a DSP to the Downloadable. *Id.* at 12–20. The portions of Dan cited by Petitioner describe methods and systems in which a developer or code producer (i.e., Dan's CPS) provides an ACL to a certification agency (Dan's CA) along with a program for certification. *See, e.g.*, Ex. 1006, 1:45–51, 2:20–28. Whereas Petitioner identifies the CA as the claimed "inspector" and the received program as the claimed "Downloadable" (Pet. 20), there is no indication in the cited portions of Dan that the CA "identifies suspicious code" in the received program. Rather, the CA is disclosed to sign a certificate identifying the author of a program and to secure the integrity of the program and its ACL. Ex. 1006, 1:40–43, 1:49–51, 2:34–37. Neither of those functions requires inspecting the content of the code to identify suspicious code. And in fact, we discern nothing in Dan that would preclude Dan's CA from providing a signed certificate if a CPS were to provide to the CA even code infected with a known virus, so long as the CPS also provided therewith an ACL accurately identifying the permissions and resources required by the code.

Moreover, we are not persuaded by Petitioner's contention that it would have been obvious to a person of ordinary skill in the art that the "same verification/enforcement techniques (*e.g.*, static and dynamic code analyzes [*sic*]) could be used by the CA in Dan to generate the ACL (*i.e.*, Downloadable security profile) instead of verifying an existing ACL that was provided by a CPS." Pet. 25. Dan teaches that "*ACL enforcement* may be static or dynamic" (Ex. 1006, 4:19 (emphasis added)), not "static and dynamic code analy[sis]," as Petitioner suggests (Pet. 25). Whereas Petitioner cites Dr. Davidson's declaration as "explaining that enforcement is akin to verification" (Pet. 25 (citing Ex. 1017 ¶¶ 116–18)) we find on this

record no evidence suggesting that either enforcement or verification is akin to *code analysis*. Additionally, although the static and dynamic enforcement methods described by Dan differ in the location and timing of the described enforcement function (i.e., at the CA before execution vs. at the client system at the time of execution) (Ex. 1006, 4:19–26), we discern no evidence that either method of ACL enforcement involves generating a DSP that identifies suspicious code.

On this record, Petitioner has not identified sufficient evidence that Dan teaches or suggests all of the limitations recited in independent claims 1, 15, 41, and 43, and, in particular, "generat[ing] . . . a first Downloadable security profile that identifies suspicious code in [a] Downloadable" and "linking . . . the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to web clients." Consequently, we are not persuaded that Petitioner demonstrates a reasonable likelihood that it would prevail at trial in showing that the subject matter of those claims or of dependent claims 7, 11, or 16 would have been obvious over Dan.

3. Obviousness over Apperson, Ji, and Cline

a. Apperson

Apperson relates generally to a "method and system for distributing and executing executable code," wherein, before sending the code to a client, a "distributing authority" associates a "privilege request code" with the executable code. Ex. 1007, Abst. The privilege request code indicates "a set of privileges or privilege categories that the executable code might perform on the client machine" during execution, including, for example,

file input/output, network operations, registry read/write rights, graphics operations, window management operations, and user list read/write rights. *Id.* at Abst., 2:44–47, 4:33–41. To prevent modification of the software code and associated privilege request code, "[t]he distributing authority digitally signs the executable code and the privilege request code, and also provides a certificate that can be traced by the client to a known certifying authority." *Id.* at 2:47–53. As part of the signing process, Apperson teaches that the privilege request code may be concatenated with (appended to) the code. *Id.* at 4:44–54, 8:17–25, Fig. 2.

b. Ji

Ji describes a system for detecting and eliminating viruses on a computer network, wherein a File Transfer Protocol (FTP) proxy server is used to scan incoming and outgoing files for viruses and to transfer those files if they do not contain viruses. Ex. 1009, Abst. Ji discloses a method for processing a file before transmission into or from a network, including the steps of receiving a data transfer command and file name; transferring the file to a proxy server or system node; performing virus detection on the file; and determining whether the file contains any viruses. *Id.* at Abst., 3:4–11. If the file does not contain any viruses, the file is transferred from the system to a recipient node. *Id.* at Abst., 3:11–12. If the file does contains a virus, the file is deleted or some other preset action is performed. *Id.* at Abst., 3:13–14.

c. Cline

Cline describes a method for certifying the portability of software between computer systems, including certification tests to ensure that

application programs will run on any conforming computer system regardless of the vendor. Ex. 1008, 2:66–3:5. The certification tests include a static analysis, in which the object code of an application program is analyzed against a "conformance database" of allowable external calls to determine whether any illegal or erroneous calls are being made, and a dynamic analysis, in which the application program is analyzed as it is being run to determine any runtime errors in the calls. *Id.* at 3:6–16. If no errors are detected in either analysis, the application program then is certified to be compatible and transportable without change between all certified compatible computer systems. *Id.* at 3:16–21.

d. Discussion

Petitioner contends that the combination of the teachings of Apperson, Ji, and Cline would have rendered obvious the subject matter of claims 1, 7, 11, 15, 16, 41, and 43 of the '844 patent. Pet. 30–52. In particular, Petitioner contends that Apperson's distributing authority corresponds to the claimed inspector system and that Apperson describes the need for code inspection to ensure the safety of distributed code. *Id.* at 31. To the extent that Apperson does not explicitly disclose that the distributing authority "receives a Downloadable," Petitioner contends that feature is taught by Ji. *Id.* at 33–34. Petitioner asserts that it would have been obvious for a person of ordinary skill in the art to combine the teachings of Apperson and Ji, because both references are directed to protecting computer systems from viruses and malicious code, and such a combination would have merely amounted to combining well-known prior art elements. *Id.* at 34–35.

Further, Petitioner asserts, to the extent that Apperson and Ji do not teach a content inspection engine to "generate" privilege request codes for a

Downloadable, this feature is taught by Cline. *Id.* at 36. According to Petitioner, "Cline is concerned with verifying/certifying external calls made by and executable (*i.e.*, those functions not defined within the executable itself) such as system calls and library calls," and "[l]ike Apperson, Cline explains that these system calls correspond to the same types of 'suspicious operations' exemplified by the '844 patent (*e.g.*, network, window management, or file operations)." *Id.* at 36–37. Petitioner also contends it would have been obvious for a person of ordinary skill in the art to combine the teachings of Apperson and Ji with those of Cline. *Id.* at 39. According to Petitioner, "Apperson in view of Ji teaches receiving a Downloadable over a network and associating it with a privilege request code," but "Apperson . . . does not provide a discussion on how a privilege request code is determined," and "[a]ccordingly, a [person of ordinary skill in the art] would have turned to Cline to determine a privilege request code for an executable." *Id.* at 39–40.

Patent Owner raises several arguments in response to Petitioner's contentions, including that Cline is not analogous art to the '844 patent, that Petitioner's proposed combination of references is the product of impermissible hindsight bias, and that Petitioner has not demonstrated that the proposed combination discloses either "[means for] generating [by the inspector] a first Downloadable security profile that identifies suspicious code in the received Downloadable" or "[means for] linking the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to web clients," as required by each of the challenged claims. Prelim. Resp. 24–34, 36–37.

As an initial matter, we are not persuaded on this record by Patent Owner's assertions that Cline is not analogous art to the '844 patent. *Id.* at 24–27. Although Cline is concerned with interoperability, rather than security per se, we decline Patent Owner's invitation to define the field of endeavor so narrowly. Both Cline and the '844 patent fundamentally are concerned with the analysis of computer code, even if their intended applications differ. Moreover, we are not prepared on this record to say that methods that may be employed in determining compatibility are not reasonably pertinent to identifying security threats. *See In re Klein*, 647 F.3d 1343, 1348 (Fed. Cir. 2011) ("A reference is reasonably pertinent if, even though it may be in a different field from that of the inventor's endeavor, it is one which, because of the matter with which it deals, logically would have commended itself to an inventor's attention in considering his problem.").

Nonetheless, we are persuaded by Patent Owner's substantive arguments that Petitioner has not demonstrated on this record that the combination of Apperson, Ji, and Cline teaches or suggests an inspector generating a DSP that identifies suspicious code in a received Downloadable, and that that combination, therefore, also does not teach or suggest linking such a DSP to the Downloadable. Prelim. Resp. 32–34, 36–37. We agree, in particular, with Patent Owner that Petitioner has not demonstrated that Apperson teaches or suggests a Downloadable security profile that identifies suspicious code. *Id.* at 32–33. Although Apperson discloses, for example, that "there must be a guarantee that . . . hostile code (viruses, Trojan horses, etc.) will not be generated and installed on [a] client system" (Ex. 1007, 1:41–48), Apperson's approach to providing that

guarantee does not involve an inspector generating a Downloadable security profile that identifies suspicious code in a received Downloadable. As Patent Owner points out, Apperson's privilege request code, cited by Petitioner for that element, merely indicates "the set of privileges or privilege categories" that executable code might perform during execution (Ex. 1007, 4:33–41), similar to the list of "resources" provided in Dan's ACL, as discussed in Section II.B.2, *supra*. For reasons similar to those stated above with respect to Dan's ACL, we are not persuaded that Apperson's privilege request code identifies "suspicious code" within the meaning of the challenged claims.

Petitioner has not cited Ji in connection with this element, and we also agree with Patent Owner (*see* Prelim. Resp. 33–34) that Cline does not remedy the deficiency in Apperson. Whereas Petitioner asserts that a person of ordinary skill in the art "would have understood that Cline's SBV (static) analysis . . . limits the conformance analysis to 'suspicious calls' (e.g., library calls or system calls) stored in the database" (Pet. 37), there is no indication in Cline that the cited "library calls or system calls" are suspicious. Indeed, because Cline is concerned with interoperability, not security, we find no suggestion in Cline that Cline's static or dynamic analyses would identify as suspicious even code infected with known viruses, so long as each of the calls made by the infected code was, individually, in Cline's compliance database.

We also agree with Patent Owner that Petitioner provides insufficient motivation to combine Apperson with Cline. Prelim. Resp. 31–32. As quoted above, Petitioner asserts that "Apperson . . . does not provide a discussion on how a privilege request code is determined," and

"[a]ccordingly, a [person of ordinary skill in the art] would have turned to Cline to determine a privilege request code for an executable." Pet. 39–40. Although Petitioner then contends that "[o]ne of ordinary skill in the art would have been motivated to combine these teachings for a number of reasons," the only motivation Petitioner actually offers is "to protect client computers on a network (*e.g.*, an Intranet) from downloaded/incoming viruses and malicious code without being language specific or requiring source code (which may not be available at all)." *Id.* at 40 (citing Ex. 1008, 11:44–63; Ex. 1017 ¶¶ 185–186). This is precisely the same motivation that Petitioner alleges for combining Apperson with Ji (Pet. 35), and, as Patent Owner points out, the portion of Cline cited by Petitioner to support this alleged motivation is unrelated to "protect[ing] client computers on a network" (Prelim. Resp. 31–32 (citing Ex. 1008, 11:44–63)). The cited paragraphs of Dr. Davidson's declaration also fail to provide a persuasive explanation as to why a person of ordinary skill in the art would turn to Cline to determine Apperson's privilege request code. Ex. 1017 ¶¶ 185–186. Accordingly, we agree with Patent Owner that Petitioner has failed to provide "articulated reasoning with some rational underpinning to support [its] legal conclusion of obviousness." Prelim. Resp. 31 (quoting *KSR*, 550 U.S. at 418).

On this record, Petitioner has not identified sufficient evidence that the combination of the teachings of Apperson, Ji, and Cline teaches or suggests all of the limitations recited in independent claims 1, 15, 41, and 43, and, in particular, "generat[ing] . . . a first Downloadable security profile that identifies suspicious code in [a] Downloadable" and "linking . . . the first Downloadable security profile to the Downloadable before a web server

makes the Downloadable available to web clients." Consequently, we are not persuaded that Petitioner demonstrates a reasonable likelihood that it would prevail at trial in showing that the subject matter of those claims or of dependent claims 7, 11, or 16 would have been obvious over the asserted combination.

4. Obviousness over Anand and Cline

Petitioner contends that the combination of the teachings of Anand and Cline would have rendered obvious the subject matter of claims 1, 7, 11, 15, 16, 41, and 43 of the '844 patent. Pet. 47–60. For substantially the same reasons as set forth in our discussion of the first and second asserted grounds in Sections II.B.2 and II.B.3, *supra*, we are not persuaded that Petitioner demonstrates a reasonable likelihood that it would prevail on this ground with respect to any of the challenged claims.

a. Anand

Anand teaches "a system for downloading content from the Internet and controlling its actions on a client machine." Ex. 1010, 1. Anand recognizes that because the downloaded content may be malicious and may damage the user's machine, downloading principals may need to "prevent content from: (1) reading private files; (2) writing executable files; (3) limit access to their system's CPU; and (4) prevent arbitrary remote communication from their system." *Id.*

In Anand, manufacturers and content rating services may create a content stamp to annotate content with authentication and execution information. *Id.* at 3. Figure 2 of Anand, reproduced below, shows the fields of the content stamp. *Id.*

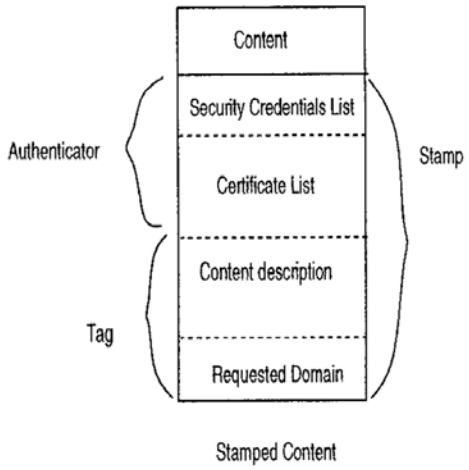

Figure 2. Structure of stamped content

As shown in Figure 2 above, the content stamp includes an authenticator, which further includes a security credentials list and a certificate list, and a tag, which further includes a content description and a requested domain. *Id.* at 3. The security credentials list includes a hash of the content. *Id.* Once a downloading principal receives encrypted, stamped content, "[t]he analysis module computes a hash of the downloaded content and compares it to the hash in the stamp to verify that the content has not been modified." *Id.* at 4.

After the content is authenticated, the analysis module uses the content stamp, the downloading principal's policy database, and some user intervention to derive the content's protection domain. *Id.* at 3. Specifically, the requested domain "specifies the protection domain that the content requests for executing the content." *Id.* at 4. The protection domain determines the access rights the content has on the downloading principal's machine. *Id.*

b. Discussion

Petitioner contends that Anand teaches nearly all limitations of the challenged independent claims, with the exception of "a 'content inspector engine' to 'generate' privilege request codes for a Downloadable," which Petitioner contends is expressly taught by Cline. Pet. 48–58. In particular, Petitioner equates Anand's manufacturer or content rating service with the claimed "inspector" or "inspector system" (*id.* at 49–50); content received from manufacturers by Anand over a network with the claimed "Downloadable" (*id.* at 50–51); and the "requested domain" in the tag portion of Anand's content stamp with the claimed "Downloadable security profile." *Id.* at 49–52. Petitioner asserts that, "[a]ccording to Anand, the 'requested domain,' identifies suspicious code in the downloadable." *Id.* at 52. "For example, Anand teaches that the 'requested domain, specifies the protection domain that the content requests for executing the content. Content may need access to the following types of resources: the file system, memory, CPU, remote principals, and the downloading principal's display.'" *Id.* (quoting Ex. 1010, 4). Petitioner also cites column 4, lines 20–34, of the '844 patent as "describing similar examples of suspicious operations." *Id.* Petitioner further contends,

> Although Anand does not teach a "content inspection engine" to "generate" privilege request codes for a Downloadable, this feature is expressly taught by Cline. . . . Cline generally teaches using static and dynamic analyses to determine the procedure calls made by executable code. Cline's SBV and DBV analyzers determine the system calls made by the program (i.e., suspicious operations). [Ex. 1017 ¶¶ 231–234]
>
> Accordingly, a [person of ordinary skill in the art] would have understood that Cline's SBV could be used to generate a requested domain for the application (i.e., a Downloadable

> security profile). [*Id.*] ¶ 235. Similarly, it would have been obvious that the log database of the DBV analysis could be used to generate a privilege request code for the application (i.e., a Downloadable security profile). *Id.*[] ¶ 236.
>
> . . . It would have been obvious for a [person of ordinary skill in the art] to combine the teachings of Anand with those of Cline. As discussed above, Anand teaches receiving a Downloadable over a network and associating it with a privilege request code. Anand, however, does not provide a discussion on how a requested domain is determined. [*Id.*] ¶ 237; [Ex. 1010,] 4. . . .
>
> Accordingly, a [person of ordinary skill in the art] would have turned to Cline to determine a requested domain for executable content. [Ex. 1017 ¶¶ 239–240]; [Ex. 1008], 11:44–63. One of ordinary skill in the art would have been motivated to combine these teachings for a number of reasons, including to protect client computers on a network (e.g., an Intranet) from downloaded/incoming viruses and malicious code without being language specific or requiring source code (which may not be available at all). [Ex. 1008], 11:44–63; [Ex. 1017] ¶ 241.

Pet. 53–54.

Patent Owner raises several arguments in response to Petitioner's contentions, including that Petitioner has not demonstrated that the proposed combination discloses "[means for] receiving [by an inspector] a Downloadable," as required by independent claims 1, 41, and 43; "[means for] generating [by the inspector] a first Downloadable security profile that identifies suspicious code in the received Downloadable," as required by each of the challenged claims; or "[means for] linking the first Downloadable security profile to the Downloadable before a web server

makes the Downloadable available to web clients," as required by each of the challenged claims. Prelim. Resp. 39–43.[4]

We agree with Patent Owner that Petitioner has failed to establish a reasonable likelihood that it would prevail on this ground. As an initial matter, we agree with Patent Owner's observation that Petitioner appears to have "simply copied and pasted" portions of its discussion of "its proposed combination of Cline with Apperson into this section and passed it off as a proposed combination with Anand." Prelim. Resp. 42. For example, Petitioner alleges that "[a]lthough Anand does not teach a 'content inspection engine' to 'generate' privilege request codes for a Downloadable, this feature is expressly taught by Cline." Pet. 53. As Patent Owner points out, however, Anand does not teach "privilege request codes" at all. Prelim. Resp. 42. And indeed, Cline does not either. Rather, "privilege request code" is a term used by Apperson. *See, e.g.*, Ex. 1007, Abst.

Similarly, Petitioner provides for the combination of Anand and Cline precisely and solely the same deficient rationale alleged previously as motivation for combining Apperson and Cline—as well as Apperson and Ji—namely, "to protect client computers on a network (e.g., an Intranet from downloaded/incoming viruses and malicious code without being language specific or requiring source code (which may not be available at all)." Pet. 54; *see also id.* at 35, 40. As explained in our discussion above with respect to Petitioner's proposed ground based on Apperson, Ji, and Cline, the portion of Cline cited by Petitioner to support this alleged motivation is

[4] Patent Owner also argues again that Cline is not analogous art to the '844 patent. Prelim. Resp. 39. For the reasons previously stated, on this record, we are not persuaded by that argument. *See supra* Section II.B.3.d.

unrelated to "protect[ing] client computers on a network," and is, accordingly, unavailing here.

Although the challenged claims require a DSP, rather than "privilege request codes," we are not persuaded on this record, in any event, that the combination of Anand and Cline teaches or suggests a DSP that identifies suspicious code, as required by each of the challenged claims. As Patent Owner points out (*id.* at 41), Petitioner appears to rely on the fact that there is overlap between the resources in Anand's requested domain and the "Example List of Operations Deemed Suspicious" in the '844 patent (*see* Pet. 52 (citing Ex. 1001, 4:20–34)), but fails to show that the requested domain identifies any suspicious code in a received Downloadable.

On this record, Petitioner has not identified sufficient evidence that the combination of the teachings of Anand and Cline teaches or suggests all of the limitations recited in independent claims 1, 15, 41, and 43 and, in particular, "generat[ing] . . . a first Downloadable security profile that identifies suspicious code in [a] Downloadable" and "linking . . . the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to web clients." Consequently, we are not persuaded that Petitioner demonstrates a reasonable likelihood that it would prevail at trial in showing that the subject matter of those claims or of dependent claims 7, 11, or 16 would have been obvious over the asserted combination.

III. CONCLUSION

On this record, we are not persuaded that Petitioner demonstrates a reasonable likelihood that it would prevail in showing the unpatentability of any of claims 1, 7, 11, 15, 16, 41, and 43 of the ’844 patent on the grounds asserted in the Petition. Consequently, the Petition is *denied* as to each of the asserted grounds.

IV. ORDER

Accordingly, it is:

ORDERED that the Petitioner is *denied*, and no *inter partes* review is instituted as to any of claims 1, 7, 11, 15, 16, 41, and 43 of the ’844 patent.

For PETITIONER:

Joseph J. Richetti
Daniel A. Crowe
BRYAN CAVE LLP
joe.richetti@bryancave.com
dacrowe@bryancave.com

For PATENT OWNER:

James Hannah
Jeffrey H. Price
KRAMER LEVIN NAFTALIS & FRANKEL LLP
jhannah@kramerlevin.com
jprice@kramerlevin.com

Michael Kim
FINJAN, INC.
mkim@finjan.com