Trials@uspto.gov
571-272-7822

Paper No. 8
Filed: 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 IPR2015-01893
Patent 7,613,926 B2

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

YANG, *Administrative Patent Judge*.

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

I. INTRODUCTION

Symantec Corp. ("Petitioner") filed a Petition for an *inter partes* review of claims 15, 18–20, 22, 25–27, and 30 of U.S. Patent No. 7,613,926 B2 (Ex. 1001, "the '926 patent"). Paper 1 ("Pet."). Finjan, Inc. ("Patent Owner") filed a Preliminary Response. Paper 6 ("Prelim. Resp.").

Based on this record, and for at least the reasons below, we determine Petitioner has not established a reasonable likelihood that it would prevail in showing the unpatentability of at least one challenged claim. *See* 35 U.S.C. § 314(a). Therefore, we *deny* the Petition for an *inter partes* review.

A. Related Proceedings

According to the parties, Patent Owner asserted the '926 patent against Petitioner in *Finjan, Inc. v. Symantec Corp.*, 3:14-cv-02998 (N.D. Cal.). Pet. 1; Paper 5, 1.

We previously denied another petition filed by Petitioner, seeking an *inter partes* review of the '926 patent on different grounds. *Symantec Corp. v. Finjan, Inc.*, IPR2015-01895, Paper 7. We also denied a petition filed by Sophos Inc., challenging certain claims of the '926 patent. *Sophos, Inc. v. Finjan, Inc.*, IPR2015-00907, Paper 8. We further denied Sophos' request for rehearing in that case. *Id.*, Paper 10. In addition, the '926 patent is the subject of IPR2016-00145, filed by Palo Alto Networks, Inc., which remains pending.

Further, the '926 patent is the subject of *Finjan, Inc. v. Sophos, Inc.*, 3:14-cv-01197 (N.D. Cal.), and *Finjan, Inc. v. Palo Alto Networks, Inc.*,

3:14-cv-04908 (N.D. Cal.). Paper 5, 1. Petitioner also has requested *inter partes* reviews of several patents related to the '926 patent. Pet. 1.

B. The '926 Patent

The '926 patent is directed to systems and methods to protect personal computers and other network accessible devices from "harmful, undesirable, suspicious or other 'malicious' operations that might otherwise be effectuated by remotely operable code." Ex. 1001, 2:27–31. The protection paradigm involves hashing an incoming Downloadable to derive a "Downloadable ID," which is used to reference security profile data for the incoming Downloadable in a database indexed according to Downloadable IDs. *Id.* at 2:27–4:49. The Downloadable security profile ("DSP") data for each Downloadable include "a list of suspicious computer operations that may be attempted by the Downloadable." *Id.* at 21:66–67. The Downloadable and a representation of the DSP data are sent to a destination computer. *Id.* at 22:1–4.

C. Illustrative Claim

Among the challenged claims, claims 15, 22, and 30 are independent. Claim 15 is illustrative and is reproduced below:

> 15. A computer-based method, comprising the steps of:
>
> receiving an incoming Downloadable;
>
> performing a hashing function on the incoming Downloadable to compute an incoming Downloadable ID;
>
> retrieving security profile data for the incoming Downloadable from a database of Downloadable security profiles indexed according to Downloadable IDs, based on the incoming Downloadable ID, the security profile data including a list of

> suspicious computer operations that may be attempted by the Downloadable; and
>
> transmitting the incoming Downloadable and a representation of the retrieved Downloadable security profile data to a destination computer, via a transport protocol transmission.

Claims 22 and 30 are directed to a system for managing Downloadables and a computer-readable storage medium, respectively. They recite limitations that are substantially similar to those in claim 15.

D. Asserted Grounds of Unpatentability

Petitioner asserts the following grounds, each of which challenges the patentability of claims 15, 18–20, 22, 25–27, and 30:

Basis	References
§ 103	Anand[1] and Dyson[2]
§ 103	Dan[3] and Hinsley[4]

In support of its patentability challenges, Petitioner relies on the Declaration of Jack W. Davidson, Ph.D. Ex. 1015.

---

[1] Rangachari Anand et al., *A Flexible Security Model for Using Internet Content*, IEEE Computer Society Proceedings of the Sixteenth Symposium on Reliable Distributed Systems, 1997 (Ex. 1003, "Anand").
[2] Patrick Dyson, U.S. Patent No. 5,050,212, issued Sept. 17, 1991 (Ex. 1010, "Dyson").
[3] Asit Dan et al., U.S. Patent No. 5,825,877, issued October 20, 1998 (Ex. 1008, "Dan").
[4] Stewart R. Hinsley et al., U.S. Patent No. 5,283,830, issued February 1, 1994 (Ex. 1009, "Hinsley").

II. ANALYSIS

A. Claim Construction

In an *inter partes* review, the Board interprets a claim term in an unexpired patent according to its broadest reasonable construction in light of the specification of the patent in which it appears. 37 C.F.R. § 42.100(b); *In re Cuozzo Speed Techs., LLC*, 778 F.3d 1271, 1278–81 (Fed. Cir. 2015), *cert. granted sub nom. Cuozzo Speed Techs. LLC v. Lee*, 136 S. Ct. 890 (2016).

The parties dispute the construction of the term "database." Pet. 13–15; Prelim. Resp. 6–9. Neither party, however, explains how the construction of this term 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. Obviousness over Anand and Dyson

Petitioner contends that the combination of Anand and Dyson renders the challenged claims obvious. Pet. 15–36. Petitioner argues that Anand teaches "virtually all of the limitations in the challenged claims." Pet. 17. The only limitation not taught in Anand, that is, using the hash of the Downloadable as an index to store and retrieve the DSPs from a database, is, according to Petitioner, taught by Dyson. *Id.* Based on the record before us, and for at least the following reasons, we are not persuaded.

Anand teaches "a system for downloading content from the Internet and controlling its actions on a client machine." Ex. 1003, 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.*

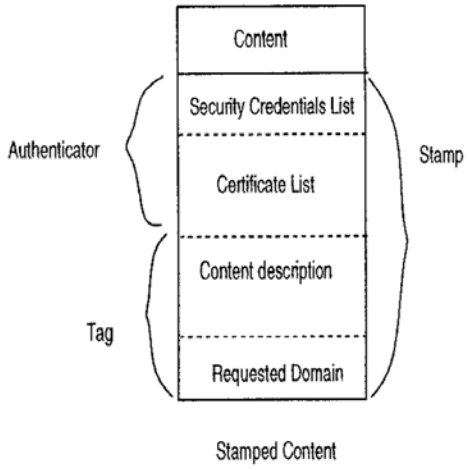

Figure 2. Structure of stamped content

Id. at 4.

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.*

Petitioner asserts the downloaded executable content in Anand, such as Java applets, Netscape plug-ins, and ActiveX controls, corresponds to the Downloadable in the challenged claims. *Id.* at 21. Petitioner also contends "content stamps and/or the requested domain" correspond to the DSPs (i.e., Downloadable security profiles) recited in the challenged claims. *Id.* at 27.

According to Petitioner, Anand teaches "security profile data including a list of suspicious computer operations that may be attempted by the Downloadable." *Id.* at 29–30 ("Anand teaches that the DSP (e.g., content stamp and/or requested domain) includes a list of suspicious operations that the Downloadable may attempt to invoke (e.g., system I/O operations)."). Petitioner also argues that Anand teaches "transmitting the incoming Downloadable and a representation of the retrieved Downloadable security profile data to a destination computer, via a transport protocol transmission." *Id.* at 30–32 ("Anand teaches that a representation of the Downloadable security profile (e.g., the content stamp including the

requested domain) is transmitted with the Downloadable (e.g., appended to or sent with the content)."). Patent Owner disputes both assertions. Prelim. Resp. 10–13, 18–19. We do not need to resolve these issues, however, because even if Anand does teach these limitations, we agree with Patent Owner that Petitioner has not shown the combination of Anand and Dyson teaches the limitation "retrieving security profile data for the incoming Downloadable from a database of Downloadable security profiles indexed according to Downloadable IDs, based on the incoming Downloadable ID." *See id.* at 13–17.

1. "Retrieving Security Profile Data for the Incoming Downloadable. . . Based on the Incoming Downloadable ID"

Anand states that "[t]he separate content hash permits stamps to be downloaded separately from the content itself, if necessary." Ex. 1003, 4. Based on this statement, Petitioner concludes that Anand teaches retrieving a content stamp (i.e., DSP) using a hash of the Downloadable (i.e., a Downloadable ID). Pet. 25. Patent Owner counters that "Anand merely discloses it is not necessary that the stamp actually be packaged together with the content." Prelim. Resp. 14. We find Patent Owner's argument more persuasive.

As an initial matter, we agree with Petitioner that Anand teaches the analysis module performing a hashing function on a Downloadable to compute a Downloadable ID, that is, the hashed value of the downloaded content. *See* Pet. 24; *see also* Ex. 1003, 4 ("The analysis module computes a hash of the downloaded content."). To the extent Petitioner also equates

"the content stamp field storing this value" with the Downloadable ID, however, we are not persuaded. *See* Pet. 24 (citing Ex. 1003, 6).

Each of the challenged claims requires "receiving *an* incoming Downloadable," and "performing a hashing function on *the* incoming Downloadable to compute an incoming Downloadable ID." Ex. 1001, 21:59–61, 22:21–23, 23:1–3 (emphases added). Here, "an incoming Downloadable" in the "receiving" step provides the antecedent basis for "the incoming Downloadable" in the "performing" step. As a result, "performing a hashing function" necessarily happens after "receiving an incoming Downloadable." *See Mformation Techs., Inc. v. Research in Motion Ltd.*, 764 F. 3d 1392, 1398 (Fed. Cir. 2014) (stating that a claim requires an ordering of steps when the claim language, as a matter of logic or grammar, requires so).

To be sure, the content stamp in Anand indeed includes a hash of the content. Ex. 1003, 3, 6. A stamp, however, is created by the manufacturers of the content and/or content rating services. *Id.* at 3, 6. As such, the hash in a stamp is computed before, not after, an incoming Downloadable is received, as the claim language requires. Thus, "the content stamp field storing [the content's hash] value" is not a Downloadable ID. *See* Pet. 24.

We now turn to the issue of whether Anand teaches "retrieving security profile data for the incoming Downloadable . . . based on the incoming Downloadable ID." In Anand, after receiving the stamped content, the downloading principal uses the public key of the manufacturer of the content or content rating service to verify that the stamp has not been modified. Ex. 1003, 4. Next, "[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.* Petitioner does not point to any persuasive evidence to show that the hash of the content is used to retrieve a stamp. In fact, as Patent Owner points out, "Anand does not state that if content is received without a stamp that it will attempt to retrieve a stamp, but rather that it is simply assumed to be from an untrusted source." Prelim. Resp. 16 (citing Ex. 1003, 4). Thus, we agree with Patent Owner that "Anand simply does not disclose that the content stamp (equated with the security profile data for the incoming Downloadable) is ever 'retrieved' at all, let alone retrieved based on 'the separate content hash' (equated with the Downloadable ID)." *Id.* at 14.

2. "A Database of Downloadable Security Profiles Indexed According to Downloadable IDs"

Petitioner concedes that "Anand does not expressly teach that the hash of the Downloadable is used as an index to store and retrieve the DSPs from a database." Pet. 17, *see also id.* at 27 ("Anand does not expressly teach that the DSPs are stored in and retrieved from a database indexed based on the Downloadable IDs."). According to Petitioner, however, Dyson teaches this feature. *Id.* at 17–18, 27–28.

Dyson teaches a method for verifying a file stored separately from a computer to be identical with a previous version of the file, before using the file. Ex. 1010, 1:48–51. Specifically, Dyson teaches generating identifiers based on the contents of each version of the file, and comparing the identifiers to verify the integrity of the later version file. *Id.* at 1:51–61.

Dyson teaches that "[p]referred methods of generating the identifier perform a 'hash' function." *Id.* at 3:13–14.

Petitioner contends that an ordinary artisan "would have recognized that storing and retrieving the content stamps (as taught by Anand) in a particular form of storage, i.e., a database, was a simple design choice." Pet. 28. We are not persuaded.

We follow "an expansive and flexible approach" in the obviousness analysis. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 415 (2007). Thus, a claimed invention may be obvious, in some instances, even when the prior art does not teach each claim limitation. Such an approach, however, does not allow us to fill a gap in prior art teachings with a panacea label "simple design choice." Instead, the party challenging a patent must either point to the record evidence or articulate persuasive argument to explain why one of skill in the art would modify the prior art to achieve the claimed invention. *Id.* at 418 ("[T]here must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness."). This, Petitioner fails to do.

Petitioner does not point to either Anand or Dyson for teaching storing content stamps in a database. Nor does Dr. Davidson provide any reasoning for doing so besides a conclusory statement parroting Petitioner's argument. *See* Ex. 1015 ¶ 130. Thus, we are not persuaded that an ordinary artisan would have had a reason to store the content stamps in "a database of Downloadable security profiles."

Furthermore, even assuming one of ordinary skill in the art would have stored the content stamps in a database, Petitioner has not sufficiently

accounted for indexing such a database "according to Downloadable IDs." According to Petitioner:

> In particular, based on the teachings in Dyson, it would have been obvious to store the content stamps in a database indexed according to the unique identifiers that are computed by hashing the executable content (as taught by both Dyson and Anand). As a result, the hash of a Downloadable would then be used to retrieve the corresponding content stamp (as taught by Anand) from the database.

Pet. 28. We, again, are not persuaded.

According to Dyson, "hash functions are known for providing a unique identifier of a piece of data," and "[i]t is known to use hash functions for building an index to items stored within a database." Ex. 1010, 3:27–30. We, however, agree with Patent Owner that Dyson does not teach using such a unique identifier to "index a completely separate data structure (i.e. using the 'Downloadable ID' to index 'Downloadable security profiles' in a database)." *See* Prelim. Resp. 17.

In sum, Petitioner does not point to sufficient evidence or present persuasive argument to show that Anand and Dyson, in view of the knowledge in the art, would have suggested "retrieving security profile data for the incoming Downloadable from a database of Downloadable security profiles indexed according to Downloadable IDs, based on the incoming Downloadable ID," as each challenged claim requires. As a result, we deny the Petition regarding the obviousness challenge of claims 15, 18–20, 22, 25–27, and 30 of the '926 patent over Anand and Dyson.

C. Obviousness over Dan and Hinsley

Petitioner also contends that the combination of Dan and Hinsley renders the challenged claims obvious. Pet. 36–51. Based on the record before us, and for at least the following reasons, we are not persuaded.

Dan teaches a form of authentication. Ex. 1008, 1:40–41. In Dan, a trusted third party, such as a certification agency, signs a certificate to identify the author of a program and to secure its integrity. *Id.* at 1:41–43. 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. *Id.* at 1:45–46.

Hinsley teaches that many computer systems control access to files by associating with each file a list of the users who are allowed to access the file and the types of access permitted to each. Ex. 1009, 1:14–19. According to Hinsley, this list is an example of an ACL. *Id.* at 1:19–20.

Petitioner equates Dan's program code with the Downloadable in the challenged claims and argues that a "hashing module" in Dan performs a hashing function on the Downloadable to compute a Downloadable ID. Pet. 39, 42. Petitioner also asserts that Dan's ACL corresponds to the DSP (i.e., Downloadable security profile) recited in the challenged claims. *Id.* at 43–44.

Petitioner concedes that "Dan does not expressly teach using this hash to retrieve an ACL (i.e., DSP) associated with the Downloadable." *Id.* at 44. According to Petitioner, however, this feature would have been obvious based on the teachings of Hinsley. *Id.* We are not persuaded.

Petitioner asserts that "Hinsley teaches that ACLs for a resource (e.g., a file) can be retrieved based upon a program identifier associated with a particular program (i.e., a Downloadable ID)." *Id.* (citing Ex. 1009, Abstract). But, as Patent Owner points out, "Hinsley only teaches that entry keys are used to select entries in an ACL, not to retrieve an ACL from a database." Prelim. Resp. 22; *see also* Ex. 1009, Abstract ("When a user attempts to access an object by way of a program, an entry in the ACL of the object is selected by matching the entry keys with at least the program identifier of the program.").

Further, Petitioner acknowledges that "[n]either Dan nor Hinsley expressly discloses using a hash as an index to retrieve an ACL (i.e., DSP) from a database." Pet. 45. Petitioner asserts, however, "[i]t was well-known in the art . . . that hashing functions were a very good way to index items in a database." *Id.* at 45–46. As support, Petitioner relies on the same teachings in Dyson it relies on in the obviousness ground based on Anand and Dyson. *Id.* (citing Ex. 1010, 3:11–35 ("[i]t is known to use hash functions for building an index to items stored within a database")). For the same reason as explained above—Dyson does not teach using a unique identifier from a data hash to "index a completely separate data structure (i.e. using the 'Downloadable ID' to index 'Downloadable security profiles' in a database)"—we, again, are not persuaded here. *See* Prelim. Resp. 23–24.

Petitioner asserts, and Patent Owner disputes, that Dan teaches (1) "security profile data including a list of suspicious computer operations that may be attempted by the Downloadable," and (2) "transmitting the

incoming Downloadable and a representation of the retrieved Downloadable security profile data to a destination computer, via a transport protocol transmission." Pet. 46–49; Prelim. Resp. 20–21, 24–25. We do not need to resolve these issues, however, because even if Dan does teach these limitations, we conclude that Petitioner has not shown the combination of Dan and Hinsley teaches the limitation "retrieving security profile data for the incoming Downloadable from a database of Downloadable security profiles indexed according to Downloadable IDs, based on the incoming Downloadable ID," as each challenged claim requires. As a result, we deny the Petition regarding the obviousness challenge of claims 15, 18–20, 22, 25–27, and 30 of the '926 patent over Dan and Hinsley.

III. CONCLUSION

Based on the record before us, and for at least the foregoing reasons, the information presented in the Petition and accompanying evidence does not establish a reasonable likelihood that Petitioner would prevail in showing the unpatentability of any one of claims 15, 18–20, 22, 25–27, and 30 of the '926 patent.

IV. ORDER

Accordingly, it is

ORDERED that Petitioner's request for an *inter partes* review of claims 15, 18–20, 22, 25–27, and 30 of the '926 patent is *denied*.

PETITIONER:
Joseph Richetti
Daniel Crowe
Bryan Cave LLP
joe.richetti@bryancave.com
dacrowe@bryancave.com

PATENT OWNER:
James Hannah
Jeffrey Price
Michael Kim
Kramer Levin Naftalis & Frankel LLP
jhannah@kramerlevin.com
jprice@kramerlevin.com
mkim@finjan.com