Trials@uspto.gov
571-272-7822

Paper No. 7
Entered: September 24, 2015

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SOPHOS, INC.,
Petitioner,

v.

FINJAN, INC.,
Patent Owner.

Case IPR2015-01022
Patent 8,677,494 B2

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

Sophos, Inc. ("Petitioner") filed a Petition (Paper 1, "Pet.") requesting *inter partes* review pursuant to 35 U.S.C. § 311 of claims 1, 10, 14, and 18 of U.S. Patent No. 8,677,494 B2 to Edery *et al.* (Ex. 1001, "the '494 patent"). Pet. 4. 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 '494 Patent

The '494 patent issued March 18, 2014, from U.S. Patent Application No. 13/290,708, filed November 7, 2011. The '494 patent also claims priority from nine earlier applications, of which the earliest-filed is U.S. Provisional Application No. 60/030,639, filed November 8, 1996 (Ex. 1005, "the '639 application"). Ex. 1001, [60], [63], col. 1, ll. 7–55.

The '494 patent describes protection systems and methods "capable of protecting a personal computer ('PC') or other persistently or even intermittently network accessible devices or processes from harmful, undesirable, suspicious or other 'malicious' operations that might otherwise be effectuated by remotely operable code." *Id.* at col. 2, ll. 51–56. "[R]emotely operable code that is protectable against can include," for

example, "downloadable application programs, Trojan horses and program code groupings, as well as software 'components', such as Java™ applets, ActiveX™ controls, JavaScript™/Visual Basic scripts, add-ins, etc., among others." *Id.* at ll. 59–64.

B. Related Proceedings

The '494 patent is the subject of a district court action, *Finjan, Inc. v. Sophos, Inc.*, 3:14-cv-01197 (N.D. Cal.), and has also been asserted in two other district court actions, *Finjan, Inc. v. Symantec Corp.*, 3:14-cv-02998 (N.D. Cal.), and *Finjan, Inc. v. Palo Alto Networks, Inc.*, 3:14-cv-04908 (N.D. Cal.). Pet. 2; Paper 5, 1. Petitioner also has filed a petition seeking *inter partes* review of a related patent, U.S. Patent No. 7,613,926 B2 to Edery *et al*. *Sophos, Inc. v. Finjan, Inc.*, Case IPR2015-00907, Paper 1.

C. Illustrative Claims

Of the challenged claims, claims 1 and 10 are independent. Each of challenged claims 14 and 18 depends directly from claim 10. Independent claims 1 and 10 are illustrative and are reproduced below:

1. A computer-based method, comprising the steps of:

receiving an incoming Downloadable;

deriving security profile data for the Downloadable, including a list of suspicious computer operations that may be attempted by the Downloadable; and

storing the Downloadable security profile data in a database.

10. A system for managing Downloadables, comprising:

a receiver for receiving an incoming Downloadable;

a Downloadable scanner coupled with said receiver, for deriving security profile data for the Downloadable, including a list of suspicious computer operations that may be attempted by the Downloadable; and

a database manager coupled with said Downloadable scanner, for storing the Downloadable security profile data in a database.

Ex. 1001, col. 21, ll. 19–25, col. 22, ll. 7–16.

D. References Relied Upon

Petitioner relies on the following references:

Exhibit	Reference
1006	ThunderBYTE Anti-Virus Utilities User Manual ("TBAV")
1008	Arnold, US 5,440,723, issued Aug. 8, 1995
1009	Ji, US 5,623,600, issued Apr. 22, 1997 (filed Sept. 26, 1995)
1010	Chen, US 5,951,698, issued Sept. 14, 1999 (filed Oct. 2, 1996)

Petitioner also relies on the Declaration of Dr. Paul C. Clark (Ex. 1002).

E. Asserted Grounds of Unpatentability

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

#	References	Basis	Claim(s) Challenged
1	TBAV and Ji	§ 103(a)	1, 10, 18
2	TBAV, Ji, and Chen	§ 103(a)	14
3	Arnold, Chen, and Ji	§ 103(a)	1, 10, 14, 18
4	Chen, Arnold, and Ji	§ 103(a)	1, 10, 14, 18

II. DISCUSSION

A. Claim Interpretation

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); Office Patent Trial Practice Guide, 77 Fed. Reg. 48,756, 48,766 (Aug. 14, 2012). *See also In re Cuozzo Speed Techs., LLC*, 793 F.3d 1268, 1278 (Fed. Cir. 2015) ("We conclude that Congress implicitly approved the broadest reasonable interpretation standard in enacting the AIA."). Under this standard, we interpret claim terms using "the broadest reasonable meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art, taking into account whatever enlightenment by way of definitions or otherwise that may be afforded by the written description contained in the applicant's specification." *In re Morris*, 127 F.3d 1048, 1054 (Fed. Cir. 1997). 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 four claim terms: "Downloadable," "suspicious program operations," "database," and "program script." Pet. 12–14. Patent Owner responds to each of Petitioner's

proposed constructions, offering alternative constructions for "Downloadable," "database," and "program script." Prelim. Resp. 8–16.

1. "Downloadable"

The term "Downloadable" is recited in each of the challenged claims. According to Petitioner, under the broadest reasonable interpretation, this term "should be understood to mean 'an executable application program which is downloaded from a source computer and can be run on the destination computer,' as defined in the '639 application." Pet. 12 (citing Ex. 1005, col. 2, ll. 1–4; Ex. 1002 ¶ 60). Petitioner further contends its proposed construction is "consistent with" and is "what one of ordinary skill in the art would understand from the specification of the '494 patent," and "is also consistent with what was agreed upon by Petitioner and Patent Owner" in the related district court proceedings. *Id.* at 12–13 (citing Ex. 1001, col. 9, ll. 46–52, Ex. 1002 ¶ 60, Ex. 1011, p. 4).

In response, Patent Owner contends that the proper construction of "Downloadable" is instead "an executable application program, which is downloaded from a source computer and run on the destination computer." Prelim. Resp. 8–9. Patent Owner points out that this is the definition provided in U.S. Patent Nos. 6,804,780 (Ex. 1014) and 6,092,194 (Ex. 1015), from which the '494 patent claims priority and which the '494 patent incorporates by reference, and is also the definition agreed to by the Petitioner in related litigation. *Id.* at 9 (citing Ex. 1001, col. 1, ll. 27–39; Ex. 1014, col. 1, ll. 50–53; Ex. 1015, col. 1, ll. 44–46; Ex. 2001, 2).

Although the broadest reasonable interpretation may differ from a construction agreed upon by the parties to a district court litigation, where claim construction is determined according to the different standard set forth

in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (*en banc*), we discern on this record no rationale for Petitioner's insertion of the phrase "can be" into the parties' previously agreed-upon construction. That is particularly the case in view of Petitioner's assertion (Pet. 12) that its proposed construction is "as defined in the '639 application," whereas the definition recited in the cited portion of the '639 application is instead identical to Patent Owner's proposed construction. *Compare* Ex. 1005, col. 2, ll. 1–4, *with* Prelim. Resp. 8–9. Indeed, Petitioner recognizes the actually recited definition elsewhere in the Petition. *See, e.g.*, Pet. 7 ("A Downloadable is described [in the '639 application] as 'an executable application program which is automatically downloaded from a source computer and run on the destination computer.'") (quoting Ex. 1005, col. 2, ll. 1–4).

We agree with and adopt substantially Patent Owner's proposed construction as the broadest reasonable interpretation of "Downloadable." Accordingly, on this record and for purposes of this Decision, we construe "Downloadable" to mean "an executable application program which is automatically downloaded from a source computer and run on a destination computer."

2. "suspicious program operations"

The term "suspicious computer operations" is recited in claims 1 and 10 of the '494 patent. Petitioner asserts that the broadest reasonable interpretation of this term is "computer instructions that are deemed to be potentially hostile." Pet. 13. According to Petitioner, this construction is consistent both "with the disclosure in the '639 application regarding 'potentially hostile operations' and 'suspect commands'" and "with the '494

patent, which describes 'harmful, undesirable, suspicious or other "malicious" operations that might otherwise be effectuated by remotely operable code.'" *Id.* (citing Ex. 1005, p. 8, l. 19–p. 9, l. 3, p. 15, l. 19–p. 16, l. 2; Ex. 1001, col. 2, ll. 54–56; Ex. 1002 ¶ 62).

Patent Owner responds that "the term . . . needs no construction and the plain and ordinary meaning within the context of the claims should apply." Prelim. Resp. 9. According to Patent Owner,

> Petitioner's proposed construction should also be rejected because computer instructions are not "computer operations that may be attempted by the Downloadable." An instruction is a low-level programmatical construct, while an operation is a high-level command that the program actually performs. This distinction is reinforced by the claim language, which recites "computer operations that may be attempted by the Downloadable" as well as the specification, which differentiates between "remotely operable code" and the "harmful, undesirable, suspicious or other 'malicious' operations that might otherwise be effectuated by remotely operable code." (Ex. 1001 at 2:54–64).

Id. at 11 (italics and boldface omitted).

We agree with Patent Owner that that this term requires no explicit construction, particularly not a construction that replaces the term "operations" with "instructions." Whereas a suspicious computer operation *might result from the execution* of instructions deemed to be potentially hostile, instructions are not operations. And indeed, as Patent Owner correctly points out (*id.* at 10), the portions of the '639 application and '494 patent cited by Petitioner do not mention the term "instruction." Moreover, we cannot discern how construing the phrase "suspicious computer operations" would add any clarity to the claim phrase itself in the context of

claims 1 and 10. Accordingly, we conclude that no explicit construction of "suspicious computer operations" is either warranted or necessary.

3. "database"

The term "database" is recited in claims 1 and 10 of the '494 patent. Patent Owner argues that, under the broadest reasonable interpretation, this term "should be understood to mean 'any structured store of data.'" Pet. 13. Petitioner contends that one of ordinary skill in the art would understand the term "database" to have this meaning, citing a claim construction order in an unrelated case, *Mangosoft, Inc. v. Oracle Corp.*, No. 02-545-SM (D.N.H.). *Id.* (citing Ex. 1020, 29). Petitioner also contends that its proposed construction is consistent with both the '639 application and the '494 patent. *Id.* at 13–14.

Patent Owner responds that the proper construction of "database" is instead "a collection of interrelated data organized according to a database schema to serve one or more applications." Prelim. Resp. 12. As Patent Owner points out (*id.*), this construction has been adopted by the district court in related litigation between the parties (*see Finjan, Inc. v. Sophos, Inc.*, No. 14-cv-01197 (N.D. Cal.), Claim Construction Order at 7 (Ex. 2003, 7)). Patent Owner contends that this "[t]his construction stays true to the claim language and most naturally aligns with the patent's description of the invention as well as the well-accepted definition of the term." Prelim. Resp. 12 (citing IBM DICTIONARY OF COMPUTING, 165 (10th ed. 1993) (Ex. 2002, 3)).

We agree with Patent Owner that the district court's construction in the related litigation between the parties represents the broadest reasonable construction of "database" in light of the claim language and the

specification of the '494 patent. *See Morris*, 127 F.3d at 1054; *see also Power Integrations, Inc. v. Lee*, ___ F.3d ____, 2015 WL 4757642, at *6 (Fed. Cir. Aug. 12, 2015) ("The fact that the board is not generally bound by a previous judicial interpretation of a disputed claim term does not mean . . . that it has no obligation to acknowledge that interpretation or to assess whether it is consistent with the broadest reasonable construction of the term."). As explained by the district court, the '494 patent does not define the term "database"; there is no evidence that Patent Owner disavowed the full scope of that term either in the Specification or during prosecution; and Patent Owner's definition appears to reflect both the context of the patent, as well as a well-accepted definition of the term. Ex. 2003, 5, 7.

Notably, in contrast, the court in the *Mangosoft* case cited by Petitioner did not construe the term "database," but rather construed "structured store of data" as "data that are organized in some recognized fashion (e.g., database files, word processing document files, or Web pages) and stored in the volatile and/or non-volatile memory of the various nodes participating in the shared memory system." *See* Ex. 1020, 23–29.

Accordingly, on this record and for purposes of this Decision, we construe "database" to mean "a collection of interrelated data organized according to a database schema to serve one or more applications."

4. Other Claim Terms

For purposes of this Decision, no other claim terms require express interpretation. *Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011) ("claim terms need only be construed 'to the extent necessary to resolve the controversy'" (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

B. Asserted Grounds of Unpatentability

1. Obviousness over TBAV and Ji

Petitioner contends that TBAV, either alone or in combination with Ji, would have rendered obvious the subject matter of claims 1, 10, and 18 of the '494 patent. Pet. 23–29. For the reasons that follow, we are not persuaded that Petitioner has established a reasonable likelihood that it would prevail on this ground with respect to any of the challenged claims.

a. TBAV

TBAV is a user manual for a set of software programs for protecting computer systems against viruses and for recovering those systems from any viruses that slip through. Ex. 1006, 6. TBAV describes two virus-scanning programs, "TbScan" and "TbScanX," as well as various utility programs for restoration of infected boot sectors and partition tables ("TbUtil"), reconstruction and removal of infected files ("TbClean," "TbDel"), definition of new virus signatures ("TbGenSig"), and other computer-security measures (e.g., "TbMem," "TbFile," "TbDisk"). Ex. 1006, 6–10.

TbScan, in particular, is described as including both signature scanning functionality, for detecting known viruses whose signatures are stored in a signature file, and heuristic scanning functionality, for disassembling and analyzing files to detect suspicious instruction sequences and yet-unknown viruses. *Id.* at 6–7, 52, 158–160. According to TBAV, heuristic scanning allows TbScan to look into a file's contents and interpret program instructions to detect their purpose. *Id.* at 160. TbScan assigns a "heuristic flag" and a score to instruction sequences known to be common in viruses but uncommon in "normal" programs. *Id.* By adding the scores associated with the flags, TbScan informs the user whether a file might be,

or probably is, infected by an unknown virus. *Id.* TbScan also provides the option to output a list of infected program files, heuristic flags, and file pathnames either to a printer or to a log file. *Id.* at 65. According to TBAV, TbScan can be used to scan files stored on disks, fixed drives, and/or network drives, and may be run either upon user request or automatically (e.g., upon startup). *Id.* at 52–53, 55–56.

TbScanX is described by TBAV as "the memory resident version of TbScan." *Id.* at 7. According to TBAV, "[t]his signature scanner remains resident in memory and automatically scans those files that are being executed, copied, de-archived, downloaded, etc." *Id.* Although at one point TBAV also states that "TbScanX is virtually identical to TbScan, with one important difference: TbScan is memory-resident" (*id.* at 89), it appears that TbScanX lacks TbScan's heuristic scanning capability, particularly in view of the above-quoted description of the program as a "signature scanner" (*id.* at 7).

b. Ji

Ji describes a system for detecting and eliminating viruses on a computer network, where 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, Abstract.

c. Discussion

Petitioner relies on TBAV alone for all elements of claims 1, 10, and 18, with the exception of "a receiver for receiving an incoming Downloadable" recited in claim 10. Pet. 24–29. Petitioner contends that that receiver "is at least implicitly taught by TBAV," but asserts further that, "[t]o the extent TBAV does not explicitly describe a receiver, . . . it would

have been obvious to one of ordinary skill in the art to apply the teachings of a receiver in Ji to the system of TBAV." *Id.* at 26–27.

In response to Petitioner's contentions, Patent Owner argues, *inter alia*, that the combination of TBAV and Ji fails to disclose "deriving security profile data for the Downloadable, including a list of suspicious computer operations that may be attempted by the Downloadable," and "storing the Downloadable security profile data in a database," as recited in independent claims 1 and 10. Prelim. Resp. 23–32.

In view of Patent Owner's arguments, we are not persuaded that Petitioner demonstrates a reasonable likelihood that it would prevail in showing that any of claims 1, 10, and 18 are unpatentable over TBAV and Ji.

With respect to the recited "deriving" step, Petitioner asserts that TBAV's TbScan "performs heuristic analysis of files," which heuristic analysis "includes detecting suspicious instruction sequences within a file and applying heuristic flags to the file." Pet. 24. According to Petitioner, "[t]he heuristic flags indicate the suspicious instructions." *Id.* Although heuristic flags reasonably could be termed "security profile data for [a] Downloadable," claims 1 and 10 both require, more particularly, "deriving security profile data . . . *including a list of suspicious computer operations that may be attempted by the Downloadable*." Ex. 1001, claims 1, 10 (emphasis added). Petitioner does not explain, nor can we discern, how TBAV's heuristic analysis discloses "deriving . . . a list of suspicious computer operations that may be attempted by the Downloadable." As we explain in our interpretation of the term "suspicious computer operations" in Section II.A.2, *supra*, Petitioner dos not persuade us that "instructions" are

themselves "operations." Further, notwithstanding Petitioner's conclusory assertion that "the list of heuristic flags in the file is a list of suspicious instructions" (Pet. 24), Petitioner does not identify where TBAV discloses such a "list of heuristic flags in the file."

Additionally, the evidence cited by Petitioner does not demonstrate that TBAV and Ji teach or suggest storing security profile data in a "database," as that term is properly construed. Neither the "log file" to which Petitioner asserts "TBAV teaches that heuristic analysis results . . . can be output" nor the "TBSCAN.SIG file" to which Petitioner asserts TBAV's TbGenSig program adds virus signatures (Pet. 25) is disclosed to be a database, and Petitioner provides no persuasive evidence in support of its assertion that "[a] person of ordinary skill in the art would understand either or both of these files could be is [sic] a database containing that contains one or more data entries" (*id.* at 25–26). Indeed, although there is a page missing from the copy of TBAV provided by Petitioner, TBAV does not appear to disclose that the log file has any particular organization or serves any other applications, which, as explained in Section II.A.3, *supra*, are among the hallmarks of a database. Instead, the log file appears instead to be a simple output of "infected program files, specifying heuristic flags . . . and complete pathnames" either to a printer or to a file that is, by default, overwritten by the results of each new scan by the TbScan program. *See* Ex. 1006, 65–66; *see also* Ex. 2003, 7 ("The practical import of adopting the IEEE definition or the IBM definition cited by the parties is largely the same, as both definitions appear to exclude 'log files.' . . . Therefore, I find that a log file does not qualify as a database in the context of this patent.").

On this record, Petitioner has not identified sufficient evidence that TBAV and Ji disclose either "deriving security profile data for the Downloadable, including a list of suspicious computer operations that may be attempted by the Downloadable," or "storing the Downloadable security profile data in a database," as recited in independent claims 1 and 10. Consequently, we are not persuaded that Petitioner demonstrates a reasonable likelihood that it would prevail at trial in showing that either of those claims or dependent claim 18 would have been obvious over the combination of those references.

2. Obviousness over TBAV, Ji, and Chen

Claim 14 depends from claim 10 and further recites the limitation "wherein the Downloadable includes program script." Ex. 1001, claim 14. Petitioner contends that TBAV, either alone or in combination with Ji and/or Chen, would have rendered the subject matter of claim 14 obvious. Pet. 34–36. For essentially the same reasons as set forth in our discussion of asserted ground 1 in Section II.B.1, *supra*, we are not persuaded that Petitioner has established a reasonable likelihood that it would prevail on this ground.

a. Chen

Chen describes methods and systems for detecting and removing viruses from macros. Ex. 1010, Abstract. In one embodiment, Chen discloses computer system 100 that includes central processing unit (CPU) 104, memory 106, communications unit 112 for facilitating communication with other systems, and various other computer components. *Id.* at col. 4, ll. 42–59. CPU 104, as directed by instructions received from memory 106, provides signals for accessing computer files, determining whether they include macros, locating the macros, scanning the macros to determine

whether viruses are present, and taking corrective action when viruses are detected. *Id.* at col. 5, ll. 3–9. In a preferred embodiment, memory 106 includes macro virus detection module 206, which in turn includes macro locating and decoding module 302, macro virus scanning module 304, macro treating module 306, virus information module 308, file correcting module 310, and data buffer 312. *Id.* at col. 5, ll. 10–14, col. 5, l. 64–col. 6, l. 9.

According to Chen, files may be targeted for access by user selection or based upon triggering events such as the opening of certain application file, system boots, or at periodic intervals, preferably prior to launch of an application program that may cause operation of a macro virus. *Id.* at col. 6, ll. 10–30. Macro locating and decoding module 302 examines targeted files to determine, among other things, whether they include embedded macros. *Id.* at col. 6, ll. 38–41, col. 12, ll. 4–65. If a macro is present, the macro is located and decoded into binary code and stored, so that it can be scanned for viruses. *Id.* at col. 12, ll. 54–57. Macro virus scanning module 304 includes routines to detect combinations of suspect instructions likely to be used by macro viruses, such as the combination of a macro enablement instruction, which allows the formatting of a file to be set to indicate that the file includes a macro for execution, and a macro reproduction instruction, which allows the macro virus to be replicated. *Id.* at col. 8, ll. 40–53.

To identify suspect instruction combinations, macro virus scanning module 304 accesses comparison data from virus information module 308, including sets of instruction identifiers that are used to identify combinations of suspect instructions in the decoded macro. *Id.* at col. 8, ll. 58–63, col. 13, ll. 7–32. If it is determined that a macro includes a combination of suspect

instructions defined and identified by the set of instruction identifiers, the macro is deemed to be infected by an unknown virus corresponding to that set of data, and macro virus scanning module 304 flags the decoded macro as infected and stores information associating the decoded macro to the set of instruction identifiers that resulted in a positive unknown virus detection in data buffer 312 so that other modules such as macro treating module 306 can treat the infected macro accordingly. *Id.* at col. 8, l. 67–col. 9, l. 11.

b. Discussion

As explained in Section II.B.1, we are not persuaded by Petitioner's argument and evidence that TBAV or TBAV and Ji teach or suggest "deriving security profile data for the Downloadable, including a list of suspicious computer operations that may be attempted by the Downloadable," or "storing the Downloadable security profile data in a database," as recited in independent claim 10, from which claim 14 depends. Petitioner relies on Chen in connection with this asserted ground only for Chen's alleged disclosure of the claim limitation "wherein the Downloadable includes program script," as recited in claim 14 (*see* Pet. 34–36), and does not argue persuasively that Chen remedies the deficiencies of TBAV or TBAV and Ji with respect to the elements of claim 10. Accordingly, we also are not persuaded that Petitioner demonstrates a reasonable likelihood that it would prevail at trial in showing that claim 14 is unpatentable over the asserted combinations of TBAV, Ji, and Chen.

On this record, Petitioner has not identified sufficient evidence that TBAV, Ji, and Chen teach or suggest either "deriving security profile data for the Downloadable, including a list of suspicious computer operations that may be attempted by the Downloadable," or "storing the Downloadable

security profile data in a database," as recited in independent 10. Consequently, we are not persuaded that Petitioner demonstrates a reasonable likelihood that it would prevail at trial in showing that dependent claim 14 would have been obvious over the asserted combinations of those references.

3. Obviousness over Arnold, Chen, and Ji

Petitioner contends that Arnold, either alone or in combination with Chen and Ji, would have rendered obvious the subject matter of claims 1, 10, 14, and 18 of the '494 patent. Pet. 39–44. For the reasons that follow, we are not persuaded that Petitioner has established a reasonable likelihood that it would prevail on this ground with respect to any of the challenged claims.

a. Arnold

Arnold describes "methods and apparatus for providing computational integrity for digital data processors and networks," including, *inter alia*, periodically monitoring a data processing system for anomalous behavior that may indicate the presence of an undesirable software entity such as a computer virus, worm, or Trojan Horse; scanning for occurrences of known types of undesirable software entities and taking remedial action if any are discovered; capturing samples of unknown types of viruses; identifying machine code portions of the captured samples; extracting an identifying signature from executable code portions and adding the signature to a signature database; and informing neighboring data processing systems on a network of an occurrence of the undesirable software entity. Ex. 1008, Abstract, col. 1, ll. 15–18, col. 4, ll. 29–56.

Figure 8 of Arnold is reproduced below.

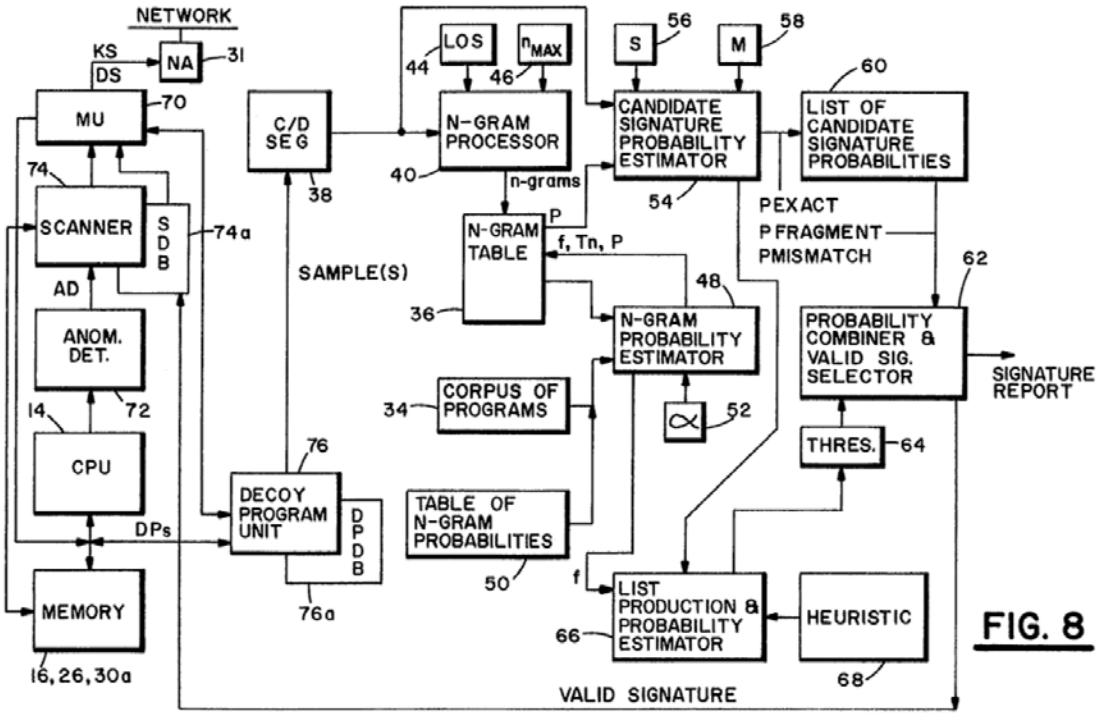

FIG. 8

Figure 8 is a block diagram of a system operable for executing Arnold's method. *Id.* at col. 3, ll. 44–45. If preliminary evidence of virus-like activity is detected, computational resources are devoted to obtaining more conclusive evidence of infection. *Id.* at col. 5, ll. 29–32. With reference to Figure 8, anomaly detector 72 detects anomalous behavior of CPU 14. *Id.* at col. 28, ll. 47–49. Upon detection of anomalous behavior, scanner 74 compares valid signatures from a signature database (SDB 74a) to programs stored in the memory to identify known viruses. *Id.* at col. 28, ll. 56–60. If no known virus is found, the anomaly may be due to an unknown virus, and a decoy program is deployed by decoy program unit 76 to obtain a sample of the unknown virus. *Id.* at col. 6, ll. 3–7, col. 29, ll. 1–17. If a modification to the decoy program is detected, decoy program unit 76 isolates the undesirable software entity and provides one or more samples

to code/data segregator 38 (also termed "invariant code identifier 38"). *Id.* at col. 29, ll. 10–14. Portions of the virus that are likely to vary from one instance of the virus are then filtered out, code-data segregation is performed to separate non-executable "data" portions from "code" portions representing machine instructions, and a viral signature is extracted from "probably-invariant" portions of the code. *Id.* at col. 7, ll. 11–21, col. 7, l. 59–col. 8, l. 6, col. 9, ll. 13–16. For each virus, one or more candidate signatures having the best "scores" are selected to represent the virus. *Id.* at col. 19, ll. 15–20. In particular, invariant code identifier 38 identifies candidate signatures and provides them to n-gram processor 40 as candidate signatures. *Id.* at col. 29, ll. 14–17. N-gram processor 40 processes the candidate signatures, and, if a valid signature for the unknown undesirable software identity is found, the valid signature is stored in signature database 74a. *Id.* at ll. 18–22.

b. Discussion

In response to Petitioner's contentions, Patent Owner argues, *inter alia*, that Arnold in view of Chen and Ji fails to disclose "deriving security profile data for the Downloadable, including a list of suspicious computer operations that may be attempted by the Downloadable" and "storing the Downloadable security profile data in a database," as recited in independent claims 1 and 10. Prelim. Resp. 38–46.

In view of Patent Owner's arguments, we are not persuaded that Petitioner demonstrates a reasonable likelihood that it would prevail at trial in showing that any of claims 1, 10, 14, and 18 are unpatentable over Arnold, alone or in view of Chen and Ji.

With respect specifically to the recited "deriving" step, Petitioner cites Arnold as teaching "deriving a signature for a virus by analyzing an invariant section of code from a program" (Pet. 40 (citing Ex. 1008, col. 10, l. 63–col. 11, l. 20, col. 19, ll. 15–20; Ex. 1002 ¶ 128)) and argues:

> To the extent Arnold does not explicitly describe deriving a list of suspicious computer operations, it would have been obvious to one of ordinary skill in the art that the signature generated by Arnold is indicative of a sequence of suspicious computer operations because it represents an invariant portion of code that caused observed anomalous behavior. Ex. 1008, col. 6, ll. 3–7; col. 7, ll. 11–21; Ex. 1002 ¶ 130.

Pet. 40–41. Whether or not Petitioner is correct that "it would have been obvious to one of ordinary skill in the art that the signature generated by Arnold is indicative of a sequence of suspicious computer operations," a proposition for which Petitioner provides no support other than a portion of Dr. Clark's declaration that parrots verbatim Petitioner's argument, claims 1 and 10 specifically require "deriving security profile data . . . including a list of suspicious computer operations *that may be attempted by the Downloadable*." Ex. 1001, claims 1, 10 (emphasis added). Petitioner does not identify, nor can we discern in the cited portions of Arnold, any teaching or suggestion that one might be able to derive a list of suspicious operations that may be attempted by a program from the invariant sections of code identified by Arnold's system. Indeed, as Patent Owner explains (Prelim. Resp. 40), Arnold appears to be unaware of the operations that may be attempted by a program. Arnold does not describe performing any analysis of the identified sections of code to determine their functions, but instead compares them with "a large corpus of typical programs" to identify potential virus signatures that are "statistically unlikely to generate false

positives when run on uninfected, legitimate programs, but will always identify the virus if it is present." Prelim. Resp. 40–41 (quoting Ex. 1008, col. 9, ll. 16–19, 42–68).

Alternatively, Petitioner relies upon Chen for this element, arguing that "Chen teaches comparing decoded binary code with instruction identifiers to reveal the presence of code portions corresponding to the instructions of the instruction identifiers" (Pet. 40 (citing Ex. 1010, col. 14, ll. 13–64; Ex. 1002 ¶ 129) and that:

> Chen teaches comparing instruction identif[i]ers to decoded binary code. The instruction identif[i]ers include unique binary code portions known to correspond to suspect instructions. The presence of the unique binary code portions within the decoded binary code reveals the presence of the suspicious instructions within the file. Ex. 1010, col. 14, ll. 13-31; Ex. 1002 ¶ 130. Arnold and Chen are both directed to virus scanning software that can detect unidentified viruses within a file. Thus, at minimum, it would have been obvious to one of ordinary skill in the art to apply the teachings of Chen of identifying suspicious computer operations in a file to the system of Arnold. Ex. 1002 ¶ 130.

Pet. 41. As Patent Owner points out, however, Chen fails to cure the other deficiencies of Arnold. Prelim. Resp. 42. We agree with Patent Owner that, notwithstanding Petitioner's statement that "Chen teaches comparing decoded binary code with instruction identifiers to reveal the presence of code portions corresponding to the instructions of the instruction identifiers" (Pet. 40), Petitioner provides "no explanation of how this assertion corresponds to the claim language, which recites 'deriving security profile data'" (Prelim. Resp. 42–43).

We also agree with Patent Owner that the Petition fails to "articulate a 'rational underpinning to support the legal conclusion of obviousness' for

the proposed combination of *Arnold* and *Chen*." Prelim. Resp. 42 (quoting *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007)). An invention "composed of several elements 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. A petitioner also must show that there was a reason to combine those elements to achieve the claimed invention with a reasonable expectation of success. *See PAR Pharm., Inc. v. TWi Pharms., Inc.*, 773 F.3d 1186, 1193 (Fed. Cir. 2014). As Patent Owner points out (Prelim. Resp. 42), the only reason provided by Petitioner for combining Arnold with Chen in the manner proposed is that they "are both directed to virus scanning software that can detect unidentified viruses within a file" (Pet. 41). The mere fact that the references are directed broadly to similar subject matter, without any further articulated reason that would have prompted a person of ordinary skill in the art to combine the limitations taught by the references, does not constitute persuasive evidence that such a combination would have been obvious. *See KSR*, 550 U.S. at 418.

On this record, Petitioner has not identified sufficient evidence that Arnold, Chen, and Ji teach or suggest "deriving security profile data for the Downloadable, including a list of suspicious computer operations that may be attempted by the Downloadable," as recited in independent claims 1 and 10, and further, has not articulated a rational underpinning to support the legal conclusion of obviousness for the proposed combination advanced in the Petition. Consequently, we are not persuaded that Petitioner demonstrates a reasonable likelihood that it would prevail in showing that the challenged claims would have been obvious over that combination.

4. Obviousness over Chen, Arnold, and Ji

Petitioner contends that Chen, either alone or in combination with Arnold and Ji, would have rendered obvious the subject matter of claims 1, 10, 14, and 18 of the '494 patent. Pet. 44–50. Petitioner again relies for the "deriving" steps of claims 1 and 10 on column 14, lines 13–64, of Chen and makes the same assertion as for ground 3 that "Chen teaches comparing decided binary code with instruction identifiers to reveal the presence of code portions corresponding to the instructions of the instruction identifiers." *Id.* at 46, 48. Because Petitioner again provides no explanation of how this assertion corresponds to the claim language, this asserted ground over Chen in view of Arnold and Ji is no more persuasive than the asserted ground over Arnold in view of Chen and Ji, despite the change in order and additional citations of columns 4–6 and 15 of Chen for certain other elements of claims 1 and 10. *See In re Bush*, 296 F.2d 491, 496 (CCPA 1961) ("where a rejection is predicated on two references each containing pertinent disclosure which has been pointed out to the applicant, we deem it to be of no significance, but merely a matter of exposition, that the rejection is stated to be on A in view of B instead of on B in view of A, or to term one reference primary and the other secondary."). Nor are Petitioner's allegations regarding motivation to combine Chen with Arnold and Ji any more persuasive that its allegations regarding the alleged motivation to combine Arnold with Ji and Chen. *See* Pet. 45–50 (citing Ex. 1002 ¶¶ 147, 151, 156, 160, 165); *see also* Section II.B.3, *supra*.

Accordingly, we are not persuaded that Petitioner demonstrates a reasonable likelihood that it would prevail in showing that any of claims 1,

10, 14, and 18 would have been rendered obvious over Chen alone or in combination with Arnold, and Ji.

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, 10, 14, and 18 of the '494 patent on the grounds asserted in the Petition. Consequently, the Petition is *denied* as to each of the asserted grounds.

IV. ORDER

Upon consideration of the record before us, it is, therefore,

ORDERED that the Petitioner is *denied*, and no *inter partes* review is instituted as to any claim of the '494 patent.

For PETITIONER:

James Heintz
Nicholas J. Panno
DLA PIPER (US) LLP
Sophos-Finjan-494IPR@dlapiper.com

For PATENT OWNER:

James Hannah
Michael H. Lee
Paul J. Andre
KRAMER LEVIN NAFTALIS & FRANKEL LLP
jhannah@kramerlevin.com
mhlee@kramerlevin.com
pandre@kramerlevin.com