Trials@uspto.gov
571-272-7822

Paper 8
Entered: August 13, 2015

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

FORTINET, INC.,
Petitioner,

v.

SOPHOS INC.,
Patent Owner.

Case IPR2015-00618
Patent 8,261,344 B2

Before BRYAN F. MOORE, PETER P. CHEN, and
MICHELLE N. WORMMEESTER, *Administrative Patent Judges.*

MOORE, *Administrative Patent Judge.*

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

I. INTRODUCTION

A. Background

Fortinet, Inc. ("Petitioner") filed a Petition (Paper 1, "Pet.") requesting an *inter partes* review of claims 1, 2, 10, 13, 16, and 17 of U.S. Patent No. 8,261,344 B2 (Ex. 1001, "the '344 patent"). Sophos Inc. ("Patent Owner") did not file a Preliminary Response. We have jurisdiction under 35 U.S.C. § 314(a), 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."

Upon consideration of the Petition, and for the reasons explained below, we determine that Petitioner has shown that there is a reasonable likelihood that it would prevail with respect to at least one of the challenged claims. We institute an *inter partes* review of claims 1, 2, 10, 13, 16, and 17 of the '344 patent.

B. Related Proceedings

The parties identify the following case involving the '344 patent: *Fortinet, Inc. v. Sophos Inc.*, Case No. 3:13-cv-005831-LB (N.D. Cal.). Pet. 5. Patent Owner further identifies two pending requests for *inter partes* review involving patents commonly owned with the '344 patent: IPR2015-00617 and IPR2015-00619. Paper 4.

C. The '344 Patent

The '344 patent is directed to classifying software (e.g., malicious software) based in part on "identifying one or more genes that appear in

software." *See, e.g.*, Ex. 1001, 2:31–50, 6:43–45. The '344 patent distinguishes its solution from other malware detection techniques, including: "[t]raditional malware protection techniques . . . based around anti-virus vendors creating signatures for known malware and products that scan systems searching for those specific signatures," *id.* at 1:59–61; "[h]ueristic techniques [that] look at various properties of a file and not necessarily the functionality of the program," *id.* at 2:10–13; and techniques that "run[] malware and attempt[] to stop execution if malicious behavior is observe[d] to happen," *id.* at 2:13–15.

The '344 patent's approach uses what it calls "genes" to classify software and detect potential malware. *Id.* at Abstract. The '344 patent proposes the following steps: (1) extracting "functional blocks" from a software file, *see, e.g.*, Ex. 1001, 5:17–28, 7:62–63; (2) searching the "functional blocks" for one or more "genes," *id.* at 5:29–45, 7:64–67; and (3) matching the identified "genes" against a "library" which includes "a number of classifications based on groupings of genes," *id.* at 5:46–50, 7:60–61, 8:1–4. Software is then classified based on this matching. *Id.*

D. The Challenged Claims

Petitioner challenges claims 1, 2, 10, 13, 16, and 17 of the '344 patent. Independent claims 1 and 16 are illustrative of the claimed subject matter and are reproduced below:

> 1. A method for classifying software, said method comprising;
>
> providing a library of gene information including a number of classifications based on groupings of genes;

identifying at least one functional block and at least one property of the software;

identifying one or more genes each describing one or more of the at least one functional block and the at least one property of the software as a sequence of APIs and strings;

matching the one or more genes against one or more of the number of classifications using a processor;

classifying the software based on the matching to provide a classification for the software; and

notifying a user of the classification of the software.

Ex. 1001, 7:57–8:5.

16. A method for generating software classifications for use in classifying software, said method comprising:

providing a library of gene information including a number of classifications based on groupings of genes;

identifying one or more genes each describing a functionality or a property of the software as a sequence of APIs and strings;

combining a plurality of genes that describe the software, thereby providing a set of genes;

testing the set of genes for false-positives against one or more reference files using a processor;

defining the software classification based on the set of genes; and

storing the set of genes and the software classification in the library.

Ex. 1001, 8:40–54.

E. Prior Art Relied Upon

Petitioner relies upon the following prior art references:

Bodorin	US 7,913,305 B2	filed Jan. 30, 2004	(Ex. 1003)
Kissel	US 7,373,664 B2	filed Dec. 16, 2002	(Ex. 1004)
Chen	US 5,591,698	issued Sept. 14, 1999	(Ex. 1005)
Apap	US 7,448,084 B1	filed Jan. 27, 2003	(Ex. 1006)

F. Asserted Grounds of Unpatentability

Petitioner asserts the following grounds of unpatentability:

Challenged Claims	Basis	Reference(s)
1 and 2	§ 102	Bodorin
10 and 13	§ 103	Bodorin
1, 2, 10, and 13	§ 102	Kissel
16 and 17	§ 103	Kissel and Apap
16 and 17	§ 103	Bodorin, Kissel, and Apap
1, 2, 10, and 13	§ 102	Chen

II. ANALYSIS

A. Claim Construction

We construe claims in an unexpired patent by applying the broadest reasonable interpretation in light of the specification. *See* 37 C.F.R. § 42.100(b); *In re Cuozzo Speed Techs., LLC*, No. 2014-1301, 2015 WL 4097949, *7–8 (Fed. Cir. July 8, 2015). Under the broadest reasonable construction standard, claim terms are given their ordinary and customary

meaning, as would be understood by one of ordinary skill in the art in the context of the entire disclosure. *See In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007). Grammatical principles may inform the plain-language meaning of a claim. *See SuperGuide Corp. v. DirecTV Enters., Inc.*, 358 F.3d 870, 885–87 (Fed. Cir. 2004). On the other hand, a "claim term will not receive its ordinary meaning if the patentee acted as his own lexicographer" and clearly set forth a definition of the claim term in the specification. *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1366 (Fed. Cir. 2002) ("*CCS*").

In light of Petitioner's challenges to the patentability of the claims, we address one term in the challenged claims. Other terms in the challenged claims need no express construction at this time.

Claim 1 recites "identifying one or more genes each describing one or more of the at least one functional block and the at least one property of the software as a sequence of APIs and strings." Ex. 1001, 7:64–67. Petitioner contends that this limitation means that "a piece of functionality or property of a program," based on Patent Owner's construction in the district court. Pet. 16 (citing Ex. 1008, 32). We are not persuaded by Petitioner's contention in this regard.

"Gene" does not appear to be an ordinary term or a term that is known in art, rather it appears to be a term used by the inventor to represent a term trademarked by the assignee of the patent. Ex. 1001, 3:58–63. Thus, although we are using the broadest reasonable interpretation, the patentee is his own lexicographer. *CCS*, 288 F.3d at 1366. The written description of the '344 patent defines "gene" by stating "[a] gene is piece of functionality or property of a program. Each piece of functionality is described using

sequences of APIs [application programming interfaces] and strings, which can be matched against functional blocks." Ex. 1001, 5:32–35. We read these sentences together as stating that a gene takes the form of APIs and strings. Petitioner states that

> during the prosecution of the '344 Patent, the Patent Owner made it clear that the '344 Patent was patentable over the cited prior art because the '344 Patent disclosed that "[t]he genes are matched against functional blocks of code and the functionality is determined by matching sequences of API calls and the strings that are references."

Pet. 1 (quoting Ex. 1002, 128). Finally, in the related district court case, the court defined "gene" as "a piece of functionality or property of a program." Ex. 3001, 18. The district court noted, and we agree, that the use of the language "APIs and strings" is redundant because it is later cited in the claims. *Id.* at 20.[1] We note also that "APIs and strings" is still a limitation of the claims despite being omitted from the definition of "gene."

We also note that the proposed construction reads "functionality or property," although the claim limitation reads "identifying one or more genes each describing one or more of the at least one functional block and the at least one property of the software." The issue becomes whether the claim requires gene to describe both a functional block and a property.

The '344 patent Specification describes functional block and property with an "or." Additionally, the Specification assigns no special meaning or different meaning to property as opposed to functional block. In fact property is mentioned only once (other than in the claims or a restating of

[1] We note that the table in Ex. 3001, at page 18, contains an error in the court's construction of "gene." The bottom of page 20 properly states the court's construction of "gene."

the claims). On the record before us, consistent with the Specification, we construe "identifying one or more genes each describing one or more of the at least one functional block and the at least one property of the software" to require that either a functional block *or* a property, but not necessarily both, must be described by a gene in the prior art to meet that claim limitation.

B. Principles of Law

Anticipation requires the disclosure in a single prior art reference of each and every element of the claimed invention, arranged as in the claim. *Lindemann Maschinenfabrik GmbH v. Am. Hoist & Derrick Co.*, 730 F.2d 1452, 1458 (Fed. Cir. 1984). A 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 ordinary skill in the art; and (4) objective evidence of nonobviousness. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966).

In that regard, an obviousness analysis "need not seek out precise teachings directed to the specific subject matter of the challenged claim, for a court can take account of the inferences and creative steps that a person of ordinary skill in the art would employ." *KSR*, 550 U.S. at 418; *see also Translogic*, 504 F.3d at 1259. A prima facie case of obviousness is established when the prior art itself would appear to have suggested the

claimed subject matter to a person of ordinary skill in the art. *In re Rinehart*, 531 F.2d 1048, 1051 (CCPA 1976).

The level of ordinary skill in the art is reflected by the prior art of record. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001); *In re GPAC Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995); *In re Oelrich*, 579 F.2d 86, 91 (CCPA 1978).

C. Anticipation of Claims by Bodorin

Petitioner asserts that claims 1 and 2 are unpatentable under 35 U.S.C. § 102(e) as anticipated by Bodorin. Pet. 7–8. To support its contentions, Petitioner provides detailed explanations as to how the prior art meets each claim limitation. *Id.* at 18–59. Petitioner also relies upon a Declaration of Dr. Seth Nielson, who has been retained as an expert witness by Petitioner for the instant proceeding. Ex. 1007.

Bodorin discloses a malware detection system that determines whether an executable code module is malware according to behaviors exhibited by the code module while executing. Ex. 1003, Abstract. Figure 2 of Bodorin is reproduced below.

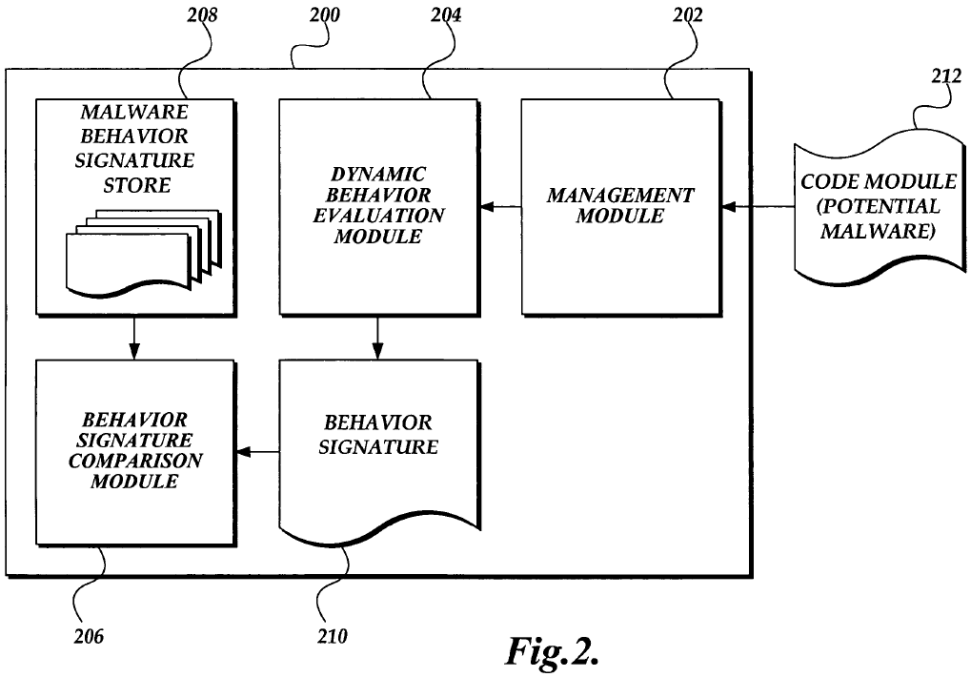

Fig.2.

Figure 2, above, shows a diagram of the malware detection system of Bodorin. Ex. 1003, 3:43–45. As shown in Figure 2, management module 202 first evaluates code module 212, which potentially may be malware, to determine the appropriate type of dynamic behavior evaluation module needed. *Id.* at 4:26–29. Once determined, code module 212 is executed inside dynamic behavior evaluation module 204, while dynamic behavior evaluation module 204 records behaviors, including interesting behaviors that are potentially associated with malware, exhibited by code module 212. *Id.* at 4:39–58. Each dynamic behavior evaluation module represents a virtual environment in which code module 212 runs. If code module 212 is malware, then the destructive behaviors of code module 212 are confined to the virtual environment. *Id.* at 4:39–50. Interesting behaviors include those shown in Table A, reproduced below.

TABLE A

RegisterServiceProcess ( )	ExitProcess ( )
Sleep ( )	GetCommandLine ( )
WinExec (application_name)	CreateProcessA (proces_name, parameters_list)
InternetOpenUrlA (URL_name)	GlobalAlloc ( )
InternetOpenA (URL_name)	GetTickCount ( )
IntenetClosetHandle ( )	CopyFileA (new_name, existing_name)
CreateFileA (file_name)	GetWindowsDirectoryA( )
ReadFile ( )	GetSystemDirectoryA ( )
WriteFile ( )	GetModuleFileNameA ( )
Close Handle ( )	LoadLibraryA (library_name)
RegOpenKeyA (key_name)	GetProcAddress (procedure_name)
RegSetValueA (subkey_name)	FindFirstFileA (file_specificator)
RegSetValuExA (subkey_name)	FindNextFileA ( )
RegCloseKey ( )	FindClose ( )
UrlDownloadToFileA (url_name, file_name)	

As shown in Table A above, the interesting behaviors include API calls such as RegisterServiceProcess() and strings such as application_name.

The present record supports the contention that Bodorin discloses maintaining a gene library and identifying genes that are APIs from functional blocks of code. Pet. 19–27; Ex. 1007 (multiple paragraphs and figures cited in Petition). Petitioner relies on Bodorin's identification of "interesting behaviors," which include APIs, from code modules to meet the limitation of "identifying at least one functional block and at least one property of the software" and "identifying one or more genes each describing one or more of the at least one functional block and the at least one property of the software as a sequence of APIs and strings," as recited in claim 1. Pet. 23–24; Ex. 1007 ¶¶ 62–63.

Petitioner relies on Bodorin's "malware behavior signature store" to meet the limitation of "providing a library of gene information including a number of classifications based on groupings of genes," as recited in claim 1. Pet. 22–23; Ex. 1007 ¶ 64. Petitioner relies on Bodorin's "behavior signature comparison module" to meet the limitation of "matching the one or more genes against one or more of the number of classifications using a

processor," as recited in claim 1. Pet. 25–26; Ex. 1007 ¶ 64. For claims 1 and 2, Petitioner relies on Bodorin's reporting code as malware to meet the limitations of "classifying the software [as malware (only in claim 2)] based on the matching to provide a classification for the software and notifying a user of the classification of the software," as recited in claims 1 and 2. Pet. 26–27; Ex. 1007 ¶ 65.

We have reviewed the proposed ground of anticipation by Bodorin against claims 1 and 2, and we are persuaded, at this juncture of the proceeding, that Petitioner has established a reasonable likelihood that Petitioner would prevail in its challenge to claims 1 and 2 on this ground.

D. Obviousness over Bodorin

Petitioner asserts that claims 10 and 13 are unpatentable under 35 U.S.C. § 103(a) as obvious over Bodorin. Pet. 27. To support its contentions, Petitioner provides detailed explanations as to how the prior art meets each claim limitation. *Id.* at 27–29. Petitioner also relies upon a Declaration of Dr. Nielson. Ex. 1007.

Claims 10 and 13 depend directly from claims 1. Claim 10 recites removing malware and claim 13 recites blocking malware.

Petitioner relies on Bodorin's description of the prior art to meet the limitation of removing or blocking malware. Pet. 28–30; Ex. 1007 ¶¶ 66–69. Petitioner concludes that one of ordinary skill in the art would have been guided by statements in Bodorin itself to combine Bodorin with existing systems that blocked or removed malware. Pet. 27. For example, Bodorin states "while the malware detection system may be used as a stand-alone product, it may also be used in conjunction with current anti-virus software." Ex. 1003, 4:2–4; *see* Pet. 27.

We have reviewed the proposed ground of obviousness over Bodorin against claims 10 and 13 and we are persuaded, at this juncture of the proceeding, that Petitioner has established a reasonable likelihood that Petitioner would prevail in its challenge to claims 10 and 13 on this ground.

E. Obviousness over Bodorin, Kissel and Apap

Petitioner asserts that claims 16 and 17 are unpatentable under 35 U.S.C. § 103(a) as obvious over the combination of Bodorin, Kissel and Apap. Pet. 43–49. To support its contentions, Petitioner provides detailed explanations as to how the prior art meets each claim limitation. *Id.* Petitioner also relies upon a Declaration of Dr. Nielson. Ex. 1007.

Kissel discloses a system for detecting the presence of malicious computer code in emails. Ex. 1004, 1:44–46. Features in an email are extracted and checked against a library. *Id.* at 5:47–60. If the extracted features do not match those in the library, then additional processing is done to determine if the feature under examination crosses a certain threshold. Ex. 1004, 6:7–8:62. If the feature exceeds the threshold, then the email is deemed to be malicious and the library is updated. *Id.* The threshold levels are calibrated by testing for false-positives. *Id.* at 8:58–62.

Apap discloses a method for detecting intrusions in the operation of a computer system. Ex. 1006, Abstract. The system disclosed by Apap first generates a model for normal behavior by a computer system. *Id.* at 5:10–20. An algorithm for detecting anomalous behavior is then employed, and the algorithm is trained and tested by comparing the detected anomalous behavior against normal behavior to generate statistics related to detection and false positives. *Id.* at 15:44–16:20.

Claim 17 depends from independent claim 16. Claim 16 recites a several limitations similar to the limitations discussed in regard to claim 1. Thus, we will not repeat that discussion here. Claim 16 also recites "testing the set of genes for false-positives against one or more reference files using a processor." Claim 17 recites that the classifications of software include malware, which is essentially the same as the limitation of claim 2 discussed above. Thus, we will not repeat that discussion here.

The present record supports the contention that the combination of Bodorin, Kissel, and Apap discloses testing the set of genes for false-positives against one or more reference files using a processor as required by claims 16 and 17. Pet. 43–49; Ex. 1007 ¶¶ 83–89.

Petitioner relies on Bodorin's use of thresholds and scores to define code as malicious code (Ex. 1003, 6:7–28), Kissel's use of false-positive thresholds and parameters to define code as malicious code (Ex. 1004, 8:47–62), and Apap's use of a (malicious code) detection rate and a false-positive rate (Ex. 1006, 15:44–51) together to meet the limitations "testing the set of genes for false-positives against one or more reference files using a processor," as recited in claim 16. Pet. 47–49. Petitioner states:

> It would have been obvious to a person of ordinary skill in the art to combine Bodorin with Kissel and Apap. There was explicit teaching, suggestion, and motivation in the prior art references themselves to combine. Claim 16 discloses a method for testing a generated set of genes for false-positives and storing the set of genes. Kissel discloses this method as a way of updating the library. Ex. 1004 at 6:7[–]8:62. Although not explicitly disclosed in Kissel, Apap discloses testing for false-positives using one or more reference files. Ex. 1006 at 15:44[–]51. Although Bodorin does not explicitly disclose a method for updating its malware behavior signature store, Bodorin nevertheless explicitly states that it is desirable to

> update the store. Specifically, Bodorin states "[i]t is anticipated that the malware behavior signature store 208 will be periodically updated with behavior signatures of known malware. The malware behavior signature store 208 should be especially updated as the more rare, 'new' malware is discovered." Ex. 1003 at 6:1[–]5. Accordingly, one of ordinary skill in the art reading this passage in Bodorin would be motivated to combine the disclosures of Bodorin with a method for updating the malware behavior signature store, which is exactly what is disclosed in Kissel and Apap.

Pet. 44; *see* Ex. 1007 ¶ 93. Petitioner also accounts for all of the limitations of claims 16 and 17.

We have reviewed the proposed ground of obviousness over Bodorin, Kissel, and Apap against claims 16 and 17, and we are persuaded, at this juncture of the proceeding, that Petitioner has established a reasonable likelihood that Petitioner would prevail in its challenge to claims 16 and 17 on this ground.

F. Remaining Grounds Challenging the Claims of the '344 Patent

Pursuant to 35 U.S.C. § 316(b), rules for *inter partes* proceedings were promulgated to take into account the "regulation on the economy, the integrity of the patent system, the efficient administration of the Office, and the ability of the Office to timely complete proceedings." The promulgated rules provide that they are to "be construed to secure the just, speedy, and inexpensive resolution of every proceeding." 37 C.F.R. § 42.1(b). As a result, and in determining whether to institute an *inter partes* review of a patent, the Board, in its discretion, may "deny some or all grounds for unpatentability for some or all of the challenged claims." 37 C.F.R. § 42.108(b).

We have reviewed Petitioner's assertion regarding the non-redundancy of the remaining grounds, that "Patent Owner may attempt to distinguish the challenged claims by arguing the '344 taught away from executing suspected malware software to observe behaviors." Pet. 29–30, 50–51. We find that the distinctions asserted by Petitioner are not based on limitations of the claim or claim construction differences. We are not persuaded that teaching away in the challenged patent, as opposed to the prior art, is relevant to the analysis. Based on the record before us, we exercise our discretion and decline to institute review of any claim of the '344 patent based on the remaining challenges in the Petition. *See* 37 C.F.R. § 42.108(a).

III. CONCLUSION

For the foregoing reasons, we are persuaded that Petitioner has demonstrated that there is a reasonable likelihood that Petitioner would prevail in showing that claims 1, 2, 10, 13, 16, and 17 of the '344 patent are unpatentable. The Board, however, has not made a final determination with respect to the patentability of the challenged claims.

IV. ORDER

Accordingly, it is

ORDERED that pursuant to 35 U.S.C. § 314(a), an *inter partes* review is hereby instituted for the following grounds of unpatentability:

Challenged Claims	Basis	Reference(s)
1 and 2	§ 102	Bodorin
10 and 13	§ 103	Bodorin
16 and 17	§ 103	Bodorin, Kissel, and Apap

FURTHER ORDERED that no other ground of unpatentability asserted in the Petition is authorized for this *inter partes* review; and

FURTHER ORDERED that pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is hereby given of the institution of a trial; the trial will commence on the entry date of this decision.

PETITIONER:

Jason Liu
Jared Newton
QUINN EMANUEL URQUHART & SULLIVAN, LLP
jasonliu@quinnemanuel.com
jarednewton@quinnemanuel.com

PATENT OWNER:

James M. Heintz
Gianni Minutoli
Nicholas Panno
DLA PIPER LLP (US)
jim.heintz@dlapiper.com
SophosFortinetIPR@dlapiper.com
nicholas.panno@dlapiper.com