

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SOPHOS LTD. & SOPHOS INC.,

Petitioners

v.

OPEN TEXT INC.

Patent Owner

Case IPR2023-00633

Patent No. 8,726,389

**DECLARATION OF PROFESSOR NENAD MEDVIDOVIC, Ph.D.
IN SUPPORT OF PATENT OWNER'S PRELIMINARY RESPONSE**

TABLE OF CONTENTS

I.	QUALIFICATIONS	2
II.	UNDERSTANDING OF RELEVANT LEGAL PRINCIPLES.....	7
III.	STATE OF THE PRIOR ART	9
IV.	The '389 Patent	11
A.	Summary Of The '389 Patent	11
B.	Claim Construction	17
C.	Level Of Ordinary Skill In The Art	18
V.	OVERVIEW OF PRIOR ART	20
A.	Overview Of Halperin.....	20
VI.	HALPERIN DOES NOT RENDER OBVIOUS THE CHALLENGED CLAIMS.....	27
A.	Independent Claims 1, 15, and 29.....	28
1.	[1B]: “ <i>information including at least an identity of an object initiating the event</i> ”	29
2.	[1C]: “ <i>receiving data about the computer object from a second remote computer...including at least an identity of an object initiating the event</i> ”	42
3.	[1E]: “ <i>correlating...data about the computer object received from the first remote computer to...data about the computer object received from the second remote computer</i> ”	49
4.	[1F]: “ <i>comparing...the correlated data...to other objects or entities to identify relationships between the correlated data and the other objects or entities</i> ”	52
5.	[1G]: “ <i>classifying...the computer object as malware on the basis of said comparison</i> ”	56
B.	Dependent Claims	63
VII.	CLAIMS AND/OR LIMITATIONS NOT ADDRESSED IN THIS DECLARATION	64
VIII.	SECONDARY OBJECTIVE INDICIA OF NONOBVIOUSNESS	64

IX.	CONCLUSION.....	65
-----	-----------------	----

EXHIBIT LIST

Exhibit	Description
2001	Declaration of Professor Nenad Medvidovic, Ph.D.
2002	<i>Curriculum Vitae</i> of Nenad Medvidovic, Ph.D.
2003	<i>Webroot, Inc. and Open Text Inc., v. Sophos LTD</i> , Case No. 6:22-cv-240-ADA-DTG, Complaint (W.D. Tex Mar. 4, 2022)
2004	<i>Sonrai Memory Ltd. v. Kingston Tech. Co. and Kingston Tech. Corp.</i> , Case No. 6:21-cv-1284-ADA, Order Denying Kingston Tech. Co. and Kingston Tech. Corp.’s Opposed Motion to Stay Pending Resolution of <i>Inter Partes</i> Review, Dkt. 94 (W.D. Tex., Oct. 18, 2022)
2005	<i>mCom IP, LLC v. Cisco Systems, Inc.</i> , Case No. 6:22-cv-00261-ADA, Order Denying Defendant’s Motion to Stay Pending <i>Inter Partes</i> Review, Dkt. 42 (W.D. Tex., Oct. 20, 2022)
2006	<i>Intellectual Ventures I LLC et al. v. Hewlett Packard Enterprise Co.</i> , No. 6:21-cv-00596-ADA, Order Denying Hewlett Packard Enterprise Company’s Opposed Motion to Stay Pending Resolution of <i>Inter Partes</i> Review, Dkt. 104 (W.D. Tex., Dec. 22, 2022)
2007	Dani Kass, “Catching Up On Patent Litigation With Judge Albright” (Law360 Mar. 14, 2023); https://www.law360.com/pulse/articles/1582438/print?section=pulse/courts
2008	U.S. District Courts – Combined Civil and Criminal Federal Court Management Statistics (Mar. 31, 2023), <i>available at</i> https://www.uscourts.gov/statistics/table/na/federal-court-management-statistics/2023/03/31-1
2009	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab. et al.</i> , Case No. 6:22-cv-243-ADA-DTG, Order Granting Fourth Amended Scheduling Order, Dkt. 160 (W.D. Tex. Jan. 22, 2023)
2010	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab. et al.</i> , Case No. 6:22-cv-243-ADA-DTG, Claim Construction Order, Dkt. 236 (W.D. Tex. Mar. 16, 2023)
2011	U.S. Patent Application Publication No. 2002/0194490 (Halperin ’490)
2012	<i>Webroot, Inc. and Open Text Inc., v. Sophos LTD</i> , Case No. 6:22-cv-240-ADA-DTG, Scheduling Order, Dkt. 54 (W.D. Tex. Aug. 8, 2022)
2013	<i>Webroot, Inc. and Open Text Inc., v. Forcepoint LLC</i> , Case No. 6:22-cv-00342-ADA-DTG, Defendant’s Preliminary Invalidity Contentions (W.D. Tex., Sept. 13, 2022) (excerpted)

Exhibit	Description
2014	<i>Webroot, Inc. and Open Text Inc., v. Sophos Ltd.</i> , Case No. 6:22-cv-00240-ADA-DTG, Defendant Sophos Ltd.’s Preliminary Invalidity Contentions (W.D. Tex., Sept. 13, 2022) (excerpted)
2015	<i>Webroot, Inc. and Open Text Inc., v. Trend Micro, Inc.</i> , Case No. 6:22-cv-00239-ADA-DTG, Defendant’s Preliminary Invalidity Contentions (W.D. Tex., Sept. 13, 2022) (excerpted)
2016	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab. et al.</i> , Case No. 6:22-cv-243-ADA-DTG, Defendant’s Preliminary Invalidity Contentions Appendix A (W.D. Tex., Sept. 14, 2022)
2017	<i>Webroot, Inc. and Open Text Inc., v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.</i> , Case No. 6:22-cv-00241-ADA, Defendant’s Preliminary Invalidity Contentions (W.D. Tex., Sept. 13, 2022) (excerpted)
2018	U.S. Patent 8,726,389 Certified File History
2019	July 13, 2022 Email from Jennifer Inghram of King & Spalding
2020	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab. et al.</i> , Case No. 6:22-cv-243-ADA-DTG, Joint Defendants’ Opening Claim Construction Brief, Dkt. 86 (W.D. Tex., Oct. 28, 2022)

I, Professor Nenad Medvidovic, Ph.D., declare as follows:

1. I have been engaged by King & Spalding LLP on behalf of OpenText Inc. (“Patent Owner”), to provide this Declaration concerning the technical subject matter relevant to the petition for Inter Partes Review (“Petition”) of U.S. Patent 8,726,389 (“the ’389 patent”).

2. I am over 18 years of age. I have personal knowledge of the facts stated in this Declaration and could testify competently to them if asked to do so.

3. I have reviewed and am familiar with the specification and the claims of the ’389 patent. In general, I will cite to the specification of a United States patent using the following formats: (Exhibit No. Col:Line Number(s)) or (Exhibit No. Paragraph Number(s)). For example, the citation (EX-1001 1:1–10) points to the ’389 patent specification at column 1, lines 1–10. Also, for convenience, I use italics to denote limitations from the challenged claims.

4. All the opinions in this Declaration are based on the documents I reviewed and my knowledge and professional judgment. In forming the opinions expressed in this Declaration, I reviewed the Petition for *Inter Partes* Review of U.S. Patent No. 8,726,389 and supporting exhibits.

5. I have been asked to provide my technical opinions regarding how a person of ordinary skill in the art would have understood the claims of the ’389 patent at the time of the alleged invention. I have also been asked to provide my

technical opinions on how concepts in the '389 patent specification relate to claim limitations of the '389 patent. In reaching the opinions provided herein, I have considered the '389 patent, its prosecution history, and the references cited above and have drawn as appropriate on my own education, training, research, knowledge, and personal and professional experience.

I. QUALIFICATIONS

6. In formulating my opinions, I have relied on my knowledge, training, and experience in the relevant field, which I will summarize briefly. A more detailed summary of my background, experience, and publications is set forth in my curriculum vitae (CV), which is submitted as Exhibit 2002.

7. I received a Bachelor of Science degree, Summa Cum Laude, from Arizona State University's Computer Science and Engineering department.

8. I received a Master of Science degree from the University of California at Irvine's Information and Computer Science department.

9. I received a Doctor of Philosophy degree from the University of California at Irvine's Information and Computer Science department. My dissertation was entitled "Architecture-Based Specification-Time Software Evolution."

10. For more than 25 years, my research has focused on the building blocks that comprise software systems: software components (computational and data

elements) and connectors (interaction-facilitating elements), and mechanisms for their implementation, primarily middleware facilities and software frameworks. In particular, my Ph.D. dissertation—whose first research results were published in 1995 while the dissertation was completed in 1999—yielded a new design style for the development and evolution of highly distributed software systems built out of independent components that exchange different types of information encoded in messages in order to accomplish specific goals. This research focused on the design principles behind such systems, the appropriate support for modeling the systems (both at the level of individual agents as well as their integrations into subsystems and entire systems), and the infrastructure for their implementation, deployment, and optimization. This work was applied across a wide range of domains, from traditional distributed systems to mobile devices and autonomous robots.

11. I am employed by the University of Southern California (“USC”) as a faculty member in the Computer Science Department and have been since January 1999. I currently hold the title of Professor with tenure. Since July 2022, I have served as my department’s Chairman. Between January 2009 and January 2013, I served as the Director of the Center for Systems and Software Engineering at USC. Between July 2011 and July 2015, I served as my Department’s Associate Chair for Ph.D. Affairs. Between July 2021 and June 2022, I served as my Department’s Vice-Chair for Tenure Affairs.

12. I teach graduate and undergraduate courses in Software Architecture, Software Engineering, and Embedded Systems, and I advise Ph.D. students. I have graduated 19 Ph.D. students and advise 7 students currently pursuing a Ph.D.

13. I served as Program Co-Chair for the flagship conference in my field—International Conference on Software Engineering (“ICSE”)—held in May 2011. I have served as Chair or Co-Chair for various other conferences in the Software Engineering field, including: the Fifth Working Institute of Electrical and Electronics Engineers (“IEEE”)/IFIP Conference on Software Architecture, the Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems, the Fifteenth International Association for Computing Machinery’s (“ACM”) Special Interest Group on Software Engineering (“SIGSOFT”) Symposium on Component Based Software Engineering, the IEEE/CSSE/ISE Workshop on Software Architecture Challenges for the 21st Century, and the Doctoral Symposia at the Sixteenth ACM SIGSOFT International Symposium on the Foundations of Software Engineering and at the 44th International Conference on Software Engineering.

14. I serve or have served as an editor of several peer-reviewed journals, including: “IEEE Transactions on Software Engineering,” “ACM Transactions on Software Engineering and Methodology,” “Journal of Software Engineering for Robotics,” “Elsevier Information and Software Technology Journal,” “Journal of

Systems and Software,” “Journal of Software Engineering Research and Development,” and “Springer Computing Journal.” Additionally, I have served as a guest editor of several special issues for different journals. Since January 2018, I have served as Editor-in-Chief of “IEEE Transactions on Software Engineering.”

15. Between September 2013 and September 2015, I served as Chair of the ICSE Steering Committee. Both prior to that (2008-2013) and subsequent to that (2015-present) I have been a member of the ICSE Steering committee. I am currently a member of the Steering Committee of the joint European Conference on Software Engineering and the Symposium on the Foundations of Software Engineering (“ESEC/FSE”). From July 2015 to September 2017, I chaired the Steering Committee of FSE, while it was a standalone symposium. I previously served as a member of the Steering Committees of the Working IEEE/IFIP Conference on Software Architecture.

16. Between July 2015 and June 2018, I served as Chair of the ACM SIGSOFT, the largest professional organization in my field of work. During the past year, I have chaired the Committee in charge of ACM’s Software System Award, one of the most important technical awards given in my field.

17. I co-authored “Software Architecture: Foundations, Theory, and Practice,” a widely used textbook in the field of Software Systems’ Architecture.

18. I have served as editor of various books in the Software Engineering field including: “Proceedings of the 3rd International Conference on Self-Adaptive and Self-Organizing Systems,” “Proceedings of the Warm-Up Workshop for the 32nd International Conference on Software Engineering,” “Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture,” and “40 Editions of ICSE – The Ruby Anniversary Celebration.”

19. I have authored or co-authored nearly 300 papers in the Software Engineering field. My most cited paper has been cited well over 3,000 times. A paper I co-authored in the 1998 International Conference on Software Engineering, my field’s flagship conference, was given ten years later, in 2008, that conference’s Most Influential Paper Award. Another one of my publications received a Most Influential Paper Award in 2020. Several of my other publications have received “best paper” or “most cited” awards.”

20. Several of my publications have focused on different aspects of computer security. A paper describing a security-vulnerability detection tool for Android devices, built in my lab, won the Best Tool Paper award at the 2017 Automated Software Engineering Conference. I have co-authored three patents in the area of cybersecurity.

21. I have been named a Distinguished Scientist of the ACM. I have been elected a Fellow of the IEEE, IEEE's highest grade that is granted to less than 0.1% of its membership annually.

II. UNDERSTANDING OF RELEVANT LEGAL PRINCIPLES

22. I am not a lawyer, and I will not provide any legal opinions. Although I am not a lawyer, I have been advised certain legal standards are to be applied by technical experts in forming opinions regarding the meaning and validity of patent claims.

23. I understand that a patent claim is invalid if it is anticipated or obvious in view of the prior art, and that a claim can be unpatentable even if all of the requirements of the claim cannot be found in a single prior-art reference. I further understand that invalidity of a claim requires that the claim be anticipated or obvious from the perspective of a person of ordinary skill in the art at the time the invention was made.

24. I have been informed that a patent claim is invalid if it would have been obvious to a person of ordinary skill in the art. In analyzing the obviousness of a claim, I understand the following factors may be taken into account: (1) the scope and content of the prior art; (2) the differences between the prior art and the claims; (3) the level of ordinary skill in the art; and (4) any so called "secondary considerations" of nonobviousness if they are present.

25. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. I understand that to prove that prior art or a combination of prior art renders a patent obvious, it is necessary to:

- 1) Identify the particular references that, singly or in combination, make the patent obvious;
- 2) Specifically identify which elements of the patent claim appear in each of the asserted references; and
- 3) Explain why a person of ordinary skill in the art would have combined the references, and how they would have done so, to create the inventions claimed in the patent. I further understand that exemplary rationales that may support a conclusion of obviousness include:
 - Combining prior art elements according to known methods to yield predictable results;
 - Simple substitution of known element for another to obtain predictable results;
 - Use of known technique(s) to improve similar devices (methods or products) in the same way;
 - Applying a known technique to a known device (method or product) ready for improvement to yield predictable results;
 - “Obvious to try” – choosing from a finite number of identified, predictable solutions with a reasonable expectation of success;
 - Known work in one field of endeavor may prompt variations of the work for use in either the same field or a different field based on design

- incentives or other market forces if the variations are predicable to a person of ordinary skill in the art; and
- Some teaching, suggestion, or motivation in the prior art that would have led a person of ordinary skill in the art to modify the prior art reference or to combine the prior art reference teachings to arrive at the claimed invention.

26. I have been informed that, in considering obviousness, hindsight reasoning derived from the patent-at-issue may not be used.

27. I have been informed that a prior art reference may be relied upon not only for what it expressly teaches, but also for what is inherent therein—i.e., what necessarily flows from the express teachings. I further understand that the mere fact that a certain thing may result from a given set of circumstances is not sufficient.

III. STATE OF THE PRIOR ART

28. “Malware” generally refers to an executable computer file or an object that is or contains malicious code, such as viruses, trojans, worms, and the like. (EX-1001 1:19-22.) Different versions of malware can function in different manners, and what works as a detection for one version of malware (*e.g.*, a virus) may be less effective against another version (*e.g.*, a worm). (*See generally* EX-1001 1:19-3:3.)

29. At the time of invention of the ’389 patent, many malware systems focused on looking for previously known malware, *e.g.*, by examining objects (*i.e.*, any executable computer file) for known signatures indicative of malware. (*E.g.*,

EX-1001 1:23-34.) “Generally, the method of dealing with malware is that when new types of malware are released, for example via the Internet, these are eventually detected.” (EX-1001 1:27-29.) Providers of anti-virus software push an update to their software intended to recognize and handle known malware. (EX-1001 1:29-33.) However, these types of systems suffered from inherent delay. (EX-1001 1:37-40.) A system would be vulnerable after release of a new piece of malware until such malware was detected and a solution generated and released to users. (EX-1001 1:40-42.)

30. Other anti-malware systems sought to protect from previously unknown malware. For example, local area networks were known to implement a centralized computer responsible for authorizing the running of files on remote computers within the network. (EX-1001 1:65-2:2.) When a remote computer encountered a file that was unknown to it, it would forward a request to the central computer. (EX-1001 2:2-5.) When the file was unknown even to the central computer, the central computer would obtain the whole file from the remote computer in order to analyze it to determine whether it should be regarded as safe or malware. (EX-1001 2:9-13.) But such an analysis had to be “carried out manually or semi-manually” by subjecting the file to a detailed analysis. (EX-1001 2:14-17.)

31. These analysis methods included, for example, emulation or interpretation, “which can still take days given the human involvement that is

typically required.” (EX-1001 2:14-17.) Other prior art systems similarly relied on a human system administrator to “make a better informed though still ‘manual’ decision as to whether or not the file is safe to run.” (*E.g.*, EX-1001 2:24-33.) “There is therefore still a considerable period of time before a new file is classified as safe or as malware.” (EX-1001 2:18-19.)

32. Thus, prior art systems “rel[ie]d on deep analysis of a new object in order to determine whether or not the object is malicious, which introduces delay and therefore risk to users during the period that the file is analysed and new anti-malware signatures distributed[.]” (EX-1001 2:63-67.)

33. Accordingly, there was a long-felt need for an improved technique to efficiently determine whether objects were safe or malware.

IV. THE ’389 PATENT

A. Summary Of The ’389 Patent

34. The ’389 Patent is entitled “Methods and Apparatus for Dealing with Malware” and describes and claims a method and system for classifying a computer “object” as malware. (*E.g.*, EX-1001 Claims 1, 15.) The ’389 patent describes a computer “object” as including a computer file, partial file, sub-program, macro, web page, any piece of code to be operated by or on the computer, or any other event. (EX-1001 8:1-4.)

35. Unlike the methods described in the prior art, the '389 Patent describes a rapid determination of whether an object (*e.g.*, a file) should be classified as safe or malware, without requiring a preliminary detailed analysis of the object. (EX-1001 3:14-24.) I explain this in further detail below.

36. The '389 patent teaches a method that may be implemented in a distributed network, such as a local area network, having a “central” or “base” computer connected to multiple “remote” computers. (EX-1001 7:59-64, Fig. 1.)

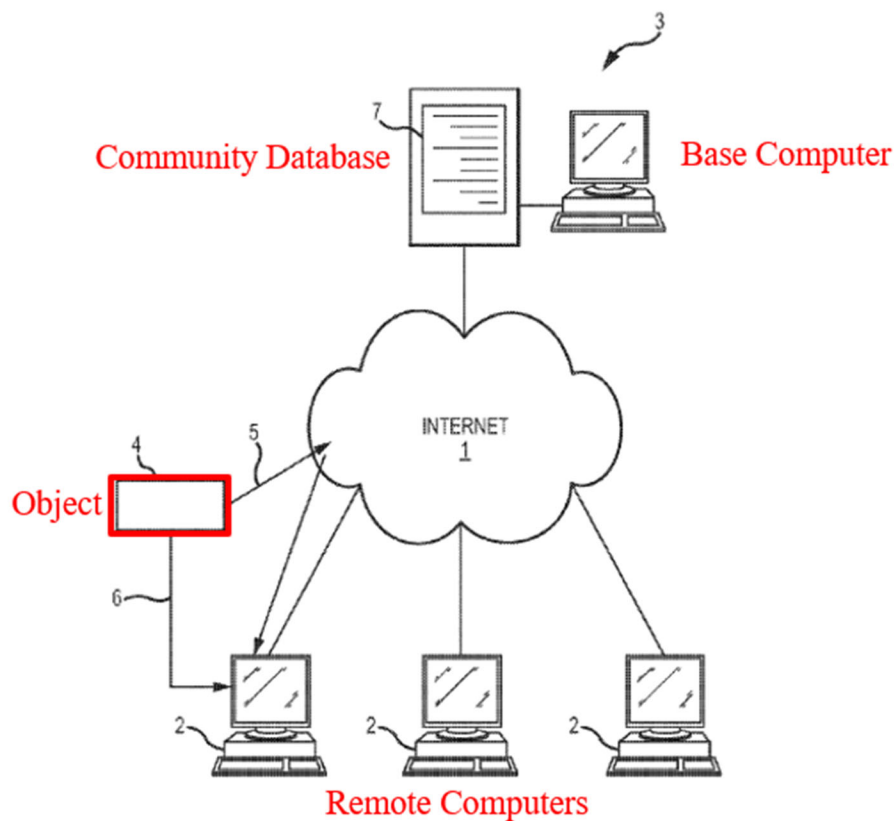


FIG.1

(EX-1001, Fig. 1, 7:59-8:20 (annotated).)

37. At a high level, the '389 patent claims a method (and system for implementing this method), in which the base computer receives information about a computer object from multiple remote computers and correlates the received data about the object. (EX-1001 3:4-13, cls. 1, 15.) After correlating the data, the base computer then compares the correlated data to other objects or entities and, on the basis of that comparison, classifies the computer object as malware. *Id.*

38. More specifically, the '389 patent discloses a “community database” that “is populated, over time, with information relating to each object run on all of the connected remote computers.” (EX-1001 8:12-14.) When an object is encountered at one of the remote computers, data related to the object is sent to the base computer, where it is stored in the community database for further analysis. (EX-1001 8:21-28, 8:47-50.) Such data may include executable instructions within the object; the size of the object; the current name of the object; the physical and folder location of the object; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and events initiated by or involving the object. (EX-1001 4:58-67.) The data stored in the community database provides “an extensive corollary of an object’s creation, configuration, execution, behaviour, identities and relationships to other objects[.]” (EX-1001 10:37-42.)

39. This object-specific data facilitates a rapid and more efficient determination of whether a particular object is malware. For example, one of the ways that this data may be provided to the base computer is in the form of a “key,” or “signature” which allows for reduced transmission and storage requirements and faster analysis at the base computer. (EX-1001 3:48-56.) A key containing data that represents executable instructions of an object “allows a comparison to be made at the base computer of only the executable instructions of the object.” (EX-1001 3:57-61.)

40. As another example, a new version of a program may be released which has minor changes compared to a previous version that is already known by the base computer. (EX-1001 3:65-4:4.) The collection of object-specific information—including executable instructions—can be useful in distinguishing between malware and mere revisions to previous software and allow for a more rapid and efficient determination of whether that object is malware. (EX-1001 4:4-12.)

41. When the base computer receives data about a particular computer object from two or more computers, the base computer can correlate this object-specific data. (EX-1001 cls. 1, 15, 10:43-46.) “Compared to the prior art that relies solely on signature matching, this aspect allows a comparison to be made between the objects and/or their effects on the different remote computers to determine

whether or not a particular object should be classed as good or as malware.” (EX-1001 3:14-18.)

42. The '389 patent teaches that this comparison preferably “identifies relationships between the object and other objects.” (EX-1001 3:37-38.) “[A]n object is considered to have a relationship based on the event performed by it upon another object, or upon it by another object.” (EX-1001 11:11-13.) In plain terms, a relationship may exist where (1) one object created or was created by the other object; (2) when one object has created a registry key to run or configure the other object; or (3) when one object was configured by or had registry keys created by the other object. (EX-1001 11:7-11.) Thus, the “other objects” may include the object-in-question, parent objects (*i.e.*, the object-in-question’s creator), child objects (*i.e.*, objects the object-in-question has created), or other process-related objects. (EX-1001 3:43-47.)

43. This relationship can then be used to make an efficient determination as to whether a particular object should be classed as malware. (*See* EX-1001 cls. 1, 15, 3:38-43.) The '389 patent explains:

In an example, this can be used to immediately and automatically to mark a child object as bad (or good) if the [] parent or other related object is bad (or good). Thus, if at least one other object to which said object is related is classed as malware, then the method may comprise classifying said object as malware.

(EX-1001 3:38-43; *see also id.* 11:5-60, 17:24-18:7.)

44. The '389 patent provides a real-world example of this method, involving the notepad.exe program, which is commonly installed on computers running the Windows operating system. (EX-1001 11:30-32.) In this case, the object is the notepad.exe program, which has a “limited range of functionality” (*e.g.*, writing and editing text). (EX-1001 11:32-35.) Thus, it would be unexpected to see an object like notepad.exe perform certain actions such as transmitting data to another computer or program. *Id.* Prior art systems might have simply flagged such behavior as suspicious. (*See* EX-1006 [0006] (describing the prior art’s propensity for false alarms due to the dynamic behavior of users and programs).)

45. The '389 patent, however, is able to run a comparison with other objects and identify relationships which may provide context for the expanded capability. (*See* EX-1001 11:35-42.) For example, the object notepad.exe may be modified by injecting new code into it, *e.g.*, via a dynamic link library injection, which would modify the capabilities of notepad.exe. (EX-1001 11:35-39.) The '389 patent discloses using the data that defines the relationships between objects to deduce whether or not the new behavior of a program can be attributed to its association with another object. (EX-1001 11:35-42.) The object can then be marked as malware or not, as appropriate. (EX-1001 11:42-44.)

46. The '389 patent's centralization of object-specific community data allows the base computer to correlate and compare the data and make a "rapid identification and determination" of "whether or not a particular object should be classed as good or as malware." (EX-1001 3:14-24, 11:55-59, cls. 1, 15.) This classification promotes future efficiency, because when the network encounters objects that have been classified in the past, the prior classification negates the need to perform time-consuming "deep analysis" of the object itself. (EX-1001 3:14-24, cl. 14.)

B. Claim Construction

47. Petitioner states that it relies on the plain and ordinary meaning of the claims. (Petition 7.) I understand that Petitioner and its co-defendants raised multiple terms for construction in a parallel proceeding in the Western District of Texas, and that the District Court issued a claim construction order on March 16, 2023, finding that no terms of the '389 patent were indefinite and that all terms should be construed to have their plain and ordinary meaning. The Court also provided several clarifying remarks, reproduced below. Consistent with the Court's order, I apply the plain and ordinary meaning of all claim terms.

Claims	Claim Term	Court's Construction
1, 15	"similar objects" / "similar computer objects"	Not indefinite. Plain-and-ordinary meaning. (EX-[Dkt. 236] 3.)

14	“a mask for said computer object that defines acceptable behaviour for the computer object”	Plain-and-ordinary meaning Note not for the jury: the plain-and-ordinary meaning of this term is not the inverse of a mask of unacceptable behavior (<i>Id.</i> 4.)
1-14, 26, 30	“events initiated [by] or involving the computer object”	Plain-and-ordinary meaning Note: the plain-and-ordinary meaning of “events” includes but is not limited to “actions or behaviours of an object acting upon another object or some other entity” (<i>Id.</i> 5.)
1	“first remote computer” / “second remote computer”	Plain-and-ordinary meaning (<i>Id.</i> 5.)
1	“correlating...at least a portion of the data about the computer object...comparing...the correlated data about the computer object”	Plain-and-ordinary meaning. The claim language recites that at least a portion of the data, but not all of the data, must be correlated before comparing. (<i>Id.</i> 6-7.)
1, 6, 8, 9, 12, 14, 30	“remote computer”	Plain-and-ordinary meaning (<i>Id.</i> 8.)

C. Level Of Ordinary Skill In The Art

48. I understand that certain issues relating to the validity of the '389 patent must be judged from the perspective of a person of ordinary skill in the relevant art,

as I discuss below. I have been asked to define the level of a “person having ordinary skill in the art” or “POSITA” that someone would have had at the time the alleged invention as claimed was made. In deciding the level of ordinary skill, I have considered the following factors, which I have been informed are relevant in determining the level of ordinary skill in the art:

- The levels of education and experience of persons working in the field;
- The types of problems encountered in the art;
- The prior art solutions to these problems;
- The rapidity with which innovations are made; and
- The sophistication of the technology.

49. A POSITA at the time of the invention of the '389 patent (which claims priority to an earlier application (GB 0513375.6) filed in Great Britain on June 30, 2005) would have had a Bachelor's degree in an accredited program of Electrical Engineering, Computer Engineering, or Computer Science or in a similar discipline, and have 2-3 years of practical work or research experience in the general fields of electrical engineering, computer science, networking, communications, and device and network security. More advanced degrees and/or training in a related discipline can compensate for shorter work experience.

50. I understand that Petitioner has defined the level of ordinary skill in the art as:

A person of ordinary skill in the art (“POSITA”) in June 2005 would have had at least a Bachelor’s degree in Computer Science, Computer Engineering, Electrical Engineering, or an equivalent discipline, and at least two years’ worth of experience developing dynamic malware detection and computer networking or network architecture, or a Master’s degree in a related field. EX-1004 ¶17. Alternatively, a POSITA could have four or more years of work experience in computer networking and in the development of client-server systems. *Id.* A POSITA would either have six months’ experience as a network or system administrator, or have taken two advanced computer-security courses. *Id.*

(Petition, 9.)

51. Despite minor variations in these two positions, these variations would not alter my opinions expressed in this Declaration.

52. Based on either standard, I was at least a person of ordinary skill in the art at the time of the invention of the ’389 patent.

V. OVERVIEW OF PRIOR ART

A. Overview Of Halperin

53. Halperin discloses a method for detecting where the collective behavior of computers on a network is suspicious, such that a virus may be present, and if so, taking containment actions to halt the further spread of the viral activity. (EX-1006

Abstract, [0036]-[0057].) Unlike the '389 patent, Halperin does not disclose a determination of whether any particular object is malware or whether an object should be classified as malware.

54. In discussing prior art approaches “to protecting against computer viruses,” Halperin explains that some antivirus software identified typical virus behavior by defining legitimate software behavior and then identifying any deviation from such behavior. (EX-1006 [0006].) But Halperin criticizes this type of software-specific (or object-specific) technique, because it does not account for the fact that computer user behavior is “quite dynamic and tends to vary over time and between different users.” (EX-1006 [0006].) As a result, application of such a technique often resulted in a false alarm whenever a user did anything unusual. (EX-1006 [0006].) Irritated by such false alarms, computers users would disable this type of antivirus software or set the sensitivity of the software so low that new viruses would not be detected. (EX-1006 [0006].)

55. Within the context of this prior art, Halperin recognized that viruses “are capable of spreading from computer to computer with little or no intervention on the part of the computer user.” (EX-1006 [0004].) “In most cases, viruses are spread when computer users send infected files to other computer users” via e-mail, data storage media, or by copying infected files from one computer to another via a network. (EX-1006 [0003].) Halperin provides the following example:

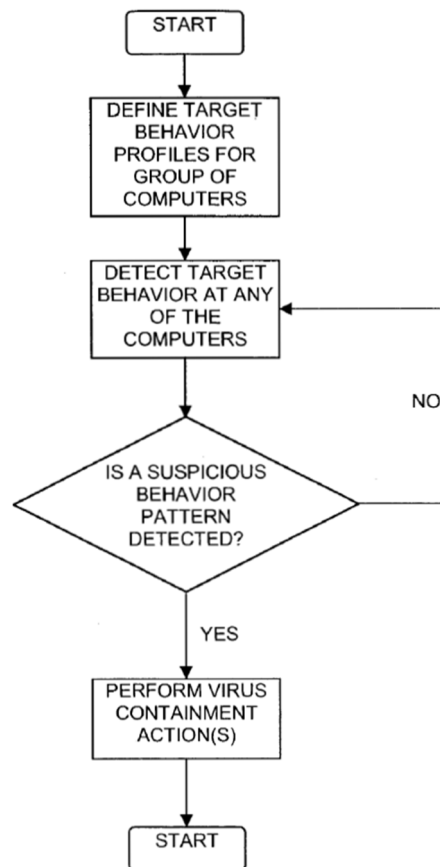
A virus that spreads via e-mail messages will typically access an e-mail program's address book or sent/received mail folders and automatically send itself to one or more of these addresses. Alternatively, the virus may attach itself to otherwise innocuous e-mail messages that are sent by a computer user to unsuspecting recipients. Other viruses appear on web pages and are spread by being downloaded into a user's computer automatically when the infected web page is viewed.

(EX-1006 [0004].)

56. Halperin proposes a solution that—rather than analyze each individual object—analyzes collective behavior of computers on the network. (EX-1006 Abstract, [0036]-[0047].) Halperin's focus on collective computer behavior is based on the idea that “computer viruses typically infect a computer system by moving from one computer to another within a computer network.” (EX-1006 [0037].) Thus, “computers that share the same network services are often infected within the same time period.” (*Id.*) Therefore, Halperin posited that a virus on the network could be detected (and false alarms reduced) by evaluating computer-level behavior across a network. (EX-1006 [0037], [0047].) Halperin explained that it would be easier to attribute abnormal activity to the presence of a virus if that activity was observed on several computers in a network, rather than just one. (EX-1006 [0037].) For example, detecting the deletion of 10+ files on one computer may not be suspicious—perhaps an employee was simply deleting an old folder. (*See, e.g.*, EX-

1006 [0042].) However, noting that 10 computers all deleted 10+ files within 1 minute of each other may appear suspicious. (*See, e.g.*, EX-1006 [0050].)

57. Accordingly, Halperin proposes a solution that analyzes collective behavior of computers on the network to determine whether their collective behavior indicates the presence of a virus. (EX-1006 Abstract, [0036]-[0047].) Figure 6 illustrates an overview of this process:



(EX-1006, Fig. 6.) I discuss the steps of this method in detail below.

58. In order to evaluate collective computer behavior, Halperin discloses defining one or more “target behavior profiles” for the networked computers. (EX-

1006 [0038], [0047].) “Target behavior may be any and all *computer*¹ activity”—*i.e.*, Halperin discloses defining profiles of *computer* behavior. (EX-1006 [0038].)

Halperin discloses exemplary “target behavior profiles” for the computers, such as:

Sending messages to more than a predefined number of users during a predefined period of time;

Sending messages not as a result of a direct user interaction with the Graphic User Interface (GUI) of the message software, but rather as the result of a directive from a software application;

Modifying operating system files such as the Microsoft Windows registry;

Deleting more than a predefined number of files on the computer’s hard disk during a predefined period of time....

(EX-1006 [0039]-[0046].) Halperin teaches that the computers may be configured with such profiles and the ability to detect associated behavior and notify the server, or, alternatively, the server may be configured with the profiles and detect associated behavior.

59. Halperin explains that the server may “correlate the presence of target behavior detected at two or more of [the] computers [] in order to determine whether the correlated target behavior corresponds to a predefined suspicious behavior pattern.” (EX-1006 [0047].) Exemplary patterns of computer behavior include:

¹ All emphasis added unless otherwise noted.

A certain percentage of the computers in the network sending more than 10 messages per minute in the last 5 minutes;

A certain percentage of the computers in the network sending messages not initiated via the message GUI in the last 1 minute;

A certain percentage of the computers in the network deleting more than 10 files in the last 1 minute....[etc.]

(EX-1006 [0048]-[0056].) That is, whereas the '389 patent relies on data that is specific to a particular computer object in order to determine whether that object is malware (Section IV(A), *supra*), Halperin relies on data that is not particular to any object, because it seeks to determine whether a certain percentage of computers on the network are behaving unexpectedly. (*See id.*)

60. Halperin explains that when the system detects a suspicious pattern of this behavior across a certain percentage of computers, this indicates “that a computer virus may have infected those computers.” (EX-1006 [0047], [0037].) The server may then initiate one or more virus containment actions. (EX-1006 [0057].)

Examples include:

Suspending any or all messages sent by computer 100, thereby preventing messages sent by computer 100 from being forwarded to recipients.

Forwarding messages that are addressed to a decoy address to a third party for analysis, such as a company or other body that produces anti-virus software.

Notifying a user at computer 100 of suspicious message activity....[etc.]

(EX-1006 [0024]-[0030].)

61. Because Halperin's method is not focused on any particular object, it does not proffer any particular means of identifying a particular object (*e.g.*, a piece of software) as the potential culprit. That is, Halperin does not disclose how one would determine whether any particular object is malware and does not disclose removing the offending software. Halperin's system only seeks to determine whether a certain percentage of computers on the network are behaving unexpectedly and reacts to contain the spread of a potential virus. (*See* EX-1006 [0057], Abstract.)

62. A POSITA would understand that where the '389 patent examines information about a particular *object* across multiple computers on a network in order to determine whether the *object* should be classed as malware, Halperin looks for the same *behavior or activity* across multiple computers on a network to determine if the *network* shows signs of malware. (EX-1006 [0037]-[0038].) Therefore, while Halperin and the '389 patent could theoretically complement each other, the two inventions are fundamentally different.

63. I understand that during prosecution of the '389 patent, another prior art reference that contains the same disclosure as Halperin was before the USPTO examiner. (EX-[CFH] 222-227, 259.) Specifically, the examiner initialed an information disclosure statement (EX-2018 259), indicating he had reviewed both

U.S. Patent Application Publication No. 2002/0194490 to Halperin (“the Halperin ’490”) and an August 1, 2013 European Search report which discusses Halperin ’490. (EX-2018 259; *see also id.* 222-227.) The report remarks on the high level of relevance of Halperin ’490 (which is referred to in the report as “document D1”, given its similarity with the claims:

[I]t is noted that document D1 appears to be very relevant for the claimed subject-matter, since it discloses collecting data related to software activity (events) from plurality of clients, and correlating this data on the server in order to detect malicious software (see D1, par.[86] to [106]).

(EX-2018 225.) I have reviewed Halperin ’490 and paragraphs 86 through paragraph 106, in particular, and find that these paragraphs are identical to paragraphs 36 through 56 of Halperin (on which the Petitioner relies). (*Compare* EX-2011 Figs. 5-6, [0086]-[0106] *with* EX-1006 Figs. 5-6, [0036]-[0056].) Even after considering this disclosure, the examiner allowed the ’389 patent.

VI. HALPERIN DOES NOT RENDER OBVIOUS THE CHALLENGED CLAIMS

64. Petitioner and its expert allege that Halperin alone renders obvious every limitation of independent claims 1, 15, and 29, and this allegation underlies their challenge of all remaining dependent claims 2-14, 16-28, and 30.

Ground 1	Section 103(a)	Obvious over Halperin	Claims 1-6, 8-13, 15-20, 22-27, 29, 30
Ground 2	Section 103(a)	Obvious over Halperin in view of Kirby	Claims 3, 5, 7, 9-13, 17, 19, 21, 23-27
Ground 3	Section 103(a)	Obvious over Halperin in view of Kirby and Mankins	Claims 14 and 28

65. As I discussed above, I apply the plain and ordinary meaning of all claim terms, consistent with the district court's claim construction order. (*See* Section IV(B), *supra*.)

66. I disagree with the arguments in the Petition and the opinions set forth by Dr. Newman for at least the reasons that I discuss in further detail below.

A. Independent Claims 1, 15, and 29

67. Halperin explains that the central tenet to its method for reducing false alarms is that:

[C]omputer viruses typically infect a computer system by moving from one computer to another within a computer network....One characteristics of such types of infection is that computers that share the same network services are often infected within the same time period. A computer virus can thus be detected by correlating behavior and/or data from different computers. Activity that cannot be confidently

attributed to a virus when observed on one computer can be clearly identified as such when observed on several computers in a network.

(EX-1006 [0037]; *see also id.* [0006].) Because Halperin is directed to detection of computer behavior profiles and the evaluation of collective computer behavior across a network, Halperin does not disclose or render obvious multiple limitations of the '389 patent, which I discuss below. (EX-1006 [0038].)

1. [1B]: “*information including at least an identity of an object initiating the event*”

68. Petitioner relies on the same analysis of Halperin to support limitations [1B], [15B], and [29], in all Grounds 1-3 of its petition. (Petition 19-21, 27, 41.)

Claim Limitation	Claim Language
[1B], [15B], [29]	<i>“...wherein said data includes information about events initiated or involving the computer object when the computer object is created, configured or runs on the first remote computer, said information including at least an identity of an object initiating the event, the event type, and an identity of an object or other entity on which the event is being performed”</i>

69. In analyzing whether Halperin discloses this limitation, it is critical to consider this language within the context of limitations [1A] and [15A], as well.

(EX-1001 cls. 1, 15.) The claims of the '389 patent recite “*at a base computer, receiving data about a computer object from a first remote computer...wherein said data includes information about events initiated or involving the computer object when the computer object is created, configured or runs on the first remote computer, said information including at least an identity of an object initiating the event.*” That is, the claims require receiving data about an object and that that data includes information about events initiated by or involving that object, as well as that that information include at least an identity of an object initiating the event.

70. Petitioner and Dr. Newman appear to point to two possible Halperin disclosures as corresponding to “*an identity of an object initiating the event*”: (1) a central database’s logging of events that includes “on which computer [the event] occurred (if relevant);” or (2) Halperin’s disclosure of generating a “checksum” for an “e-mail attachment.” Petitioner appears to further allege that the receipt of an object identity would have “necessarily” occurred or that inclusion of such an identity would have been obvious. I address each of these arguments in turn, below.

a. Halperin’s disclosure of logging a computer does not meet this limitation.

71. As I discussed in detail above, Halperin applies method, in which remote computers detect computer-level “target behavior profiles” and notify a server, accordingly. (EX-1006 [0047], [0038].) Target behavior profiles are defined for remote computers on a network. (EX-1006 [0038].) Halperin teaches that these

profiles “describe[] behavior that should be the subject of correlation analysis.” (EX-1006 [0038].) Halperin provides exemplary target behavior profiles, such as sending messages to too many users or deleting too many files during a predefined period of time. (EX-1006 [0039]-[0046].) The server then collects information regarding the target behavior profiles. (EX-1006 [0047].)

72. In a separate embodiment, Halperin discloses a distinct method in which suspicious behavior is collected at sensors and logged at a central database. (EX-1006 [0090].) Halperin teaches logging “the type of event, the time it occurred and on which computer it occurred (if relevant).” (*Id.*) “Either every predefined period of time or upon each event, the system queries the database for all events that took place during the last time slot.” (*Id.*) The system calculates the probability of receiving a certain number of events in a certain amount of time. (*Id.*) If that calculated value is lower than a predefined threshold, the system can activate a virus alarm and possibly initiate virus containment actions. (*Id.*)

73. Despite Petitioner relying on the embodiment in paragraphs [0038] – [0057] for the majority of its allegations, Petitioner gives no basis or explanation for adding the “logging” step in Halperin’s distinct paragraph [0090] embodiment. And it does not appear that the “logging” described in paragraph [0090] would be necessary within Halperin’s [0038]-[0057] embodiment, because the remote computers are already collecting the predefined “target behavior profiles.” (*See* EX-

1006 [0038]-[0046].) Accordingly, the server would not need to log every event detected by a sensor, because the remote computers are already looking for specific behavior profiles—e.g., sending messages to too many users during a predefined period of time—and then providing that behavior information to the server. (*See id.*; *see also* EX-1006 [0047].)

74. Petitioner has cherry-picked steps from distinct embodiments and provided no explanation as to why the logging step would be necessary or advantageous in an alternative embodiment. Dr. Newman’s declaration merely repeats the statements made in the petition. (EX-1004 ¶71.)

75. Even if one were to assume that Halperin’s [0038]-[0057] embodiment would be modified to log the information Halperin discloses, Halperin still does not disclose receiving data about a computer object that includes “*an identity of an object initiating the event.*” Halperin discloses logging “the type of event, the time it occurred and on *which computer it occurred (if relevant).*” (EX-1006 [0090].)

76. A POSITA would understand that a computer is not an “*object initiating the event.*” The ’389 patent explains:

“An object in this sense may be a computer file, part of a file or a sub-program, macro, web page or any other piece of code *to be operated by or on the computer*, or any other event whether executed, emulated, simulated or interpreted. An object 4 is shown schematically in the figure and may for example be downloaded to a remote computer 2 via

the Internet 1 as shown by lines 5 or applied directly as shown by line 6.”

(EX-1001 8:1-4.) As a computer cannot “be operated by or on [a] computer” and because a computer is also not an “event,” a POSITA would understand that the plain and ordinary meaning of “*object*” excludes the computer itself.

77. The claims’ use of the term “object” further support this understanding. For example, all claims recite a method or system for “classifying a computer *object* as malware.” (EX-1001, cls. 1-30.) It would be nonsensical to classify a “computer [computer]” as malware. As another example, claims 6 and 20 recite “*wherein said other objects include the object or similar objects stored on at least one of the first and second remote computers.*” (EX-1001 cls. 6, 20.) Again, it would be nonsensical to say that “*other [computers]*” include the computer or similar computers “*stored on at least one of the first and second remote computers.*” Claim 8 recites exemplary data about the computer object, including the size of the computer object, the current name of the computer object, the physical and folder location of the computer object on disk. (EX-1001, cl. 8.) A POSITA would not understand a computer to be an “*object initiating the event.*”

78. Halperin’s election to log an originating computer (and not an originating object) is unsurprising, given that Halperin’s goal is evaluating collective *computer* behavior across a network. (Section V(A), *supra*; EX-1006 [0037]-

[0038].) Accordingly, Halperin’s disclosure of logging a computer (if relevant) on which an event occurred does not disclose receiving information that includes “*an identity of an object initiating the event.*”

b. Halperin’s disclosure of a checksum does not meet this limitation.

79. Petitioner and Dr. Newman alternatively suggest that Halperin’s disclosure of generating a “checksum” for an e-mail attachment corresponds to this limitation. (Petition 21.)

80. Before explaining why Halperin’s disclosure of a checksum does not disclose this limitation, it is important to understand the full context of this limitation. Across the limitations that Petitioner has broken into [1A]/[15A] and [1B]/[15B], the ’389 patent requires:

[A]t a base computer, receiving data about a computer object from a first remote computer on which the computer object or similar computer objects are stored, **wherein said data includes information about events** initiated or involving the computer object when the computer object is created, configured, or runs on the first remote computer, **said information including at least an identity of an object initiating the event**, the event type, and an identify of an object or other entity on which the event is being performed....

(EX-1001, cl. 1; *see also id.*, cl. 15.) That is, the ’389 patent requires that the identify of an object initiating the event be a part of the information (and object data) received at the base computer. (*See id.*)

81. Halperin teaches that “when malicious software has been identified in several computing devices within a group, it is possible to isolate the mechanism that has been spreading the malicious software.” (EX-1006 [0073].) Halperin does not disclose any determination of whether a particular object is malware or how such a determination would be made, but merely notes that “when malicious software has been identified,” “a characteristic code” (also known as a “checksum”) may be generated for the “e-mail attachment that when activated causes the malicious software to spread.” (EX-1006 [0073].)

82. *First*, the petition does not provide any evidence that the generation of a checksum is tied in any way to the computers’ detection of target behavior profiles (i.e., the alleged “*data*”); indeed, there is none. (*See* EX-1006 [0073].) Petitioner has alleged that the computer behavior profiles (that define the type of computer behavior to be detected) correspond to the claimed “*data...wherein said data includes information about events initiated or involving the computer object.*” (Petition 19-20.) But there is no evidence in Halperin that suggests generating a checksum for an object that initiated the computer’s behavior. (*See* EX-1006 [0038], [0073].) Indeed, because these target behavior profiles define *computer*-level behavior, these profiles encompass many events (whether virus-related or not) and there is no evidence that Halperin generates a checksum or is even aware of objects involved in these events. (*See* EX-1006 [0038]-[0046].) Accordingly, Halperin does

not disclose collecting information about an event that includes “*an identity of an object initiating the event.*”

83. *Second*, Halperin teaches this optional generation of a “checksum” long *after* Halperin’s system has already detected target behavior profiles—*i.e.*, after it has gathered the information which Petitioner points to as “*data about a computer object.*” (See Petition 19-20; EX-1006 [0072]-[0073].) As I have discussed above, Halperin discloses a method which involves detecting computer-level behavior profiles (EX-1006 [0038]-[0046]), evaluating whether a certain percentage of computers are behaving in a particular way—*i.e.*, whether there is a suspicious pattern of behavior (EX-1006 [0047]-[0056]), and taking virus containment actions. (EX-1006 [0057].) Petitioner (incorrectly) alleges that Halperin’s server receives object-specific data somewhere in between the detecting and evaluating steps. (Petition 18-19.) But Halperin does not disclose optionally generating a checksum until *after* the server detects and evaluates the behavior profiles and takes virus containment actions. (See EX-1006 [0047], [0057], [0072]-[0073].) It would, therefore, be impossible for the behavior information to contain a checksum. That is, even if Petitioner’s allegations were correct (they are not), the server in Halperin’s system does not **receive** information (and object data) that includes “*an identity of an object initiating the event.*” Accordingly, Halperin’s disclosure of generating a checksum does not disclose this limitation.

c. Petitioner and Dr. Newman fail to explain why receipt of an object identity would be inherent.

84. As a preliminary matter, I understand that a prior art reference inherently discloses matter only where such matter would necessarily have occurred or be present. I further understand that the mere fact that a certain thing may result from a given set of circumstances is not sufficient.

85. The petition includes a one-sentence argument that “a POSITA would understand the identity of the object initiating the event necessarily must be collected to perform the correlation analysis Halperin discloses and ultimately to determine whether the object is malware.” (Petition 21.) Dr. Newman’s declaration—while varying the language of the sentence slightly—merely repeats this one-sentence argument. (EX-1004 ¶74.)

86. I disagree with Petitioner and Dr. Newman’s conclusory statement. As I discussed above, Halperin criticizes an object-specific approach to detecting virus behavior. (Section V(A), *supra*.) Specifically, Halperin describes a prior art approach that relied, for example, on “characterizing legitimate software behavior,” which will differ for each software, and “then identifying any deviation from such behavior.” (EX-1006 [0006].) Halperin opines that, because “computer user behavior is quite dynamic and tends to vary over time and between different users,” such a technique can often result in a false alarm. (*Id.*)

87. Halperin sought to overcome such a disadvantage and discloses the collection and evaluation of “target behavior,” which the server “may correlate.” (EX-1006 [0047].) While Dr. Newman opines that “the system would by necessity obtain” the identity of the object initiating an event “so that it could perform Halperin’s correlation analysis” (EX-1004 ¶74), Halperin makes clear that no object identity is necessary. (EX-1006 [0047].)

88. Halperin discloses the use of “*behavior* correlation techniques”—*i.e.*, “identifying the same *activity* in different computers at about the same time, or by identifying *repeating patterns of data* within memories of two or more computers.” (EX-1006 [0047].) The identity of an object initiating an event would not be necessary to these behavior correlation techniques, particularly since the activity or data patterns would cover many events, initiated by many objects. (*See* EX-1006 [0039]-[0046].)

89. For example, Halperin discloses a target behavior profile that corresponds to a computer “[s]ending messages to more than a predefined number of users during a predefined period of time.” (EX-1006 [0039].) In order for the server to determine whether a suspicious behavior pattern is present—*e.g.*, whether “[a] certain percentage of the computers in the network” have sent more than 10 messages per minute in the last 5 minutes—the server will need to correlate the behavior information across computers in the network to determine how many

computers detected that target behavior profile. (*See* EX-1006 [0039], [0048].) Such a determination is not object-specific and does not require the identity of the various objects on each of the computers that may have initiated an event. That is because, regardless of whether each individual instance of behavior was legitimate or not, Halperin's method will implement virus containment actions because a suspicious pattern of behavior has been detected. (*See* EX-1006 [0048], [0057].)

90. Accordingly, a POSITA would not understand an identity of an object initiating the event to be necessary to Halperin's method.

91. Petitioner's and Dr. Newman's one-sentence argument also suggests that such information would have been necessary to "decide whether or not to classify the object as malware." (Petition 21; EX-1004 ¶74.) However, Halperin does not disclose a determination of whether an object is malware.

92. As I have discussed previously, Halperin discloses the detection and containment of viral activity. (Section V(A), *supra*.) None of these steps—*i.e.*, the detection of target behavior profiles, the evaluation of collective behavior for a suspicious behavior pattern, or the containment actions—corresponds to a determination of whether a specific object is malware. (*See* EX-1006 [0038]-[0057], [0023]-[0030].)

93. The fact that Halperin discloses generating a checksum for an e-mail attachment *after* malicious software has been identified does not explain why

Halperin's server would have received the not-yet-generated checksum in its initial gathering of behavior information from the remote computers. (*See* EX-1006 [0073], [0048].) A checksum algorithm—at a high-level—merely generates a value (i.e., SUMmation CHECK) that represents the contents of an electronic resource, such as an e-mail attachment, in a highly summarized manner. And Halperin expressly teaches waiting to generate such a checksum until after the mechanism used to spread a virus has been identified. (EX-1006 [0073].) Thus, a POSITA would not understand a need for the server to receive the identity of object(s) initiating the event(s) covered by a target behavior profile.

d. Petitioner and Dr. Newman fail to explain why receipt of an object identity would be obvious.

94. Dr. Newman opines in a final one-sentence argument that “it would have been obvious to a POSITA to include an identity of an object initiating the event (*e.g.*, the email-attachment checksum in Halperin), in the suspicious-event information logged in the central database.” (EX-1004 ¶74; Petition 21.) Neither Petitioner nor Dr. Newman provide a basis for this conclusion. And once again, neither provides a basis for blending two distinct embodiments in Halperin. (Section VI(A)(1)(b), *supra*.) For many of the same reasons I have discussed above, I disagree.

95. As I explained immediately above, Halperin's detection and containment method does not need an identity of an object(s) initiating the event(s),

because regardless of which object(s) initiated events that contributed to “[a] certain percentage of the computers in the network” sending more than 10 messages and regardless of whether each individual message was legitimate or not, Halperin’s method will implement virus containment actions because a suspicious behavior pattern has been detected. (*See* EX-1006 [0048], [0057].). (Section VI(A)(1)(c), *supra*.) As there is no purpose for this information, it would not have been obvious to a POSITA to dedicate the time and resources necessary to including such information.

96. Moreover, there are good reasons *not* to generate and log a checksum for every object that initiates an event covered by a target behavior profile. Within the context of Halperin’s paragraph [0090] embodiment, Halperin discloses logging every event detected by a sensor—regardless of whether that event is legitimate or not. (EX-1006 [0090].) Even within the context of Halperin’s [0038]-[0057] embodiment, Halperin discloses a large variety of target behavior profiles that will be detected, including for example: “[s]ending messages,” “[d]eleting...files,” or “[l]oading a new software.” (EX-1006 [0038]-[0046].) To generate and collect a checksum—needlessly—for every object that initiates or is involved in one of these events would consume an extraordinary amount of both computational and transmission resources. Neither Petitioner nor Dr. Newman proffer a benefit of this inefficiency. Nor would there be a benefit from this costly expenditure of resources

because—as I explained above—there is no need for an identity of an object initiating an event within Halperin’s method.

97. Accordingly, it would not have been obvious to include an identity of an object initiating the event in the behavior information and Halperin does not render obvious receiving information about an event that includes “*an identity of an object initiating the event.*”

2. [1C]: “*receiving data about the computer object from a second remote computer...including at least an identity of an object initiating the event*”

98. Petitioner relies on the same analysis of Halperin to support limitations [1C], [15C], and [29] in all grounds of its petition. (Petition 22-23, 27, 41.)

Claim Limitation	Claim Language
[1C], [29]	“ <i>at the base computer, receiving data about the computer object from a second remote computer on which the computer object or similar computer objects are stored...wherein said data includes...information including at least an identity of an object initiating the event</i> ”
[15C]	“ <i>the base computer being constructed and arranged to receive data about the computer object from a second remote computer on which the computer object or similar computer objects are stored, wherein said data includes...information including at least an identity of an object initiating the event</i> ”

99. Petitioner asserts that Halperin’s disclosure of collecting computer behavior information corresponds to receiving data about the computer object. (Petition 22.) Petitioner—without any basis—injects its own language into Halperin’s disclosure, specifically alleging that Halperin discloses “collecting information regarding target behavior [of a potential virus].” (*Id.*) Dr. Newman does the same. (EX-1004 ¶77.) However, such a modification of Halperin’s disclosure is unsupported. I further disagree with Petitioner’s conclusions.

100. First, Halperin does not disclose “*receiving data about **the computer object** from a second remote computer.*” Second, for the same reasons previously discussed, the data Halperin discloses gathering from the remote computers does not include “*at least an identity of an object initiating the event.*”

a. Halperin does not disclose “*receiving data about the computer object from a second remote computer*”

101. Halperin does not disclose receiving behavior information about the same computer object from at least two computers.

102. The ’389 patent claims receiving data about a computer object from a first computer and then receiving data about “***the computer object***” from a second computer. (EX-1001, cls. 1, 15.) That is, the claims require receipt of data about the same computer object from at least two computers. (*See id.*)

103. This understanding of the claim’s requirements is further supported by the prosecution history, in which the applicant distinguished prior art “Omote” by asserting that:

As Omote is generally directed to a system implemented on a PC, Omote makes no mention of at least receiving data about a computer object from a first remote computer and receiving data about the computer object from a second remote computer.

(EX-[CFH] 207 (emphasis in original); *see also id.* at 208 (making the same distinction over Dozortsev).)

104. Unlike the ’389 patent, Halperin is only concerned with receiving information about computer-level *activity* across multiple computers in a network, which is not specific to any one object, let alone the same object.² Halperin discloses detecting one or more target behavior profiles at multiple computers. (EX-1006 [0038].) Examples of this include sending too many messages or deleting too many

² I note that the ’389 patent contemplates embodiments in which, for example, the object might have different creation dates across computers. (EX-1001 3:25-26, 3:57-4:4.) Regardless, Halperin is concerned only with computer-level *activity* across multiple computers and does not disclose receiving data about a particular object from a first computer and receiving data about the object from a second computer.

files in a predefined period of time. (EX-1006 [0039], [0042].) Halperin teaches that either the computers may detect such information and merely notify the server, or the server may detect the target behavior profiles “using conventional techniques.” (EX-1006 [0047].) In a separate embodiment, Halperin also teaches logging events detected by suspicious behavior sensors. (EX-1006 [0086], [0090].) The purpose of collecting such information is to identify the same *activity* in different computers or repeating patterns of data within memory. (EX-1006 [0047].)

105. However, as I have discussed previously, the computer behaviors or data patterns that the server is seeking to detect would cover many events, initiated by many objects. (See EX-1006 [0039]-[0046].) For example, Halperin discloses a target behavior profile that corresponds to a computer “[s]ending messages to more than a predefined number of users during a predefined period of time.” (EX-1006 [0039].) In order for the server to determine whether a suspicious behavior pattern is present—*e.g.*, whether “[a] certain percentage of the computers in the network” have sent more than 10 messages per minute in the last 5 minutes—the server will need to determine how many computers detected that target behavior profile. (See EX-1006 [0039], [0048].) Such a determination is not object-specific and does not require the identity of the various objects on each of the computers that initiated an event. Regardless of which object(s) initiated events that contributed to “[a] certain percentage of the computers in the network” sending more than 10 messages and

regardless of whether each individual message was initiated by a legitimate object or not, Halperin's method will implement virus containment actions because a suspicious behavior pattern has been detected. (See EX-1006 [0048], [0057].)

106. Accordingly, Halperin's disclosure of detecting target behavior profiles or logging events on multiple computers does not disclose receiving data about the same computer object from multiple computers.

107. The petition asserts that a POSITA would have known that Halperin's correlation "would require the information describe the 'computer object or similar computer objects.'" (Petition 23.) Dr. Newman makes the same statement and the only basis or explanation he provides is that a POSITA would have known this "since the system would only correlate data that described the same or similar objects." (EX-1004 ¶77.) I disagree.

108. Before I explain why this assertion is wrong, I note that both Petitioner and Dr. Newman appear to have misread the claim language. The independent claims require:

[A]t a base computer, receiving data about the computer object from a second remote computer on which the computer object or similar computer objects are stored, wherein said data includes information about events initiated or involving the computer object when the computer object is created, configured, or runs on the second remote computer, said information including at least an identity of an object initiating the event....

(EX-1001, cl. 1; *see also id.*, cl. 15.) That is, the claims require receiving data about ***the*** computer object—*i.e.*, the same computer object about which the base computer received data in the first limitation. (*Id.*) The clause “computer object or similar objects,” is recited only in reference to the computer from which the data is received. (*Id.*) That is, the second remote computer that provides data about the computer object, must also be storing “*the computer object or similar computer objects.*” (*Id.*) But the received object data must be “*about the computer object.*”

109. Dr. Newman’s assertion that a POSITA would have understood Halperin to teach collecting information about the same object because of its correlation—is incorrect. As I discussed above, Halperin discloses the use of “***behavior*** correlation techniques”—*i.e.*, “identifying the same ***activity*** in different computers at about the same time, or by identifying ***repeating patterns of data*** within memories of two or more computers.” (EX-1006 [0047].) Halperin’s first exemplary correlation technique--“identifying the same ***activity*** in different computers at about the same time”—merely requires the type of activity, the computer on which the activity took place, and the time the activity took place. (EX-1006 [0047]; *see also id.* [0090].)

110. Halperin’s only other example—“identifying repeating patterns of data within the memories of two or more computers”—is very non-specific but also does not disclose an object-specific correlation. (EX-1006 [0047].) “Data” is an

extraordinarily broad concept, which Halperin does not particularly define, that is merely a representation of something else—*e.g.*, facts, concepts, instructions, values, symbols, etc. Halperin’s disclosure of a technique which looks for repeating patterns of data could—once again—refer to repeating patterns of data that represent particular activity or behavior. (*See* EX-1006 [0047].) This technique does not disclose and certainly does not require that all of the data to be correlated be about the same object. (*See* EX-1006 [0047]-[0057].) Indeed, such a requirement could frustrate the purpose of Halperin which is detecting target behavior profiles at the computer level, across multiple computers—*i.e.*, where a certain percentage of computers on the network are behaving abnormally without regard to particular objects. (EX-1006 [0037], [0047].)

111. That is, the server does not care whether the behavior or patterns of data were about the same object—to the contrary, such broad behaviors (sending messages or deleting files) at the computer level need not be specific to any one object (or user). (*See* EX-1006 [0038]-[0046].) Because Halperin only seeks to detect collective computer behavior, these behavior correlation techniques do not require receiving information about the same object. (*See id.*; *see also* EX-1006 [0037].)

112. Accordingly, Halperin does not disclose “*receiving data about the computer object from a second remote computer.*”

- b. Halperin does not disclose “receiving data about the computer object from a second remote computer...including at least an identity of an object initiating the event”**

113. Like the data received from the first remote computer, the data received from the second remote computer must include data about the computer object that includes “*an identity of an object initiating the event.*”

114. Petitioner relies on its analysis of limitation [1B] and alleges that the second remote computer will have provided “the information described in step [1B]. (Petition 22.) And for the same reasons I discussed above in connection with limitation [1B], so does Halperin fail to disclose receiving data from a second remote computer that includes “*an identity of an object initiating the event.*” (Section VI(A)(1), *supra.*)

- 3. [1E]: “correlating...data about the computer object received from the first remote computer to...data about the computer object received from the second remote computer”**

115. Halperin does not disclose correlating data about a computer object that it received from a first computer to data about the same computer object received from a second computer.

116. Petitioner relies on the same analysis of Halperin to support limitations [1E], [15E], and [29] in all grounds of its petition. (Petition 22-23, 27, 41.)

Claim Limitation	Claim Language
------------------	----------------

[1E], [29]	<i>“correlating, by the base computer, at least a portion of the data about the computer object received from the first remote computer to at least a portion of the data about the computer object received from the second remote computer”</i>
[15E]	<i>“the base computer being constructed and arranged to correlate at least a portion of the data about the computer object received from the first remote computer to at least a portion of the data about the computer object received from the second remote computer”</i>

117. Petitioner presumes that any evidence of correlating data would meet this limitation (Petition 24), while failing to acknowledge that the ’389 patent requires correlating—specifically—data about the computer object received from the first computer to data about the computer object received from the second computer. (EX-1001, cls. 1, 15.) Petitioner’s statement that Halperin correlates “at least some of the data about the computer object (potential virus) received from each” computer cites no support.

118. Indeed, as I discussed above, Halperin discloses detection of target *behavior* profiles—i.e., computer behavior—and does not disclose collecting object-specific data or information. (Section VI(A)(2)(a), *supra*.) Thus, after detecting computer behavior that meets these target behavior profiles at two or more computers, Halperin discloses “***behavior*** correlation techniques,” which “correlate

the presence of *target behavior* detected at two or more [] computers.” (EX-1006 [0047].)

119. Halperin’s first exemplary correlation technique--“identifying the same **activity** in different computers at about the same time”—merely requires the type of activity, the computer on which the activity took place, and the time the activity took place. (EX-1006 [0047]; *see also id.* [0090].) Halperin’s only other example—“identifying repeating patterns of data within the memories of two or more computers”—is very non-specific but also does not disclose an object-specific correlation. (EX-1006 [0047].)

120. “Data” is an extraordinarily broad concept, which Halperin does not particularly define, that is merely a representation of something else—*e.g.*, facts, concepts, instructions, values, symbols, etc. Halperin’s disclosure of a technique which looks for repeating patterns of data could—once again—refer to repeating patterns of data that represent particular activity or behavior. (*See* EX-1006 [0047].) This technique does not disclose and certainly does not require that all of the data to be correlated be about the same object. (*See* EX-1006 [0047]-[0057].) Indeed, such a requirement could frustrate the purpose of Halperin which is detecting target behavior profiles at the computer level, across multiple computers. (EX-1006 [0047].)

121. The '389 patent does not merely require correlating “comparable” data, as Dr. Newman appears to assert (EX-1004 ¶83); rather, the '389 patent requires correlating data about the computer object received from a first computer with data about the computer object received from a second computer. (EX-1001, cls. 1, 15.) This understanding is further supported by the prosecution history, in which the applicant distinguished prior art “Omote” by asserting that:

As Omote is generally directed to a system implemented on a PC, Omote makes no mention of at least receiving data about a computer object from a first remote computer and receiving data about the computer object from a second remote computer. Additionally, there is no suggestion in Omote where at least a portion of the data about the computer object from the first and second remote computers is correlated....

(EX-[CFH] 207 (emphasis in original); *see also id.* at 208 (making the same distinction over Dozortsev).)

122. Accordingly, Halperin’s approach, which focuses on collective computer-level behavior, does not disclose this limitation.

4. [1F]: “*comparing...the correlated data...to other objects or entities to identify relationships between the correlated data and the other objects or entities*”

123. The petition does not address the latter half of this limitation.

124. Petitioner relies on the same analysis of Halperin to support limitations [1F], [15F], and [29] in all grounds of its petition. (Petition 24-26, 41.)

Claim Limitation	Claim Language
[1F], [29]	<i>“comparing, by the base computer, the correlated data about the computer object received from the first and second remote computers to other objects or entities to identify relationships between the correlated data and the other objects or entities”</i>
[15F]	<i>“the base computer being constructed and arranged to compare the data about the computer object received from the first and second remote computers to other objects or entities to identify relationships between the correlated data and the other objects or entities”</i>

125. The '389 patent claims comparing the correlated data about the computer object *“to other objects or entities to identify relationships between the correlated data and the other objects or entities.”* (E.g., EX-1001, cl. 1.)

126. As I explained above, the '389 patent provides an example of this step. (Section IV(A), *supra*.) For example, in a preferred embodiment, the '389 patent uses an object's relationship to other objects to assess whether an object is malware. (EX-1001 11:5-60, 17:24-18:7.) “[A]n object is considered to have a relationship based on the event performed by it upon another object, or upon it by another object.” (EX-1001 11:11-13.) That is, two objects may be said to have a relationship if, for example: (1) one object created or was created by the other object; or (2) one object created a registry key to run or configure the other object; or (3) one object was configured by or had registry keys created by the other object. (EX-1001 11:7-11.)

127. The '389 patent describes how this process might be used—for example, with the notepad.exe program (*i.e.*, object), which is commonly installed on computers running the Windows operating system. (EX-1001 11:30-32.) Because the notepad.exe program has a “limited range of functionality,” certain actions like transmitting data to another computer or program may be flagged as unexpected. (*Id.* at 11:32-35.) However, it is possible for notepad.exe’s capabilities to be modified by injecting new code into it, *e.g.*, via a dynamic link library injection. (EX-1001 11:35-38.) By identifying relationships between notepad.exe and other objects, “it is possible to deduce that the new behaviours ... can be attributed to the association with another object,” *e.g.*, that notepad.exe’s new behavior was a direct result of its relationship with another object. (*Id.* at 11:40-42.) If the new behavior is malicious, or if the program that modified notepad.exe is known to be malicious, then it is possible to classify one or both processes as unsafe based on the comparison of the two. (*Id.* at 11:42-44.)

128. This analysis is important for the '389 patent’s object-centric system because it allows the system to determine, for example, whether the object created, was created by, or is associated with another object known to be malware. (EX-1001 11:18-21.)

129. Neither Petitioner nor Dr. Newman provide any explanation as to what relationship Halperin allegedly discloses or how such relationship is identified;

therefore, the petition contains no assertion as to this portion of the limitation for me to evaluate. (Petition 24-26; EX-1004 ¶87.) Rather, Dr. Newman merely makes the conclusion that the alleged comparison must be made “in order to identify relationships between the correlated data and the other objects,” without explaining the basis for such conclusion. (EX-1004 ¶87.)

130. Because Petitioner and Dr. Newman do not explain what relationship is purportedly identified in Halperin, it is also impossible to determine what the benefit of such an alleged identification would be within Halperin’s method. As I have discussed at length, Halperin’s system is concerned with collective computer behavior (and not an object’s). (Section V(A), *supra*.) That is, Halperin detects behavior profiles defined *for a computer*, not an object. (EX-1006 [0038].) Halperin’s method does not disclose or contemplate an object-specific determination; rather, Halperin detects suspicious patterns of computer behavior across a network. (EX-1006 [0047]-[0057].) Halperin explicitly explains the advantages of this network-level focus, stating that “[a]ctivity that cannot be confidently attributed to a virus when observed on one computer can be clearly identified as such when observed on several computers in a network.” (EX-1006 [0037].) Thus, a POSITA would understand that defining object-to-object relationships, as Petitioner and Dr. Newman appear to suggest, would not only

require extensive modification of Halperin’s system and method but burn resources to no effect.

5. [1G]: “*classifying...the computer object as malware on the basis of said comparison*”

131. Petitioner’s assertion that Halperin teaches classifying anything as malware as a result of the base computer detecting a suspicious pattern of behavior is incorrect. (Petition 26.)

132. Petitioner relies on the same analysis of Halperin to support limitations [1G], [15G], and [29], in all grounds of its petition. (Petition 26-27, 41.)

Claim Limitation	Claim Language
[1G], [29]	<i>“classifying, by the base computer, the computer object as malware on the basis of said comparison.”</i>
[15G]	<i>“the base computer being constructed and arranged to classify the computer object as malware on the basis of said comparison”</i>

133. Both Petitioner’s and Dr. Newman’s analysis of this limitation is very limited—making two arguments in three sentences.

a. Halperin’s detection of a suspicious behavior pattern does not disclose classification of an object as malware.

134. **Petitioner’s first argument**, in total, not including citations is:

Halperin explains if the base computer determines “the correlated target behavior corresponds to a predefined suspicious behavior pattern of

target behavior,” it classifies the computer object as malware (e.g., a “computer virus”). Thus, by comparing the correlated data about a computer object to another object or entity, Halperin determines whether that computer object is malware and classifies it as such.

(Petition 26.) But the petition contains no assertion as to what in Halperin corresponds to a classification, making it impossible for me to evaluate any such claim.

135. The petition includes two quotes from Halperin in parenthetical citations, which do not elucidate. (Petition 26.) The first quote states, “The proposed system works in two phases to identify malicious code (from here on referred to as ‘viruses’).” (*Id.* (citing EX-1006 [0087]).) This is merely an overview of one of Halperin’s embodiments (notably, not the embodiment on which Petitioner primarily focuses most of its analysis). (EX-1006 [0087].) Halperin explains that one of its embodiments uses a learning phase and detecting phase—each phase preferably implemented on different computer networks. (*Id.*)

136. And while this quote speaks of “identify[ing] malicious code,” a review of Halperin’s method—including this embodiment—shows that this merely refers to Halperin’s goal of detecting the presence of possible viral activity in order to initiate either a virus alarm or virus containment actions. (*See* EX-1006 [0087]-[0090].) For example, at the end of the “detecting phase” referenced in in Petitioner’s quoted language, Halperin discloses calculating a probability of having measured

certain behavior information in a defined time period “assuming that there is no virus in the system.” (EX-1006 [0090].) If that result “is lower than [a] predefined threshold, the system can activate a virus alarm and possibly initiate virus containment actions.” (*Id.*) That is, Halperin is still merely detecting the presence of virus-like activity by computers on a network and initiating either an alarm or containment actions. (*See id.*)

137. Further disclosure of Halperin confirms this. Halperin teaches that the system may optionally enable an operator to enter information on whether—at a particular time slot—there is “a centralized operation that may ***look like a virus operation***, e.g. a mass software installation in the network.” (EX-1006 [0090].) This allows certain sensors that may have been activated by such an operation (e.g., Windows Registry sensors) to be disregarded at that time. (*Id.*)

138. The second quote Petitioner relies on states, “[T]he identification of virus activity may include automatic identification of suspicious activity by a server or a combination of automatic identification and a notification of a system operator and approval by that operator that the suspicious activity is truly a virus, before notifying other servers.” (Petition 26 (quoting EX-1006 [0061]).) Once again, this teaching of Halperin merely confirms that Halperin is directed to detecting potential viral ***activity*** and not classifying any particular object as malware. (EX-1006 [0061]; Section V(A), *supra.*)

139. In this paragraph [0061], Halperin explains that its method could be implemented in one of two ways. (EX-1006 [0061].) First, one might use Halperin’s method to automatically identify suspicious *activity* by a server, which can then notify other servers. (EX-1006 [0061].) Second, one might use a combination of Halperin’s method to automatically identify suspicious *activity* and a subsequent approval by that operator that the suspicious *activity* is truly the result of a virus, before notifying other servers. (EX-1006 [0061].)

140. To borrow an example from elsewhere in Halperin’s specification, the addition of a subsequent check by an operator could, for example, catch false positives, such as where there was a mass software installation on the network that was legitimate but which might have *looked like* suspicious activity. (EX-1006 [0061]; *see also id.* [0090].) However, where there was no legitimate action, such as a mass installation, to explain abnormal network-wide behavior, the operator could allow the method to continue to notifying other servers. (EX-1006 [0061].)

141. Halperin explains that once the server has notified other servers that the presence of a potential virus has been detected, other systems “may then increase or otherwise adjust its sensitivity level. (EX-1006 [0061].) For example, those systems may heighten their sensitivity level related to the decoy address or correlation parameter that first detected the presence of viral activity to increase the likelihood of detecting any similar viral activity on their own network. (EX-1006 [0061].)

142. Again, these quotes merely confirm that Halperin teaches detection of *collective computer-level behavior* on the network to detect the possibility of a viral infection and subsequent containment. (EX-1006 [0047], [0057].) Dr. Newman merely repeats the same quotes without further explanation. Accordingly, the language Petitioner quotes does not show that Halperin discloses classifying a particular object as malware.

143. Moreover, contrary to Petitioner’s conclusion, even where Halperin’s system detects a suspicious behavior pattern, Halperin discloses initiating virus containment actions—not a determination of whether *a particular object* is malware. (EX-1006 [0047], [0057].) Halperin discloses a determination of whether there is evidence of a suspicious behavior pattern—*e.g.*, whether a certain percentage of computers on the network are behaving unexpectedly. (EX-1006 [0047].) Halperin explains that detection of a suspicious behavior pattern “indicat[es] that a computer virus *may have* infected those computers.” (*Id.*) “Upon detecting a suspicious behavior pattern server [] may initiate one or more virus containment actions.” (EX-1006 [0057].) That is, Halperin implements a method of detecting where there is potential viral activity on a network and actions to contain the spread of a possible virus. (*See id.*) But these steps do not disclose or necessarily require an object-specific determination of whether a particular object is malware or classification of

such object. (*See id.*) Accordingly, Halperin's disclosure of detecting a suspicious behavior pattern does not disclose classification of a particular object as malware.

144. A POSITA would understand that Halperin's front-end omission of object classification is by design because nothing about Halperin's back-end containment teachings contemplates—much less requires—classifying any particular object as malware. (*See* EX-1006 [0047]-[0057].) For example, as I discuss in further detail in the next few paragraphs, none of Halperin's exemplary containment actions require a prior classification of an object as malware.

b. Classification of an object as malware is not inherent to Halperin's disclosure of virus containment actions.

145. **Petitioner's second argument** is:

A POSITA would further understand from Halperin that when the disclosed computer system responds to the detection of a virus (e.g., by “activat[ing] a virus alarm” or “initiat[ing] virus containment actions”), it is because the system has classified the computer object as malware.”

(Petition 26-27.) However, initiation of a virus alarm or virus containment action does not mean that the system has necessarily classified a particular object as malware. (*See* EX-1006 [0090], [0024]-[0030].)

146. As I explained above, Halperin discloses an embodiment in which, after sensors have logged event information, the system calculates the probability of having measured such behavior “assuming that there is no virus in the system.” (EX-1006 [0090].) Where that calculated value is lower than a predefined threshold, the

system can activate a virus alarm. (*Id.*) Accordingly, the initiation of a virus alarm merely means that a threshold has been exceeded—i.e., that the system is acting unexpectedly or abnormally. (*See* EX-1006 [0090].)

147. Similarly, Halperin discloses a variety of virus containment actions, such as “[s]uspending any or all messages sent by [a] computer,” or “[n]otifying a user at [a] computer [] of the suspicious message activity.” (*E.g.*, EX-1006 [0024], [0026]; *see also id.* [0025]-[0030].) These examples, along with the other containment actions Halperin discloses, do not require a previous classification of a particular object as malware to be effective. (*See id.*) To the contrary, where all messages sent by a computer are suspended—regardless of whether they are legitimate or suspicious—a POSITA would not understand this to be a result of having classified a particular object as malware; if it were otherwise, there would be no need to suspend *all* messages sent by the computer. And notifying a user of suspicious message *activity* does not require any classification of a particular object as malware, either.

148. A real-world example illustrates the flaw in Petitioner’s argument. For example, when a patient calls his doctor and says he is experiencing coughing, sneezing, and a runny nose, the doctor may recommend, among other things, that the patient stay home from work or school to avoid passing on the presumed infection to others. But the fact that the doctor detected symptoms and took

containment actions does not mean that he also necessarily diagnosed (or classified) the patient as having *any* virus—much less that the doctor identified the *particular* virus responsible for the patient’s symptoms.

149. Similarly, here, as I have explained above, where Halperin detects symptoms that may be the result of a virus, it initiates containment actions to prevent further spread from potentially infected computers. (EX-1006 [0047]-[0057].) However, this does necessitate a prior classification of a particular object as malware—particularly where none of the remedial containment actions require knowledge of which particular object might be malware.

150. Dr. Newman merely repeats the Petitioner’s argument and, therefore, suffers from the same flaws. (EX-1004 ¶¶ 88-90.)

151. Accordingly, a POSITA would not have understood Halperin to disclose classification of a particular object as malware merely because it teaches initiation of an alarm or containment actions after detecting potential viral activity.

B. Dependent Claims

152. Because Halperin does not render obvious all limitations of the independent claims, I understand that each of Petitioner’s grounds further fails to render obvious dependent claims 2-14, 16-28, and 30 as well.

VII. CLAIMS AND/OR LIMITATIONS NOT ADDRESSED IN THIS DECLARATION

153. It is my understanding that this declaration is being provided in support of Patent Owner's Preliminary Response. I have provided analysis of key limitations not included in the prior art. The fact that I have not analyzed particular limitations or claims in no way indicates or supports that these limitations or claims are disclosed or rendered obvious by the prior art, and I reserve the right to supplement my analysis if asked to do so.

VIII. SECONDARY OBJECTIVE INDICIA OF NONOBVIOUSNESS

154. I understand that in determining obviousness of the claimed invention, objective evidence of secondary considerations pertaining to nonobviousness must be considered, including long-felt but unsolved needs, failure of others, commercial success, etc. (*In re Huang*, 100 F.3d 135, 139 (Fed. Cir. 1996) (citing *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966).) I further understand that these indicia constitute independent evidence of nonobviousness that can be the most probative evidence in that regard "in avoiding the trap of hindsight when reviewing, what otherwise seems like, a combination of known elements." (*Leo Pharm. Prods., Ltd. v. Rea*, 726 F.3d 1346, 1358 (Fed. Cir. 2013).)

155. As demonstrated by my summary of the prior art, above, there was at least a long-felt and persistent need prior to the '389 patent to improve upon the time-consuming methods for determining whether computer objects are safe or

malware. (*See* Section III, *supra*.) Prior attempts to develop such methods were limited in their ability to make rapid determinations of whether an object was safe or malware and required extensive detailed analysis of objects and/or human administrator decisions. (*See* Section III, *supra*.) Where others failed, the '389 Patent succeeded. (*See* Section IV(A), *supra*.)

156. Thus, these secondary indicia only further support my opinion that the '389 Patent was non-obvious.

IX. CONCLUSION

157. For at least the foregoing reasons, it is my opinion that Petitioner's grounds do not render obvious any of the challenged claims of the '389 Patent.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Executed this 16th day of June, 2023 in Los Angeles.

Respectfully submitted,

A handwritten signature in blue ink, appearing to read "Nenad", followed by a horizontal line.

Nenad Medvidovic, Ph.D.