

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CROWDSTRIKE, INC.

Petitioner

v.

OPEN TEXT INC.

Patent Owner

Case IPR2023-00289

Patent No. 8,418,250

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

TABLE OF CONTENTS

I.	QUALIFICATIONS	2
II.	UNDERSTANDING OF RELEVANT LEGAL PRINCIPLES	7
III.	STATE OF THE PRIOR ART	9
IV.	The '250 Patent.....	11
	A. Summary Of The '250 Patent.....	11
	B. Claim Construction	18
	1. “automatically” (claims 1, 13, 25).....	18
	2. Additional Terms At Issue In The District Court	24
	C. Level Of Ordinary Skill In The Art.....	24
V.	OVERVIEW OF PRIOR ART	26
	A. Overview Of Kester.....	26
	B. Overview Of Honig.....	37
VI.	KESTER IN VIEW OF HONIG DOES NOT DISCLOSE OR RENDER OBVIOUS THE CHALLENGED CLAIMS	43
	A. Kester Is Non-Analogous To The '250 Patent.....	43
	B. Kester and Honig are Non-Analogous	50
	C. Independent Claims 1, 13, and 25	51
	1. Kester Does Not Disclose Or Render Obvious Limitations [1.5], [13.4], and [25.4]	53
	2. The Petition’s Combination Of Kester In View Of Honig Does Not Render Obvious Limitations [1.5], [13.4], And [25.4]	63

3.	A POSITA Would Not Have Been Motivated To Modify Kester In View Of Honig	69
D.	Dependent Claims	79
VII.	CLAIMS AND/OR LIMITATIONS NOT ADDRESSED IN THIS DECLARATION	79
VIII.	SECONDARY OBJECTIVE INDICIA OF NONOBVIOUSNESS	80
IX.	CONCLUSION.....	81

EXHIBIT LIST

Exhibit	Description
2001	Declaration of Professor Nenad Medvidovic, Ph.D
2002	<i>Curriculum Vitae</i> of Professor Nenad Medvidovic, Ph.D.
2003	<i>Webroot, Inc. and Open Text Inc., v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.</i> , Case No. 6:22-cv-00241-ADA-DTG, First Amended Complaint for Patent Infringement, Dkt. 60 (W.D. Tex., Sept. 8, 2022) (excerpted)
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	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab. et al.</i> , Case No. 6:22-cv-243-ADA-DTG, March 9, 2022 Transcript of Motions Hearing, Dkt. 133 (W.D. Tex. Dec. 15, 2022) (excerpted)
2008	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
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)

Exhibit	Description
2010	<i>Webroot, Inc. and Open Text Inc., v. Sophos Ltd.</i> , Case No. 6:22-cv-00240-ADA-DTG, Minute Entry for proceedings held before Judge Derek T. Gilliland, Notice of Electronic Filing, Dkt. 102 (W.D. Tex., Mar. 7, 2023)
2011	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab. et al.</i> , Case No. 6:22-cv-243-ADA-DTG, Scheduling Order, Dkt. 39 (W.D. Tex. Aug. 8, 2022)
2012	<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)
2013	<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)
2014	<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)
2015	<i>Ex Parte James M. Brennan, et al.</i> , Appeal 2021-003606, Application 15/285,875, Decision On Appeal Statement of the Case (PTAB Jul. 26, 2022)
2016	<i>Webroot, Inc. and Open Text Inc., v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.</i> , Case No. 6:22-cv-00241-ADA, Complaint for Patent Infringement, Dkt. 1 (W.D. Tex., Mar. 4, 2022)
2017	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab. et al.</i> , Case No. 6:22-cv-243-ADA-DTG, Plaintiffs’ SurReply Claim Construction Brief, Dkt. 147 (W.D. Tex. Jan. 6, 2023) (excerpted)
2018	Transcript of Deposition of Wenke Lee dated September 28, 2023
2019	Declaration of Nenad Medvidovic, Ph.D. in Support of Patent Owner’s Response

Exhibit	Description
2020	<i>Webroot, Inc. and Open Text, Inc. v. AO Kaspersky Lab, et al.</i> , Case No. 22-cv-00243-AdA-DTG, Opening Claim Construction Brief From: AO Kaspersky Lab et al., Dkt. 86 (W.D. Tex. Oct. 28, 2022) (excerpted)

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 8,418,250 (“the ’250 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 ’250 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 ’250 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. 8,418,250 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 ’250 patent at the time of the alleged invention. I have also been asked to provide my

technical opinions on how concepts in the '250 patent specification relate to claim limitations of the '250 patent. In reaching the opinions provided herein, I have considered the '250 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.

III. STATE OF THE PRIOR ART

27. “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 1:14-17.) 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 1:14-2:56.)

28. At the time of invention of the ’250 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 1:18-29.) “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 1:22-24.) Providers of anti-virus software push an update to their software intended to recognize and handle known malware. (EX-1001 1:24-

28.) However, these types of systems suffered from inherent delay. (EX-1001 1:30-37.) 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. (*Id.*)

29. 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 1:60-67.) When a remote computer encountered a file that was unknown to it, it would forward a request to the central computer. (EX-1001 1:64-2:4.) But 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 2:4-7.) But such an analysis had to be “carried out manually or semi-manually” by subjecting the file to a detailed analysis. (EX-1001 2:8-11.)

30. These analysis methods included, for example, emulation or interpretation, “which can still take days given the human involvement that is typically required.” (EX-1001 2:4-11.) 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 2:18-27.) “There is therefore still a considerable period of time before a new file is classified as safe or as malware.” (EX-1001 2:12-13.)

31. Thus, prior art systems “rel[ied] 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 2:57-61.)

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

IV. THE '250 PATENT

A. Summary Of The '250 Patent

33. The '250 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, 13, 25.) The '250 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 7:63-66.)

34. Unlike the methods described in the prior art, the '250 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 3:8-18.)

35. The '250 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 7:54-59, Fig. 1.)

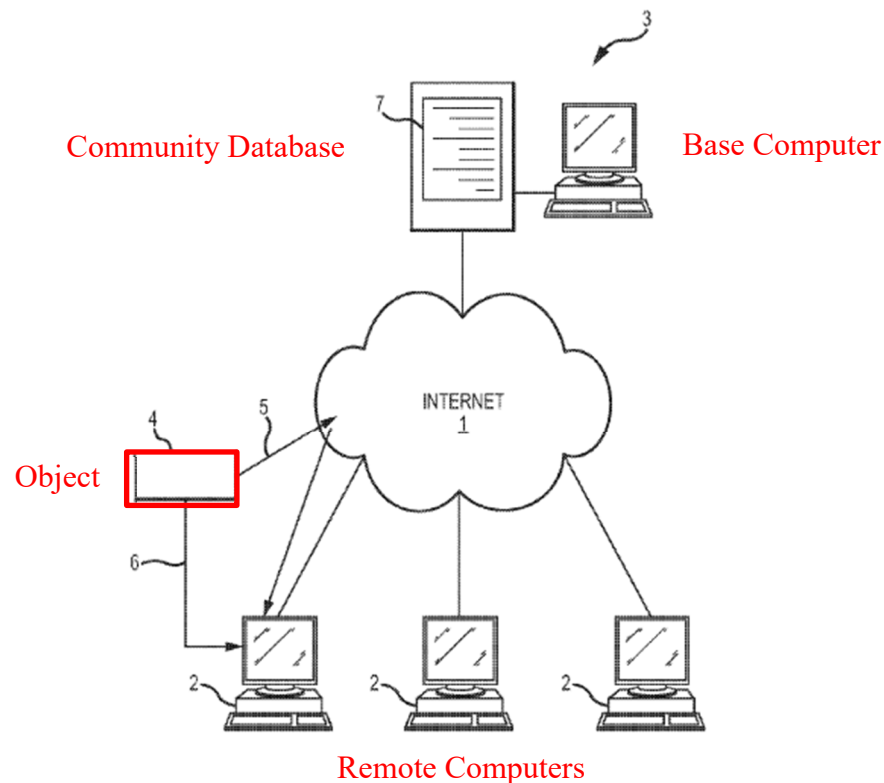


FIG.1

(EX-1001, Fig. 1, 7:54-8:16 (annotated).)

36. The '250 patent teaches that the remote computers provide data about an object to the base computer, to facilitate a comparison of data at the base computer. (EX-1001 4:43-47.) 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 4:52-61.)

37. One of the ways that this data may be provided to the base computer is in the form of a “key,” which the ’250 patent explains, allows for a quick comparison of objects and faster analysis at the base computer. (EX-1001 3:51-4:6.) “The key preferably has at least one component that represents executable instructions contained within or constituted by the object.” (EX-1001 3:51-53.) This “important preferred feature” allows a comparison to be made at the base computer of only the executable instructions of the object. (EX-1001 3:53-55.) The ’250 patent explains where objects have the same executable instructions but are differently named, this often indicates the objects are malware. (EX-1001 3:55-59.) But by using a key that represents executable instructions contained within or constituted by the object, the ’250 patent treats these objects as the “same” for the purpose of determining whether they are safe or malware. (*Id.*)

38. Figure 2 is described as “a flowchart showing schematically the operation of an example of” the ’250 patent’s method. (EX-1001 7:46-48.)

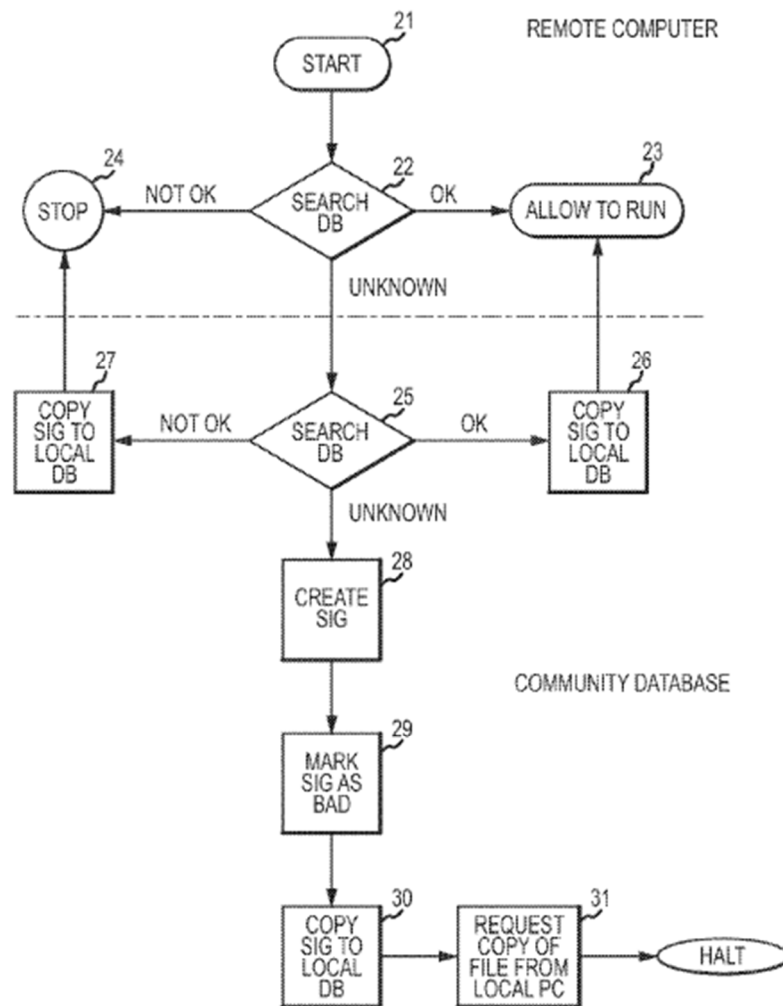


FIG.2

(EX-1001 Fig. 2, 8:16-15:5 (annotated). I describe the steps illustrated in Figure 2, below.

39. Starting at step 21, a computer object (*e.g.*, “process”) is run at a remote computer. (EX-1001 8:17-18.) At step 22, the remote computer searches its local database for a key representing that particular process, as well as any related objects and/or the event. (EX-1001 8:18-24, Fig. 2 (step 22).) If the process is known and a local key exists, the remote computer allows or stops the process based on whether

the key indicates that the process is safe or unsafe. (EX-1001 8:24-34, Fig. 2 (steps 23-24).)

40. When the object remains unknown after searching the local database, the data is passed along to the base computer. (EX-1001 8:45-47.) (This is illustrated between steps 22 and 25, as the process bridges the divide between the remote computer and the base computer. EX-1001 Fig. 2.) At step 25, the base computer searches the community database for the object's signature. (EX-1001 8:47-50, Fig. 2 (step 25).) If the base computer finds the object's signature in the community database, it may indicate to the local computer that the object is safe (EX-1001 8:60-9:30, Fig. 2 (step 26)) or unsafe (EX 1001 9:4-7, Fig. 2 (step 27)). This indication may determine whether the process is stopped or permitted to run. (EX-1001 Fig. 2, (steps 23-24, 26-27).) In either instance, the remote computer's local database may be updated with this additional information, so that the remote computer has this information immediately at hand the next time the object is encountered. (EX-1001 8:60-9:10.)

41. If the object remains unknown even after searching the community database, a signature is created for the new object, and the object is initially marked as bad or unsafe. (EX-1001 9:11-17, Fig. 2 (steps 28-29).) Once the signature has been created, it is provided to the remote computer to copy to their local database, along with a message that might instruct the remote computer not to run the object

or inform the user of the risk and offer the choice of running it; additionally, the community database may request a copy of the object itself, in case further analysis is needed. (EX-1001 9:17-25, Fig. 2 (steps 30-31).)

42. If the unknown object is allowed to run on the remote computer, the '250 patent creates a "mask" to monitor the object's actions. (EX-1001 9:26-39, 14:55-61.) "The signature of an object may comprise or be associated with a mask." (EX-1001 14:55-58.) For example, the mask parameters might indicate "the expected behaviour of the object; any external requests or Internet connections that that object might legitimately have to make or call upon the remote computer 2 to make, including details of any ports or interfaces that might be required to be opened to allow such communication, any databases, either local or over a local area network or wide area network or Internet, that may be expected to be interrogated by that object; and so on." (EX-1001 14:61-15:4.)

43. In addition to monitoring processes that are not pre-authorized, the '250 Patent discloses one embodiment in which even pre-authorized objects are monitored while running to determine when pre-authorized objects might exceed the parameters of the existing mask. (EX-1001 15:42-45, Fig 3 (steps 35 and 36).)

44. If the object's actions remain within the boundaries set by the mask, then the object is allowed to run. (EX-1001 15:45-47, Fig. 3 (step 34).) However, if the object's actions extend outside the mask, such as attempting to initiate

connections that have not been pre-authorized, then the behavior is flagged at an alert step and subsequent action may be taken. (EX-1001 15:47-51, Fig. 3 (step 37).)

45. The process is intended to undergo continuous monitoring, which allows the mask to be updated when current or future object behavior varies from prior object behavior. (EX-1001 15:64-16:3.) For example, although a process may vary from the mask's initial parameters, if there are no ill effects garnered from the unauthorized action, or if the behavior is recognized from other objects and known to be safe, then the mask may be modified to include the new behavior as a safe or allowed behavior. (EX-1001 15:9-21.) Continuous monitoring allows for similar reclassification of the object as unsafe if the object violates the parameters of the generated mask by attempting to connect to a known unsafe website or take an action known to be suspicious (*i.e.*, an action only attempted by malware). (EX-1001 15:22-25.)

46. Thus, the '250 Patent describes and claims systems and methods that provide more efficient and accurate determinations of whether an object is safe or malware. The '250 patent's "sophisticated pattern analysis" allows for "rapid determination of the nature of the object to be made, without requiring detailed analysis of the object itself." (EX-1001 3:8-18.) Its methods and system decreased reliance on signature verification and negated the need for manual, time-consuming "deep analysis" of an object. (*Id.*) This improved system performance and reduced

the number of resources required, allowing for a new method of computer object classification that is rapid, efficient, and accurate.

B. Claim Construction

47. Petitioner construes one claim term in its Petition, addressed below.

1. “automatically” (claims 1, 13, 25)

48. In the Petition, Petitioner asks the Board to construe claims to have their plain and ordinary meaning, “except where the claims recite ‘automatically,’ which the Petition assumes requires a step performed ‘without human involvement.’” (Petition 9.)

49. I understand that in a parallel proceeding in the Western District of Texas, in its March 16, 2023 claim construction order, the District Court construed the term “automatically” as having its “[p]lain-and-ordinary meaning which includes, but is not limited to, ‘without human involvement.’” (EX-2012 4.)

50. The District Court’s construction of automatically is consistent with my understanding of the term as used by the ’250 patent, for the reasons I discuss below. Therefore, in my analysis of challenged claims and prior art, I apply the plain and ordinary meaning of all claim terms.

51. The term “automatically” appears in the independent claims of the ’250 patent:

1. A method of classifying a computer object as malware, the method comprising:

....

automatically generating a mask for the computer object that defines acceptable behaviour for the computer object, wherein the mask is generated in accordance with normal behaviour of the object determined from said received data;

....

automatically monitoring operation of the object on the at least one of the remote computers;

....

(*E.g.*, EX-1001 cl. 1.) While the '250 patent does not define “automatically” explicitly, its use of the term in the specification is consistent with the term’s plain and ordinary meaning. For example, the specification describes a means of “immediately and automatically mark[ing]...as bad (or good)” a child object, based on its knowledge of the parent object and the relationship between the two objects. (EX-1001 3:31-35.) That is, a relationship between objects may exist (*e.g.*, one object may have been created by another object and thereby be seen as the “child” object of the “parent” object) and the system’s determination of bad versus good necessarily and inevitably occurs as a result of that relationship. (*Id.*) In this instance, the known relationship is what leads the method to automatically mark the child object as bad or good, such that human decision is not necessary. That is, the specification uses the term automatically to describe actions that necessarily and inevitably occur as a result of a particular circumstance—consistent with the plain

and ordinary meaning—though a human may have been involved somewhere along the way.

52. The District Court’s construction is also consistent with the applicant’s use of the term in the prosecution history, which I have reviewed. There are at least two examples in the prosecution history that are illustrative. First, the examiner made an initial rejection (eventually withdrawn) based on a combination of two references: U.S. Patent Publication No. 2003/0177394 (“Dozortsev”) and U.S. Patent Publication No. 2002/0099952 to Lambert (“Lambert”). (EX-1002 565.) The examiner explained that while Dozortsev did not teach the mask-related limitations, Lambert allegedly taught these limitations. (EX-1002 566-567.)

53. In addition to raising other limitations not disclosed by Lambert, the applicant explained that “Lambert teaches that an administrator classifies the software using the criteria and sets the rules for the token. This is clearly manual, and not automatic generation of a mask.” (EX-1002 585-586.)

54. In another rejection (eventually withdrawn), following some amendment to the claims, the examiner alternatively relied on U.S. Patent Publication No. 2005/0210035 to Kester (“Kester”)—the same reference relied upon in the petition—as allegedly disclosing the limitations not present in Dozortsev. (EX-1002 610.) The examiner explained that while Dozortsev did not disclose data about an object’s behavior, automatically generating a mask in accordance with

normal behavior, or applying such a mask, Kester allegedly taught these limitations. (EX-1002 611-612.)

55. However, in addition to raising other limitations not disclosed by Kester, the applicant explained to the examiner that Kester relies extensively on human decision. (EX-1002 632-633.) The applicant stated:

Details of network access requests are sent from the Workstation Management Module 200 to the Application Server Module 102. A Network Administrator at the Application Server Module 102 makes a decision as to what network access is allows. Additionally, a human analyst at the Application Database Factor 110 can decide what network access is allowed. The human reviewer can analyze the network access data to determine a category for the application, e.g. expected or unexpected network access. These can then be associated with policies or rules. As part of this process, the user interface 302 can display to the reviewer the ‘expected network activity’ for a program, which can be derived from previously seen behavior for the program. This information helps the human reviewer analyze the application and determine its expected behavior. The human reviewer then decides what behavior is allowed for the application. These determinations are then fed back to the hash/policy table 204.

(EX-1002 632.) And as further explained by the applicant:

This is clearly different from the claims of the present invention. The present invention is concerned with a method of classifying a computer object as malware. As part of this process, a mask is automatically

generated defining acceptable behavior for the object based on the normal behavior of the object. The automatically generated mask is used to automatically monitor the operation of the object and to reclassify the object as malware if the behavior extends beyond the mask. *A human decision is not necessary to generate the mask or monitor the operation.*

(EX-1002 633 (underline in original, bold italics added).)

56. I understand that Petitioner states that “the file history is [allegedly] unequivocal that the term ‘automatically’ excludes human involvement.” (Petition 8.) I note that Petitioner’s expert, Dr. Lee, “take[s] no position as to the correct interpretation of the claims with respect to human involvement,” and instead “assume[s] that human action is excluded and that the automatic processes must be performed entirely without human involvement.” (EX-1003 37-38.) I respectfully disagree with Petitioner’s interpretation of the applicant’s statements.

57. While the applicant explained that, in the ’250 patent, “human decision is not necessary to generate the mask or monitor the operation,” the applicant makes no statements that would lead a POSITA to conclude that an automatic process must be performed “*entirely* without human *involvement*.” (See EX-1002 633.) The distinction, here, can be seen from a comparison of the ’250 specification example with Kester. In the ’250 patent example discussed above, a child object is automatically marked as good or bad based on its relationship with another object

that is good or bad. (EX-1001 3:31-35.) That is, no human decision is necessary to the step of determining whether the child object is good or bad.

58. But Kester discloses no other means for determining an application's expected network behavior or categorizing an application. (EX-1002 632-633; Section V(A), *infra.*) Even if Kester discloses certain PC-based tools, which might supply the human analyst with data on previous behavior or "hints," if the human analyst does not decide what the expected network behavior is for the application or categorize the application, the determination or categorization will simply never take place. (EX-1002 632-633.) That is, human decision is necessary to the step of determining expected network behavior or necessary to categorizing an application according to the teachings of Kester.

59. From a POSITA's perspective, Petitioner's interpretation would be problematic, particularly in the context of software and computer-implemented methods, because its use of the term "involvement" is vague. For example, all computer-implemented methods or systems inherently require at least some degree of human "involvement," because without a human to write the code or build the machine, no such method or system could ever exist.

60. For the reasons stated above, it is my opinion that the District Court's finding that the term "automatically" should be afforded its plain and ordinary meaning is consistent with a POSITA's understanding of the term in the context of

the claims, specification, and prosecution history. Accordingly, I apply the plain and ordinary meaning of “automatically” in my analysis of the prior art.

2. Additional Terms At Issue In The District Court

61. I understand that in the District Court’s March 16, 2023 claim construction order, the Court determined that no terms of the ’250 patent were indefinite and that all terms should be given their plain and ordinary meaning. (EX-2012 3-5, 8.) Consistent with that order, I apply the plain and ordinary meaning of all claim terms.

C. Level Of Ordinary Skill In The Art

62. I understand that certain issues relating to the validity of the ’250 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.

63. A POSITA at the time of the invention of the '250 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.

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

As of June 30, 2005, a person having ordinary skill in the art (POSITA) would be a person having a bachelor's degree in computer science, computer engineering, or an equivalent, as well as two years of industry experience and would have had a working knowledge of host monitoring systems, software security analysis, and dynamic malware analysis. Additional graduate education could substitute for professional experience, or significant experience in the field could substitute for formal education.

(Petition, 9.)

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

66. Based on either standard, I was at least a person of ordinary skill in the art at the time of the invention of the '250 patent.

V. OVERVIEW OF PRIOR ART

A. Overview Of Kester

67. Kester teaches that its “invention is related to computing devices and, more particularly to monitoring and controlling application files operating thereon.” (EX-1004 [0003].) Kester explains that the increase in Internet connectivity poses risks to employers, because employees are able to download and install illegal or unlicensed applications themselves, run processes that monopolize network bandwidth, and/or expose the employer to risk to sensitive data. (EX-1004 [0005].) For example, instant messaging (IM) can increase productivity by facilitating file transfers and communication, but can also pose a risk to sensitive data. (*Id.*) As another example, “streaming media is a growing concern because of its drain on network bandwidth.” (*Id.*)

68. While software was available to manage how employees access the Internet in the workplace, Kester sought to provide a solution capable of managing “the broader intersection of employees with their computing environments.” (EX-1004 [0006].) Accordingly, Kester discloses a system and method for controlling both the launch of applications and network access requests. (*Id.* [0040]-[0041].)

69. Consistent with Kester's goal of controlling the "broader" intersection of employees with their computing environments, Kester teaches a method that is capable of controlling a wide variety of application categories. (EX-1004 Fig. 4A (listing 44 exemplary categories).) Figure 4A lists at least 44 exemplary categories of applications:

FIG. 4A

Parent group	Category	Security VC	Liability VC	Productivity VC
System	Operating Systems			
	Device Drivers			
	File Management			
	Installers			
	Miscellaneous Utilities			
Access/Privacy	Anti-virus Software			
	Authentication			
	Encryption			
	Firewalls			
	Hacking	X	X	
	Remote Access			
	Spyware	X		
	System Audit			
Productivity	Contact Managers			
	CRM			
	Data Warehouse, Analytics, Reporting			
	Database			
	Document Viewers			

	ERP/SCM			
	Graphics			
	Infrastructure			
	Presentation			
	Project Managers			
	Proprietary			
	Reference			
	Search/Retrieval/Knowledge Management			
	Software Development			
	Spreadsheets			
	Suite/integrated			
	Web/Desktop Publishing			
	Word Processing			
Communication				
	Collaboration			
	Dedicated Browsers			X
	Email			
	Instant Messaging			
	P2P Filesharing	X		
	Telephone/Conference/Fax			
	Web Browser			
Audio/Video				
	Media Players			X
	Imaging			X
Entertainment				
	Adult	X		
	Gambling	X		
	Games			X
Malware				
	Malicious Applets and Scripts			

(EX-1004 Fig. 4A, [0081]-[0082].) Consistent with Kester's acknowledgement of risks associated with communication-related applications, Kester discloses communication-related categories, such as "Instant Messaging," "Email," "P2P Filesharing" and others. (*Id.* Fig. 4A, [0005].) Consistent with Kester's acknowledgment of bandwidth problems with streaming applications, Kester

discloses audio/video-related categories, such as “media players” and “imaging.”
(*Id.*)

70. Both Petitioner and Dr. Lee characterize Kester as a method for classifying a computer object as malware. (Petition 10-11; EX-1003 38-39, 41-42.) But it is important to understand that Kester is not directed to the ability to determine whether a process was safe or malware. Rather, as I have explained above, Kester’s goal was to provide employers with control across a large variety of applications. (EX-1004 [0006], Fig. 4A.)

71. Malware is merely one of the seven exemplary parent groups of categories, and contains a single exemplary category of “malicious applets and scripts.” (EX-1004 Fig. 4A.) This is the sole mention of malware in Kester, which contains no further description of what should be categorized as malicious and/or what policy should be implemented for this category. (EX-1004 Fig. 4A, [0082].) The majority of categories depicted in Kester’s embodiment are dedicated to monitoring productivity, communication, or privacy. (*Id.*)

72. Kester’s method is focused on controlling when a user or workstation is permitted to launch an application and when and how the application can access a network. (EX-1004 Abstract, [0040]-[0042].) Kester discloses a “workstation management module” (“WMM”) that can detect the launch of an application on a workstation and/or a network access request and determine an access privilege.

(EX-1004 [0040]-[0041].) Recognizing that risk may be particular to the individual user or particular workstation, Kester teaches that the access privilege may be unique to the workstation and/or user. (EX-1004 [0042], [0057] (“module 208 applies the policy(s)/rule(s) associated with the hash that matches the application and/or the user requesting the application or network access”), [0059] (“module 208 can stop execution or network access for the application depending on a policy associated with the user at this workstation”).)

73. For example, a user (and/or their workstation) in the finance department may have access to an application that can access and adjust payroll, whereas another user in a different department may be subject to a different policy regarding that or a similar application. *See, e.g.*, EX-1004 [0057] (“module 208 applies the policy(s)/rule(s) associated with the hash that matches the application and/or the user requesting the application or network access”), [0059] (“module 208 can stop execution or network access for the application depending on a policy associated with the user at this workstation”).

74. Figure 1 of Kester illustrates an example of multiple workstations (1 through N) linked together on a local area network 100, with each workstation equipped with its own WMM:

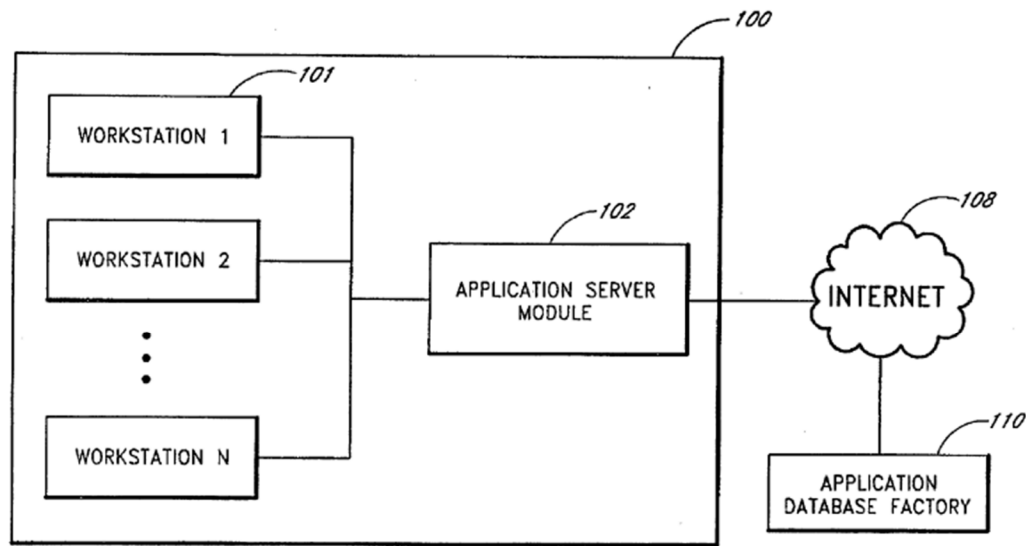


FIG. 1

(EX-1004 Fig. 1.) Kester teaches that each WMM may include a hash/policy table that allows determination of an appropriate policy for each application and/or request. (EX-1004 [0039].)

75. The hash/policy table can include a list of application names, publishers, suites, hashes, ports, protocols, I.P. addresses, categories, and rules or policies associated therewith. (EX-1004 [0054].) For example, in one embodiment, the table comprises a list of hash values associated with corresponding policies, as well as categories that are associated with the hash values and/or policies. (*Id.*) In another embodiment, the table includes names/publishers/suites or other properties which identify the application instead of hash values. (*Id.*) And in another embodiment, the table includes the port/I.P. address/protocol or other properties which identify the applications in the hash/policy table. (*Id.*) Kester's ultimate goal

is that, when a user launches an application or when the application or user requests access to the network, the policy which corresponds to that application or user can be easily identified to determine what access privileges apply to the request. (EX-1004 [0055].)

76. For example, when the WMM detects the launch of an application or a request to access a network, a “hash” or “application digest” is created, based on unique properties related to the application or user requesting access. (EX-1004 [0140]-[0142], Fig. 14 (step 1404), [0051].) The WMM then searches the hash/policy table for a corresponding hash that matches the network request. (EX-1004 [0053], [0142], Fig. 14 (step 1406).) If a match is found, then the policy associated with the hash is applied to the application in response to the network access request. (EX-1004 [0143], Fig. 14 (step 1410).)

77. However, prior to the WMM consulting the hash/policy table, Kester requires that a human reviewer make certain “predetermined associations” between the application/request and its expected network activity, any applicable categories, and a policy. (EX-1004 [0040] (“predetermined association between the launched application and one or more categories”), [0043] (“predetermined association between the application and an expected network behavior or activity”), [0072] (“The network administrator can...select or create access privileges/policies/rules for users, workstation, and/or groups of users/workstations.”).)

78. Kester discloses an “application analyst’s classification module” with which the human reviewer may interface to determine these associations. (EX-1004 [0178].) Figure 20 of Kester illustrates “a process for analyzing the network access data associated with an application at the application database factor.” (EX-1004 [0178].) In particular, Kester describes two different determinations. (*See generally* EX-1004 [0178]-[0187].)

79. For one of these determinations, after the reviewer is provided with a list of applications from the “uncategorized application database” and/or “network access database,” the reviewer determines the appropriate category or categories of the application. (EX-1004 [0178], [0134].) Figure 13 further elaborates on this specific determination of categorizing an application. (EX-1004 [0134], Fig. 13.) The application analyst classification module displays the application and any related data on the graphical user interface (GUI) for the human to review. (EX-1004 [0134].) “[T]he human reviewer uses the application, related information, and any Internet information to research the application.” (EX-1004 [0135].) “The human reviewer can further review documents, specifications, manuals, and the like to best determine the category or categories to associate with the application.” (*Id.*) The reviewer then categorizes (or classifies) each application using this evidence, “any hints from the related information,” and/or other research. (*Id.*)

80. As I explained above, this categorization determination could include any one or more of the 44 exemplary categories disclosed by Kester related to “productivity,” “audio/video,” “communication,” etc. (EX-1004 Fig. 4A.) While not included in Figure 4A, Kester further discloses two additional categories: “expected or predetermined network access” and “unexpected network access.” (EX-1004 [0178].) Thus, Kester may use the term “expected” or “predetermined” when referring to the general category of expected network access. (*See id.*)

81. Returning to Figure 20, for the other determination, Kester teaches that the human reviewer again reviews the application and any related information on the application analyst’s classification module’s GUI “to determine an expected or allowed network activity.” (EX-1004 [0179]-[0182], Fig. 20 (steps 2004, 2006).) The network activity of the same application running on different applications can be weighted in a predetermined manner. (EX-1004 [0183].) Kester explains that the “expected network activity for a given application can include one or more network attributes,” such as a specific protocol, a specific I.P. address, and a specific access port. (EX-1004 [0184].) Once the human reviewer has determined this expected network activity, the WMM stores the expected network activity in the hash/policy table. (EX-1004 [0183].) Kester teaches that for the same application, network attributes may be different for each attempted access. (EX-1004 [0185].) That is, Kester also uses “expected” or “predetermined” to refer to the expected

network activity/attributes that are unique to each entry of the hash/policy table. (*See* EX-1004 [0185].)

82. Thus, the network administrator determines a policy specific to each combination of network attributes and “[e]ach combination of the one or more attributes can be associated with one or more categories.” (EX-1004 [0185]-[0186].) And as I discussed above, this policy or category determination may be specific to the user or workstation running the application and/or making the network request. (EX-1004 [0042], [0186].)

83. Kester provides an illustrative example to help put these two different uses of “expected” within context:

For example, a first entry in the application inventory database 103 corresponds to the application when the application is operating in a predetermined manner. These expected features relate to normal network activity by the application. A second entry corresponds to the same application hash but different parsed properties or features associated with the application when the application is not operating in a predetermined manner.

(EX-1004 [0070].)

84. Thus, a POSITA would understand that Kester first uses “expected” or “predetermined” to refer to the expected network activity (or “expected features”) that are unique to each entry in the hash/policy table. (*See id.* [0070], [0179]-[0184].) That is, if an application is operating in an expected or predetermined way,

then there will be an entry in the hash/policy table which matches the network attributes associated with that operation. (*See id.*) But Kester also uses “expected” or “predetermined” to refer to the “expected network access” category—*i.e.*, in the example of paragraph [0070], whether the features for a particular application operation are “operating in a predetermined manner” and “relate to normal network activity by the application.” (*See* EX-1004 [0070], [0178], [0134]-[0135].)

85. As another example, you could have a hypothetical application request, which is an “expected network activity”—*i.e.*, the network attributes associated with that request match the “expected” or “predetermined” network attributes that are unique to an entry in the hash/policy table. (*See* EX-1004 [0070].) This same request could also be categorized as “unexpected network access”—*i.e.*, that network attribute combination, while known and recognized, was determined by the human reviewer to be “unexpected network access” for that application. (*See id.* [0070]. [0178].) Accordingly, you would have an application or request that is both “expected network activity” and “unexpected network access.”

86. While one might question the need for categories on top of the initial determination as to whether an application matches an entry in the hash/policy table, a POSITA would understand that the variety of categories was critical to Kester’s stated goal—providing employers with a system and method for controlling applications files across the “broader intersection” of employees with their

computing environment. (EX-1004 [0006].) By allowing a human reviewer to perform an in-depth review of each application and request, within the context of the particular user(s), workstation(s), and employer's policies, the reviewer could make an informed decision of any and all applicable categories across all types of applications/requests. (EX-1004 [0006], Fig. 4A, [0081]-[0082].) That is, Kester could provide the "broader" "monitoring and controlling" invention it claimed. (EX-1004 [0003], [006].)

B. Overview Of Honig

87. Honig is directed to "an architecture and data format" that enables adaptive model generation for intrusion detection systems (IDSs). (EX-1005 1:8-13, 1:44-47.) IDSs are a form of security utilized in networked systems that typically detect intrusions by collecting an audit stream from a network or system and analyzing it to look for clues of malicious behavior. (EX-1005 1:52-57.) Data-mining IDSs, specifically, collect data from sensors monitoring aspects of a system. (EX-1005 2:9-10.) For example, "[s]ensors may monitor network activity, system calls used by user processes, or file system access." (*Id.* 2:10-12.) They extract features from the raw data stream and produce formatted data that can be used for detection. (*Id.* 2:12-14.)

88. Honig teaches that machine learning and data mining algorithms can then be used on a large set of such data (*e.g.*, "training data") to build detection

models, which are then used to evaluate new data and determine whether such data is intrusive or not. (*Id.* 2:14-19.) Thus, it follows that the type of detection model that will be used in the IDS determines the type of machine learning algorithm and the type of training data that will be needed to generate the model. (EX-1005 2:29-50.) Honig first discloses “misuses detection algorithms,” which “model known attack behavior.” (EX-1005 2:29-31.) That is, they “compare sensor data to attack patterns learned from the training data,” where a match indicates that the new data is intrusive. (EX-1005 2:31-34.) Thus, Honig explains that misuses models are typically generated by “training on a large set of data in which the attacks have been manually labeled.” (EX-1005 2:34-36.) This data is very expensive to produce because each piece of data must be labeled as either normal or some particular attack. (EX-1005 2:40-42.)

89. Honig also discloses two types of “anomaly detection algorithms,” which compare sensor data to normal patterns learned from the training data. (EX-1005 2:43-46.) Honig explains that where new data diverges from this normal pattern, the algorithm classifies the new data as an attack based on the assumption that attacks have much different patterns than do normal activity. (EX-1005 2:46-50.) Most of these “algorithms require that the data used for training is purely normal and does not contain any attacks.” (EX-1005 2:67-3:2.) Thus, this “data can

be very expensive because the process of manually cleaning the data is quite time consuming” and lots of data is required. (EX-1005 3:2-5.)

90. Honig recounts the extensive manual effort and cost involved in deploying a data mining-based IDS system, due to the need for preparing training data. (EX-1005 3:6-23.) But Honig recognized that “parts of these manual prompts [sic] can be automated, including the collection and aggregation of the data and translating it into a form appropriate for training the data mining-based detection models.” (EX-1005 3:24-27.) While Honig acknowledged that “[s]ome of the processes still require some manual intervention such as labeling the data,” those processes could still be “semi-automated.” (EX-1005 3:28-30.)

91. Accordingly, Honig discloses an “architecture” that can be used to automate certain of the processes involved in training and deploying IDSs that are related to data collection, model generation, and data analysis. (EX-1005 1:10, 4:48-57.) This architecture is depicted in Fig. 1, below.

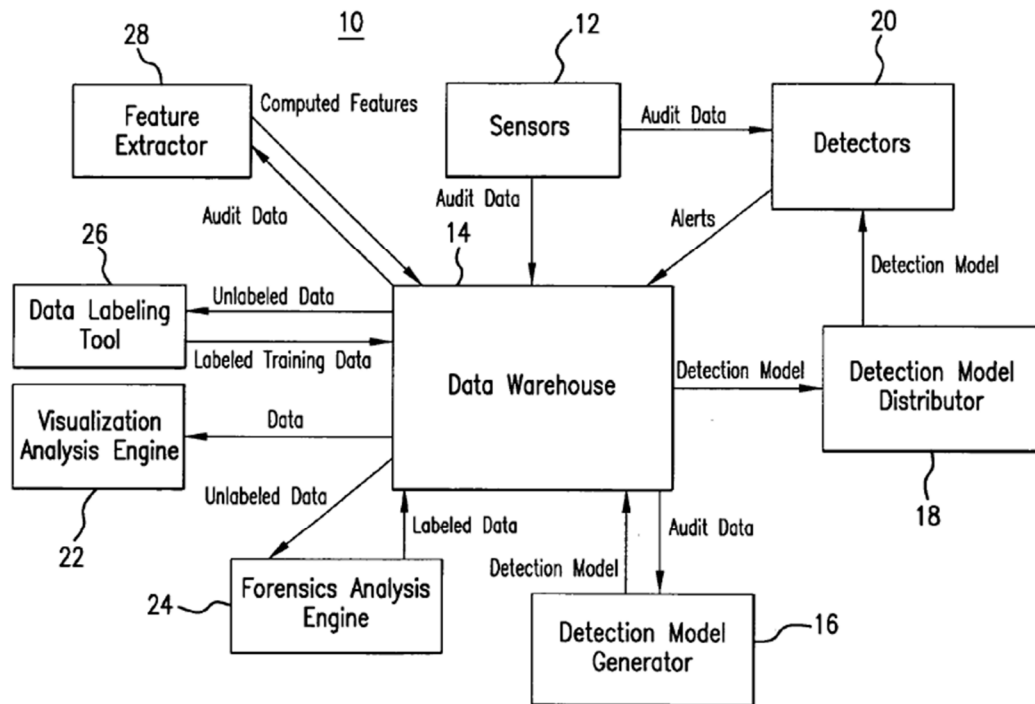


FIG. 1

(EX-1005 Fig. 1.) Honig's architecture includes: sensors and detectors (which are utilized to collect information from the monitored system); a data warehouse (utilized to store the collected data); model generators (which uses the collected data to train data mining-based detection models); model distributors (which distributes the generated models to detectors); and analysis engines to provide data analysis capabilities. (EX-1005 6:64-7:7.) Some of the advantages that Honig found in this "distributed architecture" included "a central storage of data at the data warehouse;" "a common data format" that enables reliance on different data sources; and "the support of heterogeneous data." (EX-1005 6:7-20.)

92. Honig teaches that the benefits of its architecture (and whether it has the ability to automate certain processes) is dependent upon the specific model generation algorithm that the architecture is used to support. (*See generally* EX-1005 20:33-21:38.) For example, where Honig’s architecture is used to support a misuse detection algorithm, Honig teaches that the architecture’s sensors, data warehouse, and labeling tool, can minimize the cost of labeling the data. (EX-1005 20:39-61.) Specifically, “simulated attacks can be run and the time stamps and other information about the attack recorded” by the sensors and transmitted to the data warehouse. (EX-1005 20:54-58.) Honig explains that since these were known attacks, the data labeling tool can be used to automatically label the data as attack data. (EX-1005 20:59-61.) The model generators can then use this data as training data. (EX-1005 20:60-61.)

93. For “supervised (traditional) anomaly detection” models, these algorithms need to train over normal data that contains no intrusions. (EX-1005 20:36-37, 20:64-66.) Honig teaches that the architecture’s sensors, data warehouse, and “forensics analysis engine” can “decrease the cost of creating the training set.” (EX-1005 20:65-21:22.) Specifically, sensors send the collected data to the data warehouse, where the forensics analysis engine analyzes the aggregated data to determine whether it is clean. (EX-1005 21:13-17.) Model generators can then use

the data from the data warehouse to train and generate an anomaly detection model.
(EX-1005 21:19-21.)

94. Honig further discloses “[u]nsupervised anomaly detection algorithms,” which examine unlabeled data and attempt to detect intrusions buried within the unlabeled data; therefore, Honig discloses use of such algorithms within the forensics analysis engine of its architecture. (EX-1005 21:22-38.)

95. “One specific type of model generation algorithm” Honig discloses is “Support Vector Machines (SVMs),” which can be used for anomaly detection. (EX-1005 21:39-42.) Honig describes this SVM algorithm:

The SVM algorithm is a ***binary classifier***. The operating principle for an SVM approach to intrusion detection is that data is mapped to a feature space. Inside this feature space, the SVM and a set of labeled training data are used to determine a linear decision surface (hyperplane). This surface is then used to classify future instances of data. Data is classified based upon which side of the decision surface it falls.

(EX-1005 21:46-53 (emphasis added).)

96. As a binary classifier, this SVM algorithm is limited to binary classification—*e.g.*, malicious or normal. (*Id.* 21:46-53 (“Data is classified based upon which side of the decision surface it falls.”), 8:15-18 (“models that classify activity as either malicious or normal”).) Honig explains this when it teaches that “[t]his algorithm performs best when the data is linearly separable data.” (EX-1005

22:29-31.) In non-linearly separable cases, Honig teaches that “data *must* be mapped into a higher dimension feature space *where it does become linearly separable.*” (EX-1005 22:31-34.)

VI. KESTER IN VIEW OF HONIG DOES NOT DISCLOSE OR RENDER OBVIOUS THE CHALLENGED CLAIMS

97. The Petitioner and its expert allege that the combination of Kester and Honig renders obvious every limitation of independent claims 1, 13, and 25, and this allegation underlies their challenge of the remaining dependent claims 2-12, 14-24, and 26-30.

Ground 1	Section 103(a)	Obvious over Kester in view of Honig	Claims 1-2, 7-14, 19-30
Ground 2	Section 103(a)	Obvious over Kester in view of Honig and Kennedy	Claims 3-6, 15-18

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

A. Kester Is Non-Analogous To The '250 Patent

99. I am informed that to qualify as prior art for an obviousness analysis, a reference must qualify as “analogous art,” i.e., it must satisfy one of the following conditions: (1) the reference must be from the same field of endeavor; or (2) the

reference must be reasonably pertinent to the particular problem with which the inventor is involved.

100. I understand that in determining whether two references are in the same field of endeavor, the Patent Office's classification of the prior art is some evidence of analogy or non-analogy. I further understand that comparing the structure and functions of the inventions to analyze their similarities and differences can carry even greater weight.

101. I understand that a reference is "reasonably pertinent" to a problem if it, as a result of its subject matter, logically would have commended itself to an inventor's attention in considering his problem. In determining whether a reference is reasonably pertinent, I understand that one should consider the problem faced by the inventor, as reflected – either explicitly or implicitly – in the specification.

102. It is my opinion that Kester does not satisfy either of the conditions listed above.

103. I understand that the '250 patent was classified by the USPTO as "information security" and is titled "methods and apparatus for dealing with malware." (EX-1001 p.1.) As I discussed above, the inventors of the '250 patent sought to resolve the need for detailed, manual analysis and human decision, and the inherent delay these introduced. (Section IV(A), *supra*.) The '250 patent explains that in the prior art:

[I]f the file is not known at the central computer, then the whole file is sent to the central computer where it can be analysed to determine whether it should be regarded as safe or malware. Such analysis is typically carried out manually or “semi-manually” by subjecting the file to detailed analysis, for example by emulation or interpretation, which can still take days given the human involvement that is typically required. There is therefore still a considerable period of time before a new file is classified as safe or as malware.

(EX-1001 2:4-13.) It further acknowledged the security risk, where prior art systems “rely 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.” (EX-1001 2:57-60.)

104. Accordingly, the '250 patent sought to improve upon the delays inherent in these prior art methods that relied on manual or semi-manual detailed analysis of an object and a human administrator to decide whether or not the file is safe or malware. Consistent with this goal, the '250 patent claims a method and system which “allows a rapid determination of the nature of the object to be made, without requiring detailed analysis of the object itself as such to determine whether it [is] malware.” (EX-1001 3:8-18.)

105. Thus, it is my opinion that the '250 patent's field of endeavor is the rapid determination of whether an object is malware or safe. (EX-1001 3:8-18, 1:4-5, 4:65-5:2, 8:4-7, 11:42-57, cls. 1, 13, 25.)

106. In contrast, Kester was classified as “data processing” and defines its field of invention as “related to computing devices and, more particularly to monitoring and controlling application files operating thereon.” (EX-1004 [0002]-[0003], Title, cls. 1, 23, 37, 39; *see also id.* [0008]-[0011] (describing aspects of the invention, all relating to collecting data for or updating a system which “controls” a program’s operation or network access).) For example, Kester describes various aspects of its invention, all of which relate to collecting data for or updating a system which “controls” a program’s operating or network access. (*See id.* [0008]-[0011].)

107. Dr. Lee and the petition conclude that Kester is in the field of classifying a computer object as malware based on a one-line disclosure in Kester of a single exemplary category, “Malicious Applets and Scripts.” (Petition 10-11; EX-1003 ¶62.) But this category is just one exemplary category of more than forty exemplary categories disclosed by Kester that relate to access/privacy, productivity, communication, audio/video, and other types of applications. (EX-1004 Fig. 4A [0082].)

108. While Kester discloses these categories and the fact that a human reviews data to determine whether these categories are applicable to an application, Kester makes no teachings regarding what, in the data, might indicate whether a particular category is appropriate. That is, Kester provides no guidance to a POSITA as to when or why the single “malicious applets and scripts” category should be

associated with an application or request. Kester does not even recommend a particular policy that should be applied if an application were to be categorized as malware. Thus, no POSITA would understand Kester to be in the field of malware classification, just as no POSITA would understand Kester to be in the fields of any of the other disclosed categories, such as “instant messaging” classification, “media player” classification, “gambling” classification, etc. (*See* EX-1004 Fig. 4A.) A POSITA would not have looked to Kester to inform him on the determination of whether an object is safe or malware. Thus, Kester’s field of endeavor (controlling application files) is not the same field of endeavor as the ’250 patent (rapid determination of whether an object is safe or malware).

109. I understand that during prosecution of the ’250 patent, the applicant made these same observations. (EX-1002 633.) For example, after the examiner opined in an office action that “Dozortsev and Kester are from the same field of endeavor, monitoring for malware,” the applicant explained that “Kester has no teachings relating to classifying an object as malware at all.” (EX-1002 633.) Rather, “Kester is concerned with a system to allow an employee to control what their employees can do on their computers.” (*Id.* 633.) The applicant further identified at least three reasons that “Kester is impractical for identifying malware.” (*Id.*) The examiner did not repeat his rejection based on Kester and subsequently

allowed the '250 patent to issue. Thus, this further supports my opinion that Kester is not in the same field of endeavor as the '250 patent.

110. Kester would also not have logically commended itself to an inventor's attention in considering the problems of the '250 patent, because Kester uses some of the same techniques the '250 patent sought to improve upon. As I discussed above, Kester teaches that human decision, in addition to a detailed analysis of each object, is necessary to its method for controlling application files. (Section V(A), *supra*.) For example, Kester teaches that a human reviewer conducts a detailed analysis of collected data—even Internet research or review of specifications and manuals—to determine expected network activity. (*Id.*) A POSITA would not have looked to Kester (and Kester would not have logically commended itself) to improve upon manual or semi-manual methods that relied on human administrators and detailed analysis, because Kester suffers from the same drawbacks. (*Compare* EX-1004 [0179]-[0186] *with* EX-1001 2:4-27.)

111. Kester is also not reasonably pertinent to the because Kester does not provide any teachings for how to generate a mask that defines acceptable behavior. In fact, Kester's reliance on a human analyst precludes such a mask by teaching a hash/policy table that includes expected activity that a human administrator may classify as permissible or impermissible, depending on the circumstances. As noted above, Kester is concerned with risks that may be particular to individual users or

workstations, and that access privileges may be unique to each workstation and/or user. (EX-1004 [0042], [0057], [0059]). Accordingly, any activity that the human administrator has not already “expected” is deemed “uncategorized” and sent to a human reviewer to analyze. (*Id.* [0089], [0178]-[0179]) By allowing a human reviewer to perform an in-depth review of each application and request, within the context of the particular user(s), workstation(s), and employer’s policies, the reviewer could make an informed decision of any and all applicable categories across all types of applications/requests. (*Id.* [0006], Fig. 4A, [0081]-[0082].) Thus, Kester does not provide teachings for a mask (because it only discloses a hash/policy table of “expected” activities), nor does it provide teachings for a mask that defines acceptable behavior (because the “expected” activities include both permissible actions and impermissible actions, the delineation of which must be made by a human reviewer).

112. I understand that during the prosecution history, when the examiner mistakenly opined that Dozortsev and Kester were both directed to malware, (EX-1002 612), the applicant corrected the examiner, explaining that Kester was not directed to the detection of malware and was concerned with different problems:

Kester is concerned with a system to allow an employee to control what their employees can do on their computers. This is in essence a matter of individual policy for each company and may vary between companies or even between users in a company. . . . Dozortsev is

concerned with detecting malware. Kester is concerned with allowing a company to implement policies to control what programs and network access their employees can make on their computers. These are fundamentally different problems. One of ordinary skill in the relevant art attempting to cure the deficiencies in Dozortsev would have no motivation to look to Kester for relevant teachings as the two documents are in different fields and are concerned with different problems.

(EX-1002 633-634.) The examiner subsequently allowed the '250 patent to issue, further supporting my opinion that Kester is not pertinent to the problems considered by the inventors of the '250 patent.

113. Because Kester is neither in the same field of endeavor nor pertinent to the problems considered by the inventors of the '250 patent, it is my opinion that Kester is non-analogous to the '250 patent.

B. Kester and Honig are Non-Analogous

114. It is my opinion that a POSITA would not have looked to the teachings of Honig to modify Kester (or vice versa), as these references are not in the same field of endeavor, nor are they pertinent to the same problems.

115. For instance, Kester relates to “monitoring and controlling application files” on computing devices. (EX-1004 [0002]-[0003]) In other words, Kester is directed to employee access controls, namely difficulties in controlling employee

access across various types of applications. (EX-1004 [0006]-[0007], [0040]-[0042], Fig. 4A)

116. In contrast, Honig relates to “detecting anomalies in a computer system, and more particularly to an architecture and data format for using a central data warehouse and heterogeneous data sources.” (EX-1005 1:42-47.) Honig therefore seeks to address the costs and inefficiencies associated with generating models. (EX-1005 2:65-3:12, 4:48-57.). Neither the Petition nor Dr. Lee’s declaration or deposition transcript explain how a POSITA would have these disparate references in light of the above, especially given that neither reference is reasonably pertinent to the same problems or in the same field of endeavor.

C. Independent Claims 1, 13, and 25

117. Petitioner relies on the same analysis of Kester and Kester in view of Honig to support Limitations [1.5], [13.4], and [25.4] in both Grounds 1 and 2. (Petition 27-32, 47, 51.) These limitations recite the following language:

Claim Limitation	Claim Language
[1.5]	<i>“automatically generating a mask for the computer object that defines acceptable behaviour for the computer object, wherein the mask is generated in accordance with normal behaviour of the object determined from said received data”</i>

[13.4]	<i>“automatically generate a mask for an object that is initially classed not as malware, said mask defining acceptable behaviour for the object built in accordance with normal behaviour of the object determined from said received data...”</i>
[25.4]	<i>“automatically generating a mask for the computer object, the mask defining acceptable behaviour for the object built in accordance with normal behaviour of the object determined from said received data”</i>

118. As I discussed above, I apply the plain and ordinary meaning of *“automatically,”* consistent with the district court’s claim construction order. (See Section IV(B)(1), *supra.*) Though, given the overlap between Petitioner’s and Patent Owner’s construction of this term, analysis of these limitations under either construction would not alter my conclusion.

119. First, neither Kester nor Honig, alone or in combination, disclose or suggest a mast that defines acceptable behavior of an object.

120. Second, Kester does not disclose or render obvious *“automatically generating a mask.”*

121. Third, Kester in view of Honig does not disclose *“automatically generating a mask,”* because the step in Kester which Petitioner and Dr. Lee

correlate to “*generating a mask*” is manually performed by a human reviewer, and the Petitioner and Dr. Lee point to no motivation to automate this step, particularly. Petitioner and Dr. Lee also fail to provide any detail or description as to how a POSITA might go about implementing Honig’s teachings in Kester and fail to explain how Kester in view of Honig renders obvious “*automatically generating a mask.*”

122. And fourth, it would not have been obvious to automate Kester in view of Honig, as Honig does not disclose a model generator that could be utilized in Kester because it would render Kester’s goals unattainable.

1. Kester Does Not Disclose Or Render Obvious Limitations [1.5], [13.4], and [25.4]

123. It is my opinion that neither Kester nor Honig disclose or suggest a mask that defines acceptable behavior for an object.

124. Petitioner and Dr. Lee appear to allege that a human reviewer’s determination of the expected network activity, which is stored in the hash/policy table, is the same as “*generating a mask.*” (Petition 28; EX-1003 ¶¶ 117-118.) This disclosure, however, is not the same as “generating a mask . . . that defines *acceptable behaviour for the computer object*” as claimed in the ’250 patent (EX-1001 claim 1)

125. Petitioner appears to allege that Kester’s determination of expected network activity is equivalent to a mask. (Petition 28, 33-36, 38). However, Kester’s

expected network activity, which is contained within a hash/policy table generated by a human, is not a mask that “defines acceptable behaviour” because it actually includes *any* behavior, including unacceptable behavior. As such, Kester’s hash/policy table of expected network activity merely defines expected behavior, which could include unacceptable behavior.

126. For instance, Kester teaches that when an application does not operate in a “predetermined manner,” it is placed in a database for further review so that a human can review the application’s activities. These unknown or unexpected activities may then be added by the human to the hash/policy table. As such, the hash/policy table and the human’s review and addition of activities to the hash/policy table not only do not define acceptable behavior of a computer object, but they are also agnostic to whether the behavior is acceptable or unacceptable. In fact, whether the activity of an application is operating in a “predetermined” or “expected” manner is dependent on whether that activity is contained in Kester’s hash/policy table, not whether the activity is acceptable (or, for that matter, unacceptable).

127. The claimed invention of the ’250 patent limits the mask to acceptable behavior so that future object behavior can be allowed or disallowed based on whether it falls within the mask or outside the mask. This distinction provides the benefits of avoiding the delay associated with waiting for a human to review each object and its activities and then designate whether or not those activities as

permissible. This distinction also provides the benefit of automatically reclassifying an object from “not-malware” to “malware” when the object’s behavior extends beyond the behavior defined by the mask.

128. Kester cannot provide the above benefits. For example, the ’250 patent allows an object to run when its monitored behavior falls within the claimed “mask.”

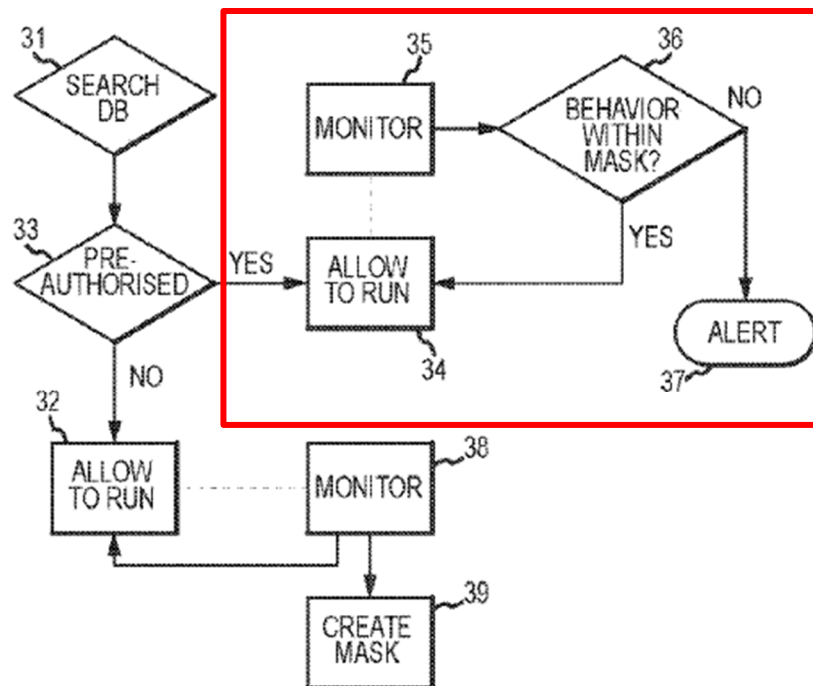


FIG.3

(EX-1001 Fig. 3; id.i cls. 1, 13, 25.)

129. However, when that behavior extends beyond the mask, the object is not allowed to run and is instead reclassified as malware. (EX-1001, cls. 1, 13, 25.) Because Kester’s expected network activity contains both acceptable and unacceptable activity, and activity extending beyond this “expected network activity” is merely enqueued so that a human can later review this activity, an

application whose behavior is within the Kester's alleged mask cannot categorically be allowed to run. (EX-1004 [0042]-[0043], [0185]-[0186].) To do otherwise would contradict Kester's stated purpose of applying predetermined policies on a workstation-by-workstation basis across all expected activity, thereby controlling its employees' interaction with the internet. (EX-1004 [0006], [0042]-[0043].)

130. Similarly, Kester teaches that an application whose behavior is not within Kester's alleged mask may categorically be allowed – this is because there is no correlation between “expected” activity and “acceptable” or “unacceptable” activity, and to categorically disallow unexpected activity could lead to a reduction of employee productivity, which would be in direct contradiction to Kester's stated intention. (*See id.* at [0059], [0144]-[0145])

131. As a result of Kester's expected network activity defining unacceptable behavior, it cannot be used as a tool for informing when an object must be reclassified as malware. Kester's disclosure provides that when an application is operating unexpectedly (*i.e.*, its hash does not match an existing hash in the hash/policy table) the application or the application's particular activity has not yet been reviewed, categorized, or associated with a policy by an administrator. (EX-1004 [0059].) Accordingly, the application is added to the “uncategorized applications database” and subsequently reviewed by an administrator. (EX-1004 [0070]-[0071].) This “unexpected” activity does not mean that the activity is

necessarily malicious or that the application should be categorized as such. Instead, Kester teaches that a human administrator must review the application and network attributes associated with its activity to determine whether the activity should be permitted or not, which employees will be permitted, and whether the application should be categorized and/or recategorized. (EX-1004 [0070]-[0072].)

132. Petitioner’s assertion that Kester discloses this limitation by disclosing “expected or allowed network activity (also referred to as expected or predetermined behavior),” (Petition 28.), improperly equates “expected” network activity with “allowed” network activity, which are not the same. Kester discloses that expected network activity may be allowed or disallowed based on the determination of a human reviewer, and that determination is in a hash/policy table. If an application’s activity extends beyond that defined in the hash/policy table, then that activity is essentially queued for further review by a human for later incorporation into the hash/policy table.

133. This is because some activity may be allowed for some employees, but not allowed for other employees. For example, an employer might want to limit certain activities and access controls to managerial employees. Accordingly, Kester discloses a human administrator who reviews applications and associated network access data to determine what access controls and associated activities to associate with different types of employees. The human administrator’s determinations are

stored in a hash/policy table, which Kester teaches may be specific to a particular workstation and uploaded to that workstation. Thus, if a manager's workstation attempts to engage in a particular activity (*e.g.*, a network access request), a lookup may be performed in the hash/policy table and the attempted activity may be allowed or disallowed based on what the human entered for that particular workstation's hash/policy table.

134. Indeed, Kester discloses that because “network attributes associated with the application may be different for each attempted access,” Kester's system “may allow a first combination of one or more network attributes while disallowing a second combination of the one or more network attributes.” (EX-1004 [0185], [0083].) As such, Kester defines activities as “expected” or “predetermined” to indicate that a human reviewer has previously reviewed them and assigned a predetermined association with a policy (to allow or not allow) that has been stored in the hash/policy table. (EX-1004 [0057]-[0059] (“If the execution launch detection module...does not find the application hash in the hash/policy table [] (for example, the application is uncategorized or the application is behaving unexpectedly)...”).)

135. According to Kester's teachings, if an application is operating in an expected or predetermined manner, *i.e.*, within the bounds of the “mask” that Petitioner alleges Kester teaches, those operations might be disallowed from continuing to run based on the policy associated with those operations and stored in

the hash/policy table provided by the human. This is consistent with Kester's stated goal (which is non-analogous to that of the '250 patent) – "to manage the broader intersection of employees with their computing environments" on a workstation-by-workstation basis. (EX-1004 [0006], [0115]-[0116].)

136. Accordingly, Kester discloses relying on a hash/policy table containing "expected" or "predetermined" activity reviewed by a human administrator, and that the hash/policy table may include acceptable activity *as well as* unacceptable activity, rather than define only acceptable behavior which I understand is required by the district court's construction of "mask."

137. It is also my opinion that Kester does not disclose or teach "*automatically generating a mask.*" As noted above, Petitioner and Dr. Lee appear to allege that a human reviewer's determination of the expected network activity, which is stored in the hash/policy table, is the same as "*generating a mask.*" (Petition 28; EX-1003 ¶¶ 117-118.) This disclosure of Kester, alone, already fails to disclose these limitations because it does not disclose "*automatically generating a mask.*"

138. Petitioner and Dr. Lee appear to concede that Kester's determination of this activity requires a human decision and that Kester, alone, cannot disclose this limitation. (Petition 28 ("Kester teaches a human reviewer... to determine the 'expected or allowed network activity'"); EX-1003 ¶ 117 ("module 302 allows the human reviewer to analyze each application and any additional data...to determine

an expected or allowed network activity”); EX-2018 6:14-20 (“Q: Do you agree that Kester teaches that it’s a human analyst who performs Kester’s categorization steps? A: . . . so to answer your question, I would say that human reviewer is involved in the categorization and recategorization.”), 8:3-9:1 (“ . . . [T]he human has to manually generate a hash.”).) However, I address this in further detail, due to the petition’s mischaracterization of Kester as merely “involv[ing] human input.” (Petition 29.) Rather, as I’ve explained above, human decision is *necessary* to Kester’s method. (Section V(A), *supra*.)

139. Kester teaches that human decision is necessary to at least three of its “predetermined associations.” The only means Kester discloses for determining an application’s expected network activity (and attributes) used to populate the hash/policy table is human decision. (EX-1004 [0179]-[0184], [0145], Fig. 14 (1416).) The only means Kester discloses for categorizing expected network access is human decision. (EX-1004 [0178]-[0181], [0187], [0134]-[0136].) And the only means Kester discloses for determining a policy to be associated with each combination of network attributes and category is human decision. (EX-1004 [0072] (“The network administrator can...select or create access privileges/policies/rules for users, workstation, and/or groups of users/workstations.”).)

140. While Kester discloses several PC-based software tools that might “enable the human reviewer,” “enhance productivity of the human reviewer,” or

“aid” the human reviewer in performing particular steps, even with these tools, Kester teaches the need for a human decision to determine expected network behavior, categorize applications, and determine an appropriate policy. (EX-1004 [0090], [0092]-[0093], [0097], [0134], [0179], [0187].) Accordingly, Kester does not disclose “*automatically generating a mask*” as required by all of the challenged claims.

141. Furthermore, Kester’s teachings, alone, would deter a POSITA from automating Kester for the same reasons the applicant provided during prosecution of the ’250 patent. As I discussed above (Section IV(B)(1), *supra*), during prosecution of the ’250 patent, the applicant overcame a rejection based on Kester by explaining that unlike Kester, human decision was not necessary to generate a mask in the ’250 patent. Critically, the applicant further explained that no POSITA would have been motivated to automate Kester’s method, “as to do so would mean that the system of Kester would no longer be able to perform its function of allowing a company to implement appropriate policies governing what their employees can do on their computers.” (EX-1002 634.)

142. As I discussed above regarding Kester, a variety of data may inform the human reviewer’s decision regarding expected network activity, categories applied to an application, and which activities and categories are acceptable for different employees. (Section V(A), *supra*; EX-1004 [0178]-[0184].) This includes, for

example, “related information and any Internet related information” (EX-1004 [0180]); “prior or contemporaneous network activity” for the same or different application (*id.* [0181]-[0182]); “documents, specifications, manuals, and the like” (*id.* [0135]); and “hints from the related information, and/or other research” (*id.*). And within the context of all this data, the human reviewer determines the expected network activity and appropriate categories for each application, while keeping in mind that the access privilege will be “unique...for the workstation 101 and/or user.” (EX-1004 [0042], [0185]-[0186].) That is, Kester discloses a method in which human decision was necessary to weigh and balance data-based information with user-specific or workstation-specific concerns and company-specific policies—and it would not be readily clear to a POSITA how such a potentially complex mental judgment could be meaningfully captured by a binary model or classifier. The USPTO’s allowance of the ’250 patent following the applicant’s arguments further supports my opinion. (Section IV(B)(1), *supra.*)

143. Neither the petition nor Dr. Lee acknowledge or address the applicant’s argument made during prosecution, nor do either provide a detailed explanation of how such a detailed mental analysis would be automated. (*See* EX-1003 ¶¶ 125-132; Petition 29-32.) Instead, the Petition and Dr. Lee appear to assume that under their proposed combination of Kester and Honig, all expected network activity would be permitted and all unexpected network activity would not be permitted. This

assumption is false, as Kester shows that expected network activity may be permitted or denied based on the circumstances of the activity in question. (EX-1004 [0042], [0185]-[0186]) I discuss this in further detail below, within the context of Petitioner's combination of Kester and Honig.

144. Accordingly, Kester alone neither discloses nor renders obvious “*automatically generating a mask.*”

2. The Petition's Combination Of Kester In View Of Honig Does Not Render Obvious Limitations [1.5], [13.4], And [25.4]

145. The petition purports to modify Kester's method in view of Honig's teachings. Before I address why a POSITA would not, in fact, have been motivated to do such a thing, I must first point out that Petitioner and Dr. Lee appear to make an error in their analysis of “*automatically generating a mask,*” and fail to explain how a combination of Kester in view of Honig would have actually rendered obvious this limitation. (*See generally* EX-1003 ¶¶ 125-132; Petition 29-32.)

146. As I have discussed above, Petitioner and Dr. Lee allege that in Kester, the human reviewer's determination of expected network activity is purportedly the equivalent of “*generating a mask.*” And in their arguments that Kester in view of Honig renders obvious “*automatically*” generating a mask, the petition and Dr. Lee rely heavily on Kester's statement that “[c]ategorization can be based upon word analysis, adaptive learning systems, and image analysis.” (EX-1004 [0092])

(emphasis added).) But in relying on this statement, Petitioner and Dr. Lee imply that it would have been obvious to automate *a different step* in Kester—specifically, the human reviewer’s determination of expected network activity. (See Petition 29-31, 21; EX-1003 ¶¶121-125.)

147. Kester does not suggest the use of an adaptive learning system to aid the determination of expected network activity (*i.e.*, the step which Petitioner alleges is “*generating a mask*”). Rather, Kester discloses adaptive learning systems (along with word and image analysis) as one of the tools which may aid a human in *categorizing* applications/requests, which Petitioner has not shown to be “*generating a mask*.” (Note that the distinction between these two steps is more fully discussed in my overview of Kester. See Section V(A), *supra*.)

148. Indeed, Petitioner and Dr. Lee appear to admit this, arguing that “*Kester* explains the process for *classifying or categorizing* applications ‘can be based upon [] adaptive learning system[.]’” (Petition 29-30 (emphasis added).) The petition asserts that “*Kester* expressly contemplates use of adaptive learning systems” for its categorization process. (Petition 31.) Dr. Lee asserts that a POSITA “would have found it obvious to look to known adaptive learning systems to perform *Kester*’s *classification and/or categorization* processes.” (EX-1003 ¶122 (emphasis added).)

149. Thus, Petitioner and Dr. Lee fail to provide any explanation or discussion of why a POSITA would rely on Kester’s statement regarding

categorization to automate *a different step*—i.e., the determination of expected network activity. And Kester in view of Honig cannot disclose or render obvious “*automatically generating a mask.*”

150. Additionally, as I previously discussed, Kester does not disclose the use of an adaptive learning system in place of human decision. (EX-1004 [0092].) Rather, Kester discloses several PC-based software tools that might “enable the human reviewer,” “enhance productivity of the human reviewer,” or “aid” the human reviewer in performing particular steps. (EX-1004 [0090], [0092]-[0093], [0097], [0134], [0179], [0187].) Kester explains that when the human reviewer interfaces with the system, he or she is provided “buttons preloaded with *algorithmically derived hints to enhance productivity of the human reviewer.*” (EX-1004 [0090] (emphasis added).) “The application analyst’s classification module 302 displays this information to the human reviewer *to aid* in categorizing the application.” (EX-1004 [0097] (emphasis added).) But even with these aids, Kester teaches the need for a human decision to ultimately determine expected network behavior, categorize applications, and determine an appropriate policy. (*Id.*) Accordingly, a POSITA would understand that even if he were to implement an adaptive learning system to aid in categorization, such a modification would not disclose or render obvious “*automatically*” generating a mask because such an adaptive learning system would only aid a human in generating the supposed mask.

151. This is particularly true where there is a lack of explanation in the petition as to what, precisely, the combination of Kester and Honig includes and/or how the combined system would function. While the petition and Dr. Lee repeatedly assert that “[i]t would have been obvious to a skilled artisan to ‘automatically’ generate (*i.e.*, without human input) *Kester’s* [alleged] mask based on the teachings of *Honig*,” neither the petition nor Dr. Lee explain exactly *what* the combination of Kester and Honig would have looked like or *how* the combination might have worked. (*See* EX-1003 ¶¶ 125-132; Petition 29-32.)

152. For example, the petition states generally that “*Honig* provides a detailed discussion of adaptive learning model generators that were known in the field.” (Petition 31.) Honig explains that “detection models can [] vary greatly.” (EX-1005 7:64-65.) Despite Honig explicitly teaching that “[t]he challenge in automating these processes is the need to support these different types of data and different types of detection models,” Petitioner and Dr. Lee fail to explicitly identify the type of detection model/algorithm that could be utilized in a newly modified Kester, how that model would operate, and how that model would be generated. (*See* EX-1005 7:64-67.)

153. As I discussed above in my overview of Honig, Honig discloses several different categories of models/algorithms, and the degree of automation possible in Honig’s architecture is dependent upon which type of model/algorithm is

implemented and what training data is necessary. (Section V(B), *supra*; EX-1005 20:33-21:38.) Dr. Lee mischaracterizes Honig when he states that “the model generation algorithms are ‘fully automated.’” (EX-1003 ¶ 124.) In fact, Honig teaches that “[t]he process for converting the data into the appropriate form for training is fully automated.” (EX-1005 7:19-20.) But which steps will be automated within the model generation process is, again, dependent upon the type of model/algorithm being implemented. (See EX-1005 20:33-21:38.) Yet neither Petitioner nor Dr. Lee identify which type of model/algorithm a POSITA would be motivated to use in the combination of Kester and Honig.

154. After completing most of his analysis, Dr. Lee references “the SVM algorithms of Honig” for the first time in his conclusion. (EX-1003 ¶.) But Honig discloses both a “standard SVM algorithm” (*i.e.*, “a supervised learning algorithm”) and an “unsupervised variant” of this SVM algorithm, each of which has unique training data needs due to the unique type of model these algorithms generate. (EX-1005 23:33-48; *see also* EX-2018 35:8-11 (“But, again, might talk about labeling, right, creating label. That’s part of the supervised machine learning process. If it’s unsupervised, you don’t get label data.”), 36:6-38:3.) Furthermore, Dr. Lee generally identifies other types of models, while failing to explain their relevance or how or why those models would be incorporated into Kester either alone or in addition to one of Honig’s models. (EX-1003 ¶¶ 129-130.) Neither the Petitioner

nor Dr. Lee specify what type of algorithm a POSITA would be motivated to use in Kester, which type of training data would be necessary, or how Kester's method would be further modified to supply and label such training data.

155. As another example, the petition describes Honig's system as "scalable" with components that can be "plugged" into the system architecture. (Petition 31.) Yet, neither Petitioner nor Dr. Lee identify which component(s) might be plugged into Kester, which components would be necessary to support the unidentified detection model in Kester, and how these components would function within the context of Kester's system.

156. As I discussed above, the components of Honig's architecture that are necessary to automate certain steps depend on the model/algorithm being implemented. (Section V(B), *supra*.) Dr. Lee acknowledges this when he explains that "[a]utomated model generation trains detection models from data stored *in the data warehouse*." (EX-1003 ¶ 123.) Dr. Lee also references "automated data collection," "model generation," and "forensic analysis" (EX-1003 ¶¶ 123-124), which means that the combination would at least need the sensors and data warehouse (for automated data collection), detection model generator (for model generation), and the forensics analysis engine. (EX-1005 Fig. 1, 8:14-24.) But neither Petitioner nor Dr. Lee are clear as to what modules a POSITA would

purportedly be motivated to incorporate or how such modules would function in Kester's system.

157. Because the petition does not explain how the modified version of Kester (*i.e.*, the combination of Kester and Honig) would function, it does not show that such a combination would render obvious “*automatically generating a mask.*”

3. A POSITA Would Not Have Been Motivated To Modify Kester In View Of Honig

158. Additionally, a POSITA would not have been motivated to implement the teachings of Honig in Kester, because the two references are non-analogous, because Kester's teachings would have deterred a POSITA from doing so, and because Honig discloses a “binary” classifier.

a. Kester and Honig are non-analogous.

159. Kester and Honig are neither in the same field of endeavor nor pertinent to the same problems; thus, a POSITA would not have looked to Honig as relevant to Kester's methods.

160. As I discussed above, Kester's field of endeavor is monitoring and controlling application files on computer devices and Kester sought to address the need for a method for managing all the ways employees interacted with their computing environment to reduce liability risk. (Section VI(A), *supra*; EX-1004 [0002]-[0003].)

161. In contrast, Honig’s invention relates to “detecting anomalies in a computer system, and more particularly to an architecture and data format for using a central data warehouse and heterogeneous data sources.” (EX-1005 1:42-47.) Honig sought to address the cost and inefficiencies associated with model generation. (EX-1005 2:65-3:12, 4:48-57.) As Honig has nothing to do with controlling application files, it is non-analogous to Kester and no POSITA would have considered combining their teachings.

b. Kester’s discussion of an adaptive learning system would not have motivated a POSITA to modify Kester.

162. As I briefly discussed above, Petitioner and Dr. Lee rely on a single sentence from Kester regarding “adaptive learning systems” to argue that Kester “expressly contemplate[s]” automating its processes and that, therefore, a POSITA would have been motivated to look to Honig. (Petition 29-30; EX-1003 ¶ 125.) But Kester merely states that *categorization*—specifically “can be *based upon* word analysis, adaptive learning systems, and image analysis.” (EX-1004 [0092].) That is, a POSITA would not understand Kester to be suggesting the use of these tools for the determination of expected network activity; indeed, it is unclear how such tools could be used to determine expected network activity and neither the petition nor Dr. Lee explain.

163. Even for the categorization step that Kester actually references in this sentence, Kester merely teaches that such determination can be “*based upon*” an

adaptive learning system. (EX-1004 [0092].) This is consistent with Kester's repeated teachings that "PC-based tools" may be used to "enable the human reviewer," "enhance productivity of the human reviewer," or "aid" the human reviewer in its determinations. (EX-1004 [0090], [0092]-[0093].) For example, Kester teaches that its graphical user interface (GUI), with which the human reviewer interacts, "can include buttons preloaded with *algorithmically derived hints* to enhance productivity of the human reviewer." (EX-1004 [0090] (emphasis added).) Thus, a POSITA would understand that even if one were to implement an adaptive learning system according to Kester's teaching, such a system would merely provide hints to aid or enhance productivity when the human reviewer categorizes applications/requests. (*See id.* [0090], [0092].) However, human decision would still be necessary to make the actual determination of applicable categories. (EX-1004 [0178].) Kester discloses no alternative to human decision in determining expected network activity, categorizing applications/requests, and assigning an appropriate policy for a user or workstation.

164. Accordingly, Kester does not suggest, nor would a POSITA consider ways of automating the determination of expected network activity or the categorization of applications.

c. A POSITA would not be motivated to implement Honig's binary classifier in Kester

165. As I explained above, both Petitioner and Dr. Lee fail to explain how a POSITA would implement Honig's teachings in Kester, as well as what the resulting combination would look like or how it would function. (*See* Section VI(B)(2), *supra.*)

166. It is possible this lack of explanation is a result of the fact that a POSITA would not be motivated to implement Honig's teachings, because Honig discloses a "binary classifier," which would be of no use in Kester's system, as it would significantly reduce Kester's capabilities of applying policies to various activities on a case-by-case, workstation-by-workstation, and/or user-by-user basis. (EX-1005 21:46-47.)

167. As I explained above, Kester sought to improve upon "traditional filter systems" by providing a "new solution[] to manage the broader intersection of employees with their computing environments." (EX-1004 [0006]-[0007]; Section V(A), *supra.*) In order to achieve this goal, Kester permits a human administrator to control applications/requests through the use of a large variety of categories, one or more of which may be applies to an application/request to aid in the determination of a controlling policy. (EX-1004 [0081]-[0082], Fig. 4A.) As a reminder, these categories span everything from instant messaging and email to media players and games. (EX-1004 Fig. 4A.)

168. That is, Kester’s method for controlling all types of application files cannot be realized through a binary determination. A binary determination can be thought of as “this or that,” “match or no match,” “A or B,” “yes or no.” (See EX-1005 21:46-53.) The ’250 patent discloses one embodiment in which the binary determination is safe or unsafe/malware. (EX-1001 11:42-52.) Honig also discloses an exemplary binary determination of malicious or normal. (EX-1005 8:15-18.) In contrast, Kester’s method provides the ability to control all applications related to the system, and therefore discloses more than forty categories that can be applied to associate each category individually with the appropriate policy. (EX-1004, Fig. 4A.)

169. While, as I stated, Petitioner and Dr. Lee are vague as to what type of model/algorithm a POSITA would purportedly utilize in Kester, Dr. Lee references “the SVM algorithms of Honig.” (EX-1003 ¶ 128.) But Honig explicitly states that “[t]he SVM algorithm is a *binary* classifier.” (EX-1005 21:46-47.) Honig explains:

The operating principle for an SVM approach to intrusion detection is that data is mapped to a feature space. Inside this feature space, the SVM and a set of labeled training data are used to determine a linear decision surface (hyperplane). This surface is then used to classify future instances of data. Data is classified based upon which side of the decision surface it falls.

(EX-1005 21:47-53.) At his deposition, Dr. Lee further admitted that Honig teaches a binary classifier. (EX-2018 49:22:50:9.) Because such a classifier is limited to a binary determination, Honig teaches that “[t]his algorithm performs best when the data is linearly separable data.” (EX-1005 22:29-30.)

170. If it is Petitioner’s position that a POSITA would generate a binary model of “expected network activity” to replace the human’s determination of network attributes for the hash/policy table (which the petition still fails to explain or support (Section VI(B)(2), *supra*)), neither Petitioner nor Dr. Lee make any effort to explain how this model would function within or in place of Kester’s hash/policy table. For example, neither explain how such a model would function in conjunction with the categories and associated policies. Neither explain how such a model could be customized for specific workstations and users—one of the key capabilities in Kester’s system. (EX-1004 [0042].) That is, a human decision would still be necessary to determine applicable categories and the appropriate policy, to the extent that remained possible in tandem with the model.¹ (*See id.* [0042], [0048], [0186].)

¹ I question whether this would be possible, because the human reviewer would have no insights or understanding of why the machine learning algorithm made certain determinations. One of the well-known limitations of utilizing such algorithms is that the reasons an algorithm came to a certain determination are not readily apparent

171. Furthermore, as I previously discussed, Kester explicitly teaches the use of any adaptive learning system, image analysis, etc. as an *aid*, but still relies on human decision to make the ultimate determination. (Section VI(B)(2), *supra*. There is no reason a POSITA would have been motivated to spend the time and money necessary to modify Kester when human decision would still be necessary. Accordingly, the petition fails to explain how the combination would render obvious “*automatically generating a mask*” and a POSITA would not have been motivated, due to the incompatibility of Honig’s classifier and Kester’s hash/policy table, to combine their teachings.

172. If it is Petitioner’s position that a POSITA would generate a binary model of the “expected network access” category based on Honig’s teachings (which the petition still fails to explain or support (Section VI(B)(2), *supra*)), a POSITA would not be motivated to do so due to the severe limitations of a binary model. Kester sought to provide a system and method capable of handling multiple questions with a potentially unlimited number of possible outcomes, such as

to a human reviewer (hence the current, large push across industry and academic research for “explainable AI”). Without understanding the determination, it would be impossible for the human reviewer to then apply the correct categories and/or policies.

determining what type of network activity is occurring, who is attempting the activity, who is allowed to attempt that activity, and under what circumstances that activity should be allowed, among others. In contrast, a binary model would be restricted to the application of a single category—*e.g.*, “expected network access” or not. That is, the resulting system would either be incapable of determining whether to apply any of the other categories Kester discloses (including “malicious applets and scripts” or “unexpected network access”) or a human reviewer would be required to work in tandem with the binary model to the extent that was even possible.² (*See* EX-1004 Fig. 4A.) A POSITA would, again, not be motivated to go to the time and expense of modifying Kester when it would *reduce* Kester’s capabilities.

173. Petitioner and Dr. Lee do not appear to understand the limitations of Honig’s model, because they assert that such a model could allegedly both: (1) be determine whether something should be categorized as expected network access or not; and (2) determine whether something should be categorized as malware or not. (Petition 30; EX-1003 ¶¶ 122, 124.) As Honig’s SVM algorithm is a *binary*

² Again, this is likely not possible or would at least be ill-advised, because without understanding the determination made by a classifier, the human reviewer would be unequipped and unable to apply the correct categories and policy.

classifier capable of only a binary decision, it would not be capable of both. (*See* EX-1005 21:46-53.) Furthermore, neither the petition nor Dr. Lee have made the assertion that applying Kester’s “malicious applets or scripts” category is related to “*generating a mask*,” so these statements appear irrelevant and only intended to mischaracterize Kester as analogous to Honig or the ’250 patent.

174. For the reasons above, a POSITA would not have been motivated to implement Honig’s binary classifier, because doing so would have severely *reduced* the abilities of Kester’s method, making Kester’s stated goal unattainable.

175. Additionally, as I previously discussed, Honig teaches that each specific algorithm requires a specific type of training data used to train the model:

Machine learning and data mining algorithms are used on a large set of such data (e.g., “training data”) to build detection models....This data is very expensive to produce because each piece of data must be labeled as either normal or some particular attack.

(EX-1005 2:14-42, 2:67-3:13.) “In the case of anomaly detection, each instance of data must be verified to be normal network activity.” (EX-1005 3:9-13.) Neither Petitioner nor Dr. Lee explain what type of training data would be necessary to train the hypothetical model that they would purportedly incorporate into Kester or how that training data would be labeled. Given Kester’s reliance on human decision to determine expected network activity/access and determine the parameters under which expected network activity/access may be permitted, a POSITA would

understand that the need for human decision would persist. A POSITA would also understand that in order to eliminate the involvement of a human entirely, a labor and cost extensive amount of continuous retraining of the model and training data would persist, if it is even possible to do so. Accordingly, a POSITA would not be motivated to implement Honig's teachings where the need for human determinations would persist.

176. Taking Petitioner's own example from Kester—Petitioner asserts that Kester teaches classifying an object as “not malware,” because Kester discloses categorizing an application “in the parent group titled ‘access/privacy’ and ‘operating system’ category.” (Petition 26 (citing EX-1004 [0098]).) But in that same example, Kester further explains that the same application could additionally be categorized as “spy ware.” (EX-1004 [0098].) Kester's system is capable of returning multiple results on multiple questions in order to manage employees' use of their workstations.

177. For example, as previously discussed, depending on the specific algorithm that is implemented, “a set of labeled training data” may be needed to train the model. (EX-1005 21:46-50.) Thus, even if a POSITA were to incorporate Honig's teachings, a human reviewer would still be required to determine a set of expected network activity and categorize the expected network access to label the data needed to train Honig's model. (*Id.*) Honig explains that one of the critical inefficiencies

(and cost) of adaptive model generation is the labeling of training data as normal, attack, clean, etc., as well as the need to “translate[] [the data] into a form suitable for training.” (EX-1005 2:14-16, 2:34-42, 2:67-3:13.) “In the case of anomaly detection, each instance of data must be verified to be normal network activity.” (EX-1005 3:9-13.) Accordingly, a POSITA would not be motivated to implement Honig’s teachings where the need for human determinations would persist.

178. Accordingly, the petition fails to show that Kester in view of Honig renders obvious “*automatically generating a mask.*”

D. Dependent Claims

179. Because Kester in view of Honig does not render obvious all limitations of the independent claims, I understand that Grounds 1 and 2 fail to render obvious dependent claims 2-12, 14-24, and 26-30 as well.

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

180. 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

181. 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).)

182. As demonstrated by my summary of the prior art, above, there was at least a long-felt and persistent need prior to the '250 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 and required extensive manual or semi-manual detailed analysis of objects and/or human administrator decisions. (*See* Section III, *supra*.) Where others failed, the '250 Patent succeeded. (*See* Section IV(A), *supra*.)

183. Thus, these secondary indicia only further support my opinion that the '250 Patent was non-obvious.

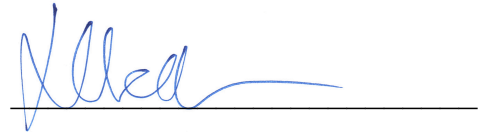
IX. CONCLUSION

184. 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 '250 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 19th day of December, 2023 in Los Angeles.

Respectfully submitted,



Nenad Medvidovic, Ph.D.