

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SOPHOS LTD. & SOPHOS INC.,

Petitioners,

v.

OPEN TEXT INC.,

Patent Owner.

Case IPR2023-00699

Patent No. 8,418,250

PATENT OWNER'S RESPONSE

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. THE ’250 PATENT	2
A. Inadequacy of the Prior Art & The Need for Improved Malware Detection.....	2
B. Innovations of the ’250 Patent.....	4
C. Claim Construction.....	8
1. “classifying” the computer object.....	8
2. “object”	10
3. The preamble of claim 25 is limiting.....	12
III. LEVEL OF ORDINARY SKILL IN THE ART	13
IV. OVERVIEW OF THE RELEVANT PRIOR ART	14
A. Overview of Halperin	14
B. Overview of Mankins	19
C. Overview of Kirby	21
V. PETITIONER FAILS TO SHOW A MOTIVATION TO COMBINE ITS PRIOR ART OR A REASONABLE EXPECTATION OF SUCCESS IN ACHIEVING THE INVENTION CLAIMED IN THE ’250 PATENT	22
A. Petitioner nowhere articulates its proposed combination of Halperin with Mankins.	24
B. There was no motivation to combine Halperin with Mankins.	31
C. There was no reasonable expectation of success in combining Halperin with Mankins.	38

D.	Adding Kirby further destabilizes Petitioner’s combination.....	41
VI.	PETITIONER FAILS TO SHOW THAT ITS COMBINATIONS RENDER OBVIOUS EVERY LIMITATION OF THE CHALLENGED CLAIMS ...	43
A.	[1C], [13C] Petitioner fails to show that Halperin discloses or renders obvious “ <i>classifying the computer object as malware</i> ”	45
1.	Halperin does not disclose or render obvious “ <i>classifying the computer object as malware.</i> ”	46
2.	Petitioner’s arguments that Halperin teaches classifying a “computer object” are flawed.	49
B.	[1D] Petitioner fails to show that Halperin in view of Mankins discloses or renders obvious “ <i>initially classifying the computer object as not malware.</i> ”	55
C.	[1I] Petitioner fails to show that Halperin in view of Mankins discloses or renders obvious “ <i>reclassifying the computer object as malware</i> ”	59
D.	[25P] Petitioner fails to show that Halperin in view of Mankins discloses or renders obvious “[<i>a method of providing data about a computer object from a remote computer to a base computer so that a comparison can be made at the base computer with similar data received from other remote computers, the method comprising.</i> ”	64
E.	[2], [14], [25B] Petitioner fails to show that Halperin in view of Mankins discloses or renders obvious “ <i>wherein the data about the computer object that is sent from the plural remote computers to the base computer includes one or more of: ... the current name of the object.</i> ”	65
VII.	CONCLUSION.....	68

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>ActiveVideo Networks, Inc. v. Verizon Comms., Inc.</i> , 694 F.3d 1312 (Fed. Cir. 2012)	26, 30, 31
<i>ADT LLC v. Vivent, Inc.</i> , IPR2022-00642, Paper 7 (PTAB Oct. 4, 2022)	31
<i>Catalina Mktg. Int'l, Inc. v. Coolsavings.com, Inc.</i> , 289 F.3d 801 (Fed. Cir. 2002)	12
<i>In re Cyclobenzaprine Hydrochloride Extended-Release Capsule Pat. Litig.</i> , 676 F.3d 1063 (Fed. Cir. 2012)	33
<i>DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.</i> , 567 F.3d 1314 (Fed. Cir. 2009)	38
<i>Echolochem, Inc. v. S. Cal. Edison Co.</i> , 227 F.3d 1361 (Fed. Cir. 2000)	40
<i>Elekta Ltd. v. ZAP Surgical Sys., Inc.</i> , 81 F.4th 1368 (Fed. Cir. 2023)	24, 40
<i>Harmonic Inc. v. Avid Tech., Inc.</i> , 815 F.3d 1356 (Fed. Cir. 2016)	25
<i>Intel Corp. v. PACT XPP Schweiz AG</i> , 61 F.4th 1373 (Fed. Cir. 2023)	32
<i>Intelligent Bio-Sys., Inc. v. Illumina Cambridge Ltd.</i> , 821 F.3d 1359 (Fed. Cir. 2016)	22, 23
<i>Juniper Networks, Inc. v Correct Transmission, LLC</i> , IPR2021-00682, Paper 26 (PTAB Oct. 3, 2022)	40, 43
<i>In re Keller</i> , 642 F.2d 413 (CCPA 1981)	25

<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	22, 30, 31
<i>Lowe’s Home Ctrs., LLC v. Epistar Corp.</i> , IPR2018-00932, Paper 7 (PTAB Nov. 1, 2018).....	24
<i>Lumenis Be Ltd v. BTL Healthcare Techs. A.S.</i> , IPR2021-01280, Paper 8 (PTAB Mar. 9, 2022)	54
<i>Lyft, Inc. v. Quartz Auto Techs., LLC</i> , IPR2020-01450, Paper 7 (PTAB Mar. 4, 2021)	25, 27, 30
<i>In re Magnum Oil Tools Int’l, Ltd.</i> , 829 F.3d 1364 (Fed. Cir. 2016)	24
<i>Netflix, Inc. v. DivX, LLC</i> , 84 F.4th 1371 (Fed. Cir. 2023)	57, 65
<i>New England Power Generators Ass’n, Inc. v. FERC</i> , 707 F.3d 364 (D.C. Cir. 2013).....	58
<i>In re Nuvasive, Inc.</i> , 842 F.3d 1376 (Fed. Cir. 2016)	31
<i>Palo Alto Networks, Inc. v. Centripetal Networks, Inc.</i> , IPR2021-01153, Paper 10 (PTAB Jan. 24, 2022)	31
<i>Pers. Web Techs., LLC v. Apple, Inc.</i> , 848 F.3d 987 (Fed. Cir. 2017)	25, 27, 28, 31
<i>Pfizer, Inc. v. Apotex, Inc.</i> , 480 F.3d 1348 (Fed. Cir. 2007)	41
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (en banc)	8, 10
<i>In re Rail Freight Fuel Surcharge Antitrust Litig.</i> , 725 F.3d 244 (D.C. Cir. 2013).....	58
<i>Raytheon Techs. Corp. v. Gen. Elec. Co.</i> , 993 F.3d 1374 (Fed. Cir. 2021)	49

<i>In re Robertson</i> , 169 F.3d 743 (Fed. Cir. 1999)	54
<i>Renishaw PLC v. Marposs Societa’</i>	10
158 F.3d 1243, 1249 (Fed. Cir. 1998)	
<i>Roland Corp. v. Inmusic Brands, Inc.</i> , IPR2018-00335, Paper 14 (PTAB July 2, 2018)	54
<i>SIMO Holdings Inc. v. Hong Kong uCloudlink Network Tech. Ltd.</i> , 983 F.3d 1367 (Fed. Cir. 2021)	13
<i>T-Mobile US, Inc. v. TracBeam LLC</i> , IPR2015-01687, Paper 10 (PTAB Feb. 12, 2016)	31
<i>Teva Pharms. USA, Inc. v. Corcept Therapeutics, Inc.</i> , 18 F.4th 1377 (Fed. Cir. 2021)	43
<i>TikTok Inc. v. Advanced Coding Techs. LLC</i> , IPR2022-01546, Paper 9 (PTAB June 15, 2023)	25
<i>TQ Delta, LLC v. CISCO Sys., Inc.</i> , 942 F.3d 1352 (Fed. Cir. 2019)	32
<i>TriVascular, Inc. v. Samuels</i> , 812 F.3d 1056 (Fed. Cir. 2016)	24
<i>WBIP, LLC v. Kohler Co.</i> , 829 F.3d 1317 (Fed. Cir. 2016)	33
<i>Xerox Corp. v. Bytemark, Inc.</i> , IPR2022-00624, Paper 9 (PTAB Aug. 24, 2022)	55
Statutes	
35 U.S.C. §§ 316(e), 318(a)	2
Other Authorities	
37 C.F.R. § 42.24	1
37 C.F.R. § 42.65(a)	55

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>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)
2004	<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)
2005	<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)
2006	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
2007	U.S. District Courts—Federal Court Management Statistics—Profiles—During the 12-Month Periods Ending March 31, 2018 Through 2023, “Table N/A—U.S. District Courts—Combined Civil and Criminal Federal Court Management Statistics (March 31, 2023)”, <i>available</i> https://www.uscourts.gov/sites/default/files/data_tables/fcms_na_distprofile0331.2023.pdf
2008	<i>Webroot, Inc., Open Text Inc., v. AO Kaspersky Lab et al.</i> , Case No. 6:22-cv-00243-ADA-DTG, Order Granting Third Amended Scheduling Order, Dkt. 142 (W.D. Tex., Dec. 27, 2022)

Exhibit	Description
2009	<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)
2010	<i>Webroot, Inc. and Open Text Inc., v. Sophos Ltd.</i> , Case No. 6:22-cv-240-ADA-DTG, First Amended Complaint For Patent Infringement, Dkt. 42 (W.D. Tex. June 23, 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. 17, 2022)
2012	<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)
2013	<i>Webroot, Inc. and Open Text Inc., v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.</i> , Case No. 6:22-cv-00241-ADA-DTG, Defendant's Preliminary Invalidity Contentions (W.D. Tex., Sept. 13, 2022) (excerpted)
2014	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab.</i> , Case No. 6:22-cv-00243-ADA-DTG, Defendant's Preliminary Invalidity Contentions Appendix A (W.D. Tex., Sept. 14, 2022) (excerpted)
2015	Declaration of Professor Nenad Medvidovic, Ph. D. in Support of Patent Owner's Response
2016	Transcript of the deposition of Dr. Richard Newman taken December 7, 2023 (excerpted)
2017	Marius Gheorghescu et al., An Automated Virus Classification System, VIRUS BULLETIN CONFERENCE 2005, 294-300 (Oct. 2005)

Open Text Inc. (“Patent Owner”) submits this response to the petition of Sophos Ltd. & Sophos, Inc. (collectively, “Petitioner”) seeking *inter partes* review (“IPR”) of the U.S. Patent No. 8,418,250.

I. INTRODUCTION

Petitioner fails to show by a preponderance of the evidence that the '250 patent is obvious.

The '250 patent uses a mask to classify computer objects—individual programs—as malware. But both of Petitioner's grounds rely on U.S. Patent Application Publication No. 2004/0111632 (“Halperin”) as a primary reference, which takes a fundamentally different approach to malware remediation from that claimed by the '250 patent. Halperin evaluates *aggregate computer and network-level behavior* to find symptoms that may indicate malware has infected a network. It does not analyze the operation of a single program or associate symptoms of malware infection with specific programs. Even when Halperin's system determines malware may be present, Halperin uses containment actions—such as suspending a computer from sending messages—and does not “*classif[y]*” any program as malware, as the '250 patent claims. In short, these different systems teach different approaches to solve different problems.

Petitioner also fails to identify a plausible motivation to combine Halperin's network-intrusion-response system with U.S. Patent No. 7,900,194 ("Mankins"), which examines potential malware not by looking for computer-level behavior on a network but by probing system calls within an individual computer's operating-system kernel. Simply put, there would have been no motivation to add kernel-level, program-specific filters to Halperin's computer- and network-level system. In fact, Halperin expressly disparages techniques that analyze individual software programs. Similarly lacking is Petitioner's proposed motivation to combine Halperin and Mankins with a third reference, U.S. Patent No. 7,694,150 ("Kirby"). Kirby teaches a heuristic behavioral method of responding to malware—a method Halperin rejects.

Petitioner, moreover, fails to explain how these references teach or render obvious limitations in the '250 patent's dependent claims. The Board should conclude that Petitioner has failed to show by a preponderance of the evidence that any claim of the '250 patent is unpatentable. See 35 U.S.C. §§ 316(e), 318(a).

II. THE '250 PATENT

A. Inadequacy of the prior art & the need for improved malware detection

"Malware" generally refers to malicious computer code, *e.g.*, viruses, trojans, worms, etc. (EX-2015 ¶ 28.)

Much prior art anti-malware software focused on locating and stopping *known* malware. (EX-1001 1:18-2:18.) For example, some anti-virus software analyzed objects for “signatures”—patterns within the object known to be indicative of malware. (EX-1001 1:18-28.) This approach has significant drawbacks: whenever new malware was detected, anti-malware service providers would be required to construct and distribute new signatures as updates to their anti-malware programs. (*Id.*) Other prior art systems compared objects against databases of known files to determine whether a file had been known for long enough that it could be deemed safe. (EX-1001 1:43-47.) These methods for detecting known malware suffered from inevitable extensive delay—either while waiting for malware to be detected and a signature generated or waiting long enough that a program could be deemed safe. (EX-1001 1:30-37, 1:48-54; EX-2015 ¶¶ 29-31.)

Other anti-malware systems also sought to protect against previously unknown malware. For example, some anti-malware systems that used a central computer and local computers on a network relied on analysis at the central computer to determine whether a file was known or unknown. (EX-1001 1:60-2:4.) But when a file was unknown, the system had to provide the entire file to the central computer for a detailed analysis. (EX-1001 2:4-11.) That analysis typically required human involvement and as a result could “take days.” (*Id.*) Other systems similarly relied

on a human administrator to “make a better informed though still ‘manual’ decision as to whether or not the file [was] safe to run.” (*E.g.*, EX-1001 2:18-27.) Thus, such prior art systems still resulted in considerable delay before a new file could be classified as safe or malware. (EX-1001 2:12-13; EX-2015 ¶¶ 29-31.) During such delays, malware could continue to spread, making the eventual cleanup more difficult.

Accordingly, there was a long-felt need for an improved technique to efficiently determine whether files were safe or malware. (EX-2015 ¶ 32.)

B. Innovations of the '250 Patent

The '250 patent is entitled “Methods and Apparatus for Dealing with Malware” and claims a method and system for classifying a computer object as malware. (*E.g.*, EX-1001 Claims 1, 13, 25.) Specifically, the patented technology introduced a method and system for “a rapid determination of” whether an object is safe or malware, without the need for time-consuming preliminary analysis of the object itself. (EX-1001 3:8-18.) The specification’s definition of “object” makes clear that “objects” according to the '250 patent consist of pieces of code to be operated by or on the computer. (*Id.* 7:63-66; EX-2015 ¶ 34.)

The '250 patent discloses a method in which “a base computer” receives data about a computer object from each of a plurality of “remote computers,” compares

the received data, and classifies the computer object as malware or not malware based on its comparison. (EX-1001 21:45-52.) This method can be used in the exemplary system illustrated below:

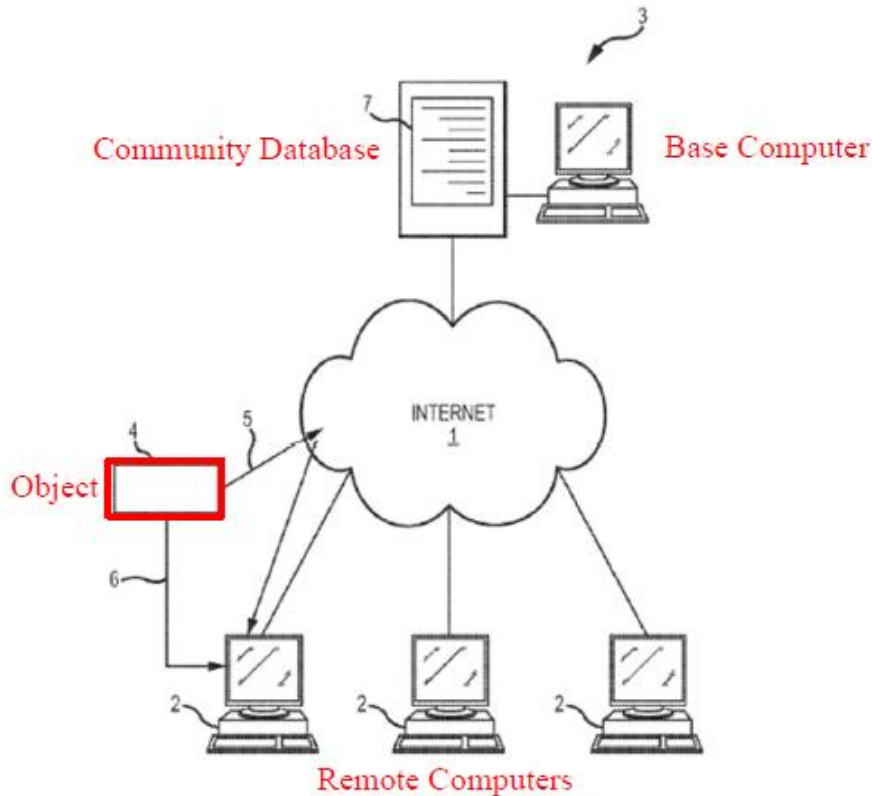


FIG.1

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

When an unknown object runs on a remote computer, data related to the object is passed to the base computer to be stored in the community database and for further analysis. (EX-1001 8:17-18, 8:44-47.) Data is preferably sent in the form of a “key” or “signature,” instead of the entire object, which allows reduced transmission and

storage requirements, quick comparison of objects, and rapid analysis at the base computer. (EX-1001 3:51-4:6.) The base computer searches the community database for the object's signature. (EX-1001 8:47-50, Fig. 2 (step 25).) If the signature is found, the object is marked "safe" or "unsafe" according to its designation in the database, and the system provides a signature, a message, or both to the remote computer, allowing that remote computer to perform a local comparison the next time the object is encountered. (EX-1001 8:60-9:7, Fig. 2 (steps 26-27).) In this case, the object may not be allowed to run, or the user may be given an informed choice whether to run the object. (EX-1001 9:7-10.)

When the object remains unknown after checking the database, the system creates a signature representative of the unknown object. (EX-1001 9:11-17.) This signature is then provided to the remote computer and a message is sent that might instruct the remote computer not to run the object or to inform the user of the risk and offer the choice of running it. (EX-1001 9:17-25, Fig. 2 (steps 30-31).)

A key differentiator from signature-based prior art is that the '250 patent further discloses automatically generating a mask to monitor the object's behavior to determine whether reclassification is needed. (EX-1001 4:21-30; EX-2015 ¶¶ 43-47.) The '250 patent claims automatically "generating a mask for said object that defines acceptable behaviour for the object." (EX-1001 4:21-23; *id.* cl. 1.) This

“mask is generated in accordance with normal behaviour of the object determined from” the data received about the object. (EX-1001 cl. 1.) The mask might identify, for example, interfaces that the object might need to access to communicate, or databases that the object may interrogate. (EX-1001 14:61-15:4.) The system then monitors operation of that object within the mask's parameters. (EX-1001 9:26-39, 14:55-61, Fig. 3 (steps 38-39); EX-2015 ¶¶ 43-47.)

If the behavior of the object remains within the parameters set by the mask, the object is allowed to run. (EX-1001 15:45-47.) However, if the object's behavior exceeds the parameters of the mask, the system flags this aberrant behavior and revokes the object's permission to run. (EX-1001 15:47-52.) By continuously monitoring a process each time it runs, the system can note any behavior that differs from previous behavior and update the mask or reclassify the object as unsafe. (EX-1001 15:64-16:3.)

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 systems decrease reliance on signature verification and negate the need for manual, time-consuming “deep analysis” of an object. (*Id.*; EX-2015 ¶ 47.)

C. Claim Construction

On March 16, 2023, in parallel district court litigation, the district court concluded that no terms of the '250 patent were indefinite and construed all terms as having their plain and ordinary meaning. (EX-2009 at 3-5, 8.) However, two claim terms require construction to resolve the parties' dispute based on the arguments presented in the petition: "classifying" and "object." Further, the preamble of claim 25 is limiting.

1. "classifying" the computer object

In its Order granting institution of this petition, the Board noted that "on [the institution-decision] record, the claimed 'classifying' [in Claim 1] does not preclude classifying by a human, as claim 1 contains no recitation of a device configured to perform the 'classifying.'" Paper 17 at 23 n.9. But the plain and ordinary meaning to a POSITA, considering the teachings of the specification, requires that the base computer performs "classifying." (EX-2015 ¶¶ 49-51.)

First, the specification, which is "the single best guide to the meaning of a disputed term," rejects human intervention as too slow. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed. Cir. 2005) (en banc). The '250 patent's specification highlights prior art in which "human involvement" may cause "days" of delay in "determin[ing] whether [an object] should be regarded as safe or malware." (EX-

1001 2:8–13.) Similarly, the specification points to a second piece of prior art in which a “human system administrator can ... make a better informed though still ‘manual’ decision as to whether or not the file is safe to run.” (EX-1001 2:23-27.) Such “delay,” the patent teaches, causes “risk to users during the period that the file is analysed and new anti-malware signatures distributed.” (EX-1001 2:59-64.) Those delays are a problem the system claimed in the ’250 patent was invented to solve. (EX-2015 ¶¶ 49-51.)

Accordingly, the patent teaches that the “classifying” step is the result of “comparing *in the base computer* the data about the computer object” under consideration. (EX-1001 3:4–5; EX-2015 ¶¶ 49-51.) Unlike with human intervention, this automatic classification “allows a rapid determination of the nature of the object to be made.” (EX-1001 3:13-14; EX-2015 ¶¶ 49-51.)

Second, Petitioner has waived any argument to the contrary. Petitioner states that claim 1 is “materially identical” to claim 13 “except that claim 13 is an apparatus version.” (Petition 42, 50.) Claim 13 recites a “*base computer being constructed and arranged to classify* the computer object as malware when the base computer determines that the data indicates that the computer object is malware.” (EX-1001 21:49-52 (emphasis added)); see *Phillips*, 415 F.3d at 1314 (use of the same term in different claims is a “valuable source[] of enlightenment”). Petitioner cannot

belatedly contend that the meaning of “classify” in claim 1 is broader than it is in claim 13. In any event, Petitioner cannot escape the fact that claim 1 expressly provides that the “determining” step—in which data about the object is analyzed to see whether it “indicates that the computer object is malware”—is performed in the base computer. (EX-1001 at 20:29-30.)

Accordingly, under the plain and ordinary meaning of “classifying” to a POSITA, “classifying” is performed by the base computer.

2. “object”

When a patentee has elected to be its own lexicographer “by providing an explicit definition in the specification for a claim term,” its selected definition controls. *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1249 (Fed. Cir. 1998).

In the '250 patent, the patentee has explicitly defined the meaning of “object.” (EX-2015 ¶¶ 52-55.) The specification explains that a computer object may be one of five things: (1) a computer file; (2) a part of a file or a sub-program; (3) a macro; (4) a web page; or (5) any other piece of code. (EX-1001 7:63-66.) Further limiting these five categories is that each of them is “to be operated by or on the computer, or any other event whether executed, emulated, simulated or interpreted.” *Id.* That

is, the words “computer” and “any other event” operate as grammatical objects of the prepositions “by or on.”

The specification separately defines the word “event” at columns 12-13. An event

“define[s] the actions or behaviours of an object acting upon another object or some other entity. The event has three principal components: the key of the object performing the act (the ‘Actor’), the act being performed (the ‘Event Type’), and the key of the object or identity of another entity upon which the act is being performed (the ‘Victim’).”

(EX-1001 12:64-13:2.)

Taking into account both of these definitions, a computer object cannot be an event within the context of the ’250 patent because it would be circular to define a computer object as an event when an event is defined based on the objects on which it acts. (EX-2015 ¶¶ 52-55.) Instead, the specification’s definition of “event” reinforces the patent’s definition of “object,” *i.e.*, that objects are “operated by or on ... any other event.” (EX-1001 7:63-66; EX-2015 ¶¶ 52-55.) The “Actor” and “Victim” objects are the computer objects being operated by or on the event.

The petition contains no argument that “object” as used in the ’250 patent should include “events.” However, certain positions of Petitioner’s expert, Dr. Newman, are based on twisting the definition of “object” to include “events.” (EX-

2016 66:2-67:1.) To the extent the Petitioner implicitly relies on those unpreserved positions—as further explained below—Petitioner's arguments should be rejected.

Indeed, further confirming that an object according to the '250 patent should not include events, Dr. Newman himself testified that his baseline understanding of “object” accords with the above definition. For example, he stated that his “preference would be to refer to objects as only things like files or pieces of files” and that he doesn't “know that [he] would say ‘event’ in [the definition] personally.” (EX-2016 49:15-17, 112:18-22.) In sum, all relevant evidence points to the exclusion of “events” from the plain and ordinary meaning of “object” within the '250 specification.

3. The preamble of claim 25 is limiting

A claim's preamble may be limiting “if it recites essential structure or steps, or if it is ‘necessary to give life, meaning, and vitality’ to the claim.” *Catalina Mktg. Int'l, Inc. v. Coolsavings.com, Inc.*, 289 F.3d 801, 808 (Fed. Cir. 2002). The preamble of claim 25 of the '250 patent is limiting because it recites an essential step—“a comparison ... made at the base computer with similar data received from other remote computers.” (EX-1001 cl. 25.) The comparison step is crucial to the limitations of the claim that involve determining “normal behaviour of the object determined from said received data.” (EX-1001 cl. 25, 3:8-41 (“this aspect allows a

comparison to be made between the objects and/or their effects on the different remote computers to determine whether or not a particular object should be classed as good or as malware.”.) But the comparison step is not included anywhere else in the body of the claim. The only comparison mentioned in the body of claim 25 is between the mask and the object’s behavior, not between data received from multiple computers. (EX-1001 23:19-22.) The omission from the body of the claim of a complete description of the invention is a key reason to treat a preamble as limiting. *SIMO Holdings Inc. v. Hong Kong uCloudlink Network Tech. Ltd.*, 983 F.3d 1367, 1375 (Fed. Cir. 2021) (finding a preamble limiting in part because the body of the claim did not identify the structure of the invention).

III. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art 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 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. (EX-2015 ¶¶ 57-60.) More

advanced degrees or training in a related discipline or both can compensate for shorter work experience. (*Id.*)

IV. OVERVIEW OF THE RELEVANT PRIOR ART

A. Overview of Halperin

Halperin (U.S. Patent Application 2004/0111632 A1) was published June 10, 2004. (EX-1006 at 1.) Halperin is directed to detecting patterns of *collective computer behavior on a network* that suggest the presence of malware activity, and—when it detects such symptoms—taking containment actions to halt further spread. (EX-1006 Abstract, [0036-0057].)

Halperin observed that some prior art antivirus software identified virus behavior by characterizing a program's legitimate behavior and identifying any deviations from that behavior. (EX-1006 [0006].) Halperin criticized this type of object-specific technique as often resulting in false alarms due to the fact that computer user behavior tends to vary over time and between users. (EX-1006 [0006]; EX-2015 ¶¶ 62-69.) Halperin contended that due to this technique's shortcomings, users frequently either disabled this type of antivirus software entirely or set its sensitivity so low that it failed to find existing viruses. (EX-1006 [0006].)

Halperin further recognized that “[i]n most cases, viruses are spread when computer users send infected files to other computer users,” for example, via e-mail,

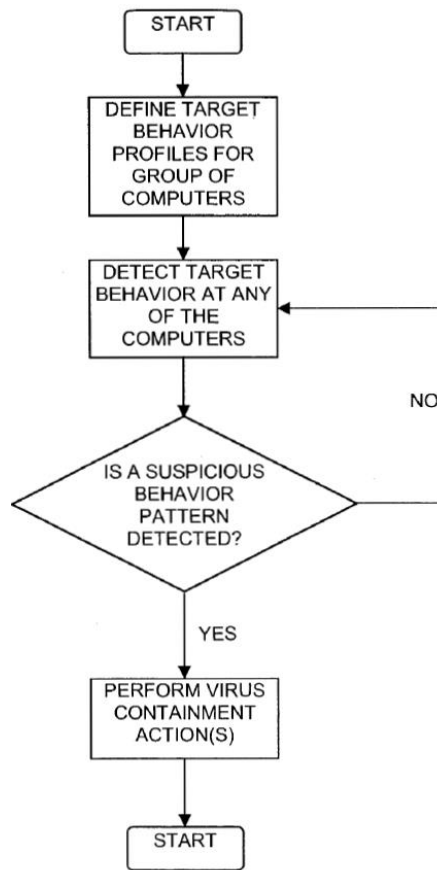
data-storage media (*e.g.*, disk or CD), or by sharing infected files from one computer to another. (EX-1006 [0003].) Some malware—known as a worm—was even capable of spreading on its own “with little or no intervention on the part of the computer user.” (EX-1006 [0004].) For example:

A virus that spreads via e-mail messages will typically access an e-mail program's address book or sent/received mail folders and automatically send itself to one or more of these addresses. Alternatively, the virus may attach itself to otherwise innocuous e-mail messages that are sent by a computer user to unsuspecting recipients.

(EX-1006 [0004].)

Thus, Halperin sought to provide a *network-level detection and containment method*, while overcoming the “false alarm” problem it perceived in the prior art. (EX-1006 [0037]; EX-2015 ¶¶ 62-70.) Instead of monitoring or analyzing a particular computer object—like the system in the '250 patent—Halperin's proposed solution detects suspicious patterns of “target behavior” across a group of computers on a network (indicating the potential presence of malware in the network) and takes one or more of what Halperin calls “virus containment” actions. (EX-1006 Abstract, [0038]-[0057]; EX-2015 ¶¶ 62-69.)

Figure 6 illustrates this overall process:



(EX-1006, Fig. 6.)

Halperin starts by defining the kinds of behavior it will target using one or more “target behavior profiles” for a group of computers on a network. (EX-1006 [0038], Fig. 6.) Examples of target behavior profiles include, among others: “[s]ending messages to more than a predefined number of users during a predefined period of time;...[m]odifying operating system file such as the Microsoft Windows registry; [and] [d]eleting more than a predefined number of files on the computer’s hard disk during a predefined period of time....” (EX-1006 [0039]-[0046].)

Whereas the '250 patent looks at whether the same or similar *objects* are behaving in a certain fashion across multiple computers on a network, Halperin looks at examples of the same *computer-level behavior*. (EX-1006 [0037]-[0038]; EX-2015 ¶ 70.)

Halperin describes using “detectors (e.g. file deletion and Windows Registry access) and decoys (e.g. email decoy)” which it sometimes refers to as “sensors.” (EX-1006 at [0086].) Those sensors operate in a server on the network to correlate the presence of target computer behavior on multiple computers, *e.g.*, by “identifying the same *activity* in different computers at about the same time.” (EX-1006 [0047].) Halperin then teaches determining whether this collective computer-level “target behavior” corresponds to a “predefined suspicious behavior pattern,” which would indicate that a virus may have infected the network. (EX-1006 [0047].)

The exemplary patterns of suspicious behavior Halperin discloses identify a “certain percentage of the computers in the network” all doing the same thing, most often within a relatively short amount of time. (EX-1006 [0048]-[0056]; EX-2015 ¶ 68.) For example, Halperin states that when a “certain percentage of computers in the network send[] more than 10 messages per minute in the last 5 minutes,” malware may be present on the network. (EX-1006 [0048].) Whereas the '250 patent relies on data that is specific to a particular computer object to determine whether that

object is malware (Section II.B), Halperin relies on data that is not tied to a particular object. (EX-2015 ¶ 68.) Instead, it seeks to determine whether a certain percentage of computers on the network are behaving unexpectedly. The system neither identifies nor requires identifying any particular computer object that may be causing the target behavior by the computers on the network. (EX-2015 ¶ 68.)

Halperin teaches that if the server detects a suspicious behavior pattern, it may initiate “virus containment actions.” (EX-1006 [0057], [0023]-[0030].) Because the system teaches no means of identifying a particular piece of software as the malware culprit or determining whether any software is or is not malware, Halperin's proposed solutions amount to either telling a human that there might be a problem (*id.* [0025]-[0027]; EX-2015 ¶ 70.) or halting the flow of information across the network (*id.* [0024], [0028]-[0030]; EX-2015 ¶ 70.) Halperin does not teach removing the offending software. (See EX-1006 [0024]-[0030]; EX-2015 ¶ 70.)

Thus, whereas the '250 patent examines information about the same or similar *computer objects* across multiple computers on a network to determine whether the computer *object* should be classified as malware, Halperin looks for the same *behavior or activity* across multiple computers on a network to determine whether that activity is symptomatic of a malware infection. (EX-1006 [0037]-[0038]; EX-2015 ¶ 70.)

B. Overview of Mankins

Mankins (U.S. Patent No. 7,900,194) issued March 1, 2011, from an application filed March 23, 2005. (EX-1008 at 1.) Mankins is directed to detecting intrusion in a computer system by monitoring system calls within the computer's operating system (OS) kernel. (EX-1008 Abstract.) Mankins defines the OS kernel as the portion of the OS "configured to interface computer programs ... with shared resources." (EX-1008 4:11-13.) Mankins teaches using a filter called a "Bloom filter" to determine whether a given system call falls outside or likely falls within an acceptable set of system calls for a given program.

Bloom filters are computer-science constructs that use hash functions to compress the values in a given set into a bit vector. (EX-2015 ¶ 72.) Mankins treats Bloom filters as within a POSITA's knowledge. (See EX-1008 7:34-41, 8:19-25.) Once the Bloom filter bit vector has been constructed, a computer can quickly and easily determine whether a value is likely a member of the set or definitely not a member of the set. (EX-2015 ¶ 72.) The computer hashes the value using the same hash function used to create the bit vector and compares the output to the bit vector. (*Id.*) The result is either positive (which means that the value is *likely* part of the set) or negative (which means that the value is *definitely not* part of the set). (*Id.*) Thus,

a Bloom filter comes with an inherent risk of incorrect identifications—values that are not within the set but that nonetheless return a match to the filter. (*Id.*)

Mankins describes a “Bloom filter data object” as “a set of hash functions and a bit vector ... of a given length.” (EX-1008 8:19-22.) Mankins teaches “encod[ing] each unique system-call sequence associated with expected operation of a computer program ... into a particular Bloom filter data object.” (EX-1008 7:44-46.) Doing so allows the system to “compactly store data representative of predefined normal program behavior.” (EX-1008 7:51-52.)

Mankins also notes that Bloom filters generate errors but do so at a known rate correlated to the bit vector's length. (EX-1008 8:64-67.) Mankins teaches that because Bloom filters “throw[] away data to represent the data in a compressed format,” achieving “lower error rates” requires a larger “bit vector.” (EX-1008 8:44-45, 54-58.) Accordingly, Mankins suggests that a “small size[]” for a Bloom filter data object according to the invention is “approximately two kilobytes ... or three kilobytes” but that the system according to the invention “may be configured to generate a Bloom filter data object of any suitable size.” (EX-1008 8:49-52.)

Mankins teaches monitoring a computer program's system calls within the OS kernel to define the program's expected behavior—including, optionally, monitoring system calls while running the program under training conditions. (EX-

1008 12:51-54; *see id.* 11:19-30.) Mankins then teaches encoding data representative of the program's system-call behavior using hash functions to generate a Bloom filter data object. (EX-1008 12:55-61.) That object is appended to the program, and when the program runs, the data object is embedded in the OS kernel. (EX-1008 12:52-65.) The system according to Mankins then compares the system calls the program makes during operation with the data in the Bloom filter data object. In one embodiment, the comparison may be used to detect anomalous behavior. (EX-1008 10:66-67.) A reporting subsystem may then "automatically invoke predetermined actions to protect the computer or to analyze the nature and/or source of the detected intrusion." (EX-1008 11:8-11.)

C. Overview of Kirby

Kirby (U.S. Patent No. 7,694,150 B1) issued April 6, 2010, from an application filed June 22, 2004. (EX-1007 at 1.) Kirby is directed to an "integrated approach" to antivirus protection that "combines the behavioral remedy against unknown transmissions with the signature matching of known harmful transmission." (EX-1007 Abstract.) Kirby teaches a predefined "behavioral heuristic indicative of actions known to be harmful or undesirable." (EX-1007 at 7:48-50.) Kirby's behavior-monitoring approach determines whether actions taken

by a piece of software match that heuristic and are thus presumptively undesirable.
(EX-1007 at 10:54-60.)

In addition to that behavior-monitoring heuristic approach, Kirby teaches a separate “parallel” signature-matching approach. (EX-1007 at 9:25.) Kirby describes signature matching as a “[c]onventional virus detection mechanism[.]” (EX-1007 at 7:30-40.) In such an approach, Kirby teaches performing a “content based check” on a file against a “cache” of “malware objects previously discovered.” (EX-1007 at 9:27-40.)

V. PETITIONER FAILS TO SHOW A MOTIVATION TO COMBINE ITS PRIOR ART OR A REASONABLE EXPECTATION OF SUCCESS IN ACHIEVING THE INVENTION CLAIMED IN THE '250 PATENT

To combine Halperin with Mankins, Petitioner must show two distinct requirements: a motivation to combine the proffered prior art in a manner that teaches or renders obvious the claimed invention, and a reasonable expectation of success in achieving the claimed invention. *E.g.*, *Intelligent Bio-Sys., Inc. v. Illumina Cambridge Ltd.*, 821 F.3d 1359, 1367 (Fed. Cir. 2016). To show motivation to combine, Petitioner must demonstrate “a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007). And to show a reasonable expectation of success, Petitioner must

demonstrate that a POSITA would be likely to succeed in combining the references to meet the limitations of the claimed invention. *Intelligent Bio-Sys.*, 821 F.3d at 1367.

Petitioner fails to explain why a POSITA would have combined Mankins's system with Halperin's, other than counsel's hindsight desire to use bits of one reference to plug holes in the other following the '250 patent's blueprint. Petitioner's failure to identify a motivation to combine is no surprise because Halperin teaches away from systems like the one disclosed in Mankins. Further, Petitioner fails to show that a relevant artisan would have expected success in making the invention claimed in the '250 patent from these mismatched references.

And, here, Petitioner fails to meet an even more fundamental requirement—it fails to explain what the proposed combination *is*. Petitioner does not adequately explain the combination of Halperin's network-level symptom-detection-and-containment system with Mankins's intra-kernel Bloom filter analysis of an individual piece of software's individual system calls. Each of these deficiencies alone is a sufficient basis for the Board to conclude that Petitioner has not met its burden to prove that the challenged claims of the '250 patent are unpatentable over its first combination.

Nor does Petitioner's attempt to shoehorn its third reference—Kirby—into its Halperin-Mankins combination solve any of those problems. Petitioner's description of the Halperin-Mankins-Kirby combination is even more perfunctory and conclusory than that of its Halperin-Mankins combination, and as with Mankins, Halperin expressly teaches away from heuristic systems like the one Kirby teaches. As a result, the petition fails to show that the '250 patent is unpatentable.

A. Petitioner nowhere articulates its proposed combination of Halperin with Mankins.

As a threshold matter, “obviousness requires ... a finding that a skilled artisan would have been motivated to combine the teachings of prior art”—and not just to compile a list of concepts, but to combine the teachings “in such a way that the combination discloses the claimed limitations.” *Elektro Ltd. v. ZAP Surgical Sys., Inc.*, 81 F.4th 1368, 1374 (Fed. Cir. 2023). Accordingly, “[i]t is Petitioner's burden to explain specifically how the references would be combined to produce the claimed invention.” *Lowe's Home Ctrs., LLC v. Epistar Corp.*, IPR2018-00932, Paper 7 at 19 (PTAB Nov. 1, 2018) (citing *TriVascular, Inc. v. Samuels*, 812 F.3d 1056, 1066 (Fed. Cir. 2016); *In re Magnum Oil Tools Int'l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016)). The petition lacks any such explanation.

Instead, the petition reads like exactly what it is: an attempt to map disparate concepts from the prior art onto the specific invention claimed in the '250 patent.

Petitioner never explains how a POSITA would have combined those concepts into a cohesive system, much less why anyone would try to make them into one. Instead of making those required showings, Petitioner recites what Halperin and Mankins teach *alone*. It fails to assert “what the combined teachings of those references would have suggested” to a POSITA. *In re Keller*, 642 F.2d 413, 425 (CCPA 1981) (restating the test for Section 103 obviousness). The Board has rejected “[t]his ‘pick-and-choose’ presentation of evidence,” which “is challenging to follow and makes it difficult to piece together in order to understand Petitioner’s proposed combination or modification.” *Lyft, Inc. v. Quartz Auto Techs., LLC*, IPR2020-01450, Paper 7 at 17-20 (PTAB Mar. 4, 2021).

“It [i]s Petitioner’s ‘burden to explain to the Board how’ the combination of references ‘rendered the challenged claims unpatentable.’” *TikTok Inc. v. Advanced Coding Techs. LLC*, IPR2022-01546, Paper 9 at 7 (PTAB June 15, 2023) (citing *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016). While the features of a secondary reference need not be bodily incorporated into the structure of the primary reference, “[a] clear, evidence-supported account ***of the contemplated workings of the combination is a prerequisite*** to adequately explaining and supporting a conclusion that a relevant skilled artisan would have been motivated to make the combination and reasonably expect success in doing so.” *Pers. Web*

Techs., LLC v. Apple, Inc., 848 F.3d 987, 994 (Fed. Cir. 2017) (overturning a panel's decision where "the Board nowhere clearly explained, or cited evidence showing, how the combination of the two references was supposed to work").

Petitioner's defective approach here does not and cannot carry that burden. See *ActiveVideo Networks, Inc. v. Verizon Comms., Inc.*, 694 F.3d 1312, 1327 (Fed. Cir. 2012) (affirming summary judgment of no invalidity where the defendant's expert "failed to explain how specific references could be combined, which combination(s) of elements in specific references would yield a predictable result, or how any specific combination would operate or read on the asserted claims").

First, Petitioner suggests using Mankins as a "sensor" in Halperin. (Petition 21-24.) But as explained below, this explanation does not disclose an actual combination of the prior art.

Petitioner suggests two different versions of using Mankins as a "sensor" in Halperin—one that incorporates a Bloom filter at the base computer and another that incorporates a Bloom filter at a remote computer. (Petition 22-23.) Petitioner offers no reason for selecting one version over the other, nor does it explain—beyond the most general terms—how either version would work. That approach—placing the burden on the Board and Patent Owner to discern the "workings of the combination"

and to choose for themselves among multiple avenues—is insufficient. *See Pers. Web Techs.*, 848 F.3d at 994; *Lyft*, IPR2020-01450, Paper 7 at 17-20.

Petitioner also does not explain how it would integrate Mankins's intra-kernel, program-specific approach as a “sensor” in the program-agnostic network-monitoring system taught by Halperin. This is no minor oversight because integrating Mankins wholesale into a Halperin system would be extraordinarily resource intensive. (EX-2015 ¶ 85.) Nothing in Halperin ties malicious activity it detects to any particular program. (EX-2015 ¶ 86.) In such an environment, a Mankins add-on would need to generate and run a Bloom filter for *every* piece of software running on *every* remote computer, and then report the results back to the Halperin central computer. (EX-2015 ¶ 86.) Though Petitioner conclusorily asserts that this would be a cost-free endeavor (Petition 24), it would in fact be anything but. (EX-2015 ¶ 86.)

Further undermining Petitioner's contention that Mankins would operate as a “sensor” in Halperin is Petitioner's assertion that Mankins's “process would begin” only “if the system [according to Halperin] does not identify malware based on the behavioral data received from remote ‘computers 500.’” (Petition 23.) Petitioner does not explain why, if Mankins were incorporated as a Halperin sensor, it would run separately from (and after) Halperin's other sensors, or whether Halperin's other

sensors would continue or stop running when Mankins's process begins to run. (See EX-2015 ¶ 84.) Nor does Petitioner explain at what point the Halperin sensors, which run continuously, would be deemed to have “not identif[ied] malware.” (EX-2015 ¶ 84.) In short, this is far from the requisite “clear, evidence-supported account.” It is, instead, a counsel-designed, hindsight-motivated attempt to stitch together the disparate parts of two ill-matched references using the '250 patent's claims as a blueprint.

Second, Petitioner's analysis of the claims merely identifies individual teachings from each of Mankins and Halperin—and still does not provide an “account of the contemplated workings ***of the combination***.” *Pers. Web Techs.*, 848 F.3d at 994 (emphasis added). For example, below is Petitioner's discussion of the purported combination in each limitation of claim 1:

Claim 1 limitation	Petitioner's asserted reference
[1P]	“Halperin discloses or renders obvious [1P].” (Petition 24.)
[1A]	“This limitation includes multiple parts, each of which is disclosed or rendered obvious by Halperin.” (Petition 25.)
[1B]	“Halperin discloses or renders obvious [1B].” (Petition 27.)
[1C]	“Halperin discloses or renders obvious [1C].” (Petition 28.)

Claim 1 limitation	Petitioner's asserted reference
[1D]	"Both Halperin and Mankins disclose or render obvious [1D]." (Petition 29.) The petition then explains how Petitioner contends that each reference <i>separately</i> discloses this limitation. (Petition 29-30.)
[1E]	"Mankins discloses or renders obvious [1E]." (Petition 30.)
[1F]	"Mankins discloses or renders obvious [1F]." (Petition 35.)
[1G]	"Mankins discloses or renders obvious [1G]." (Petition 38.)
[1H]	"Halperin and Mankins disclose or render obvious [1H]. ... [I]t would have been obvious to a POSITA to augment Halperin's disclosure of 'virus containment actions' with Mankins's disclosure of 'predetermined actions designed to protect the computer" (Petition 40-41.)
[1I]	"Halperin and Mankins disclose or render obvious [1I]. ... A POSITA would have understood ... Halperin's disclosure of 'virus containment actions' and Mankins's disclosure of 'predetermined actions designed to protect the computer' to include reclassifying the computer object as malware." (Petition 41-42.)

In short, of the nine limitations in claim 1, in only one of them ([1H]) does Petitioner even *suggest* there might be some role for combining Mankins's teachings with Halperin's, though again Petitioner is silent about how a person of ordinary skill in the art would go about the suggested "augment[ation]." And that "augment[ation]"—unrelated to *detecting* malware but to the actions the system takes when it suspects malware is present—has nothing to do with use of Mankins as a "sensor" for Halperin on which the Petitioner relies in its motivation to combine section. (Petition 21-24.) What's more, in the very next limitation, Petitioner suggests again that the same teachings of Halperin and Mankins should apply *separately*. (See Petition 41-42.)

In each of the remaining limitations, Petitioner simply points at the two references' separate purported teachings. The result is a mix-and-match petition in which Petitioner leaves it up to the Board (and Patent Owner) to interpret what the separate teachings might suggest to a POSITA if combined. *ActiveVideo*, 694 F.3d at 1327; *KSR*, 550 U.S. at 418 ("A patent composed of several elements is not proved obvious by merely demonstrating that each of its elements was, independently, known in the prior art."); *Lyft*, IPR2020-01450, Paper 7 at 17-20 (rejecting "'pick-and-choose' presentation of evidence"); *Palo Alto Networks, Inc. v. Centripetal Networks, Inc.*, IPR2021-01153, Paper 10 at 16-17 (PTAB Jan. 24, 2022).

In presenting this disjointed and amorphous combination, Petitioner has hobbled Patent Owner's ability to address the combination of disclosures from the petition. *T-Mobile US, Inc. v. TracBeam LLC*, IPR2015-01687, Paper 10 at 20 (PTAB Feb. 12, 2016) (“[w]e will not...improperly shift the burden of deciphering the arguments onto the Patent Owner”). Because Petitioner does not describe “***the contemplated workings of the combination***,” Petitioner's Halperin-Mankins ground cannot succeed. *Pers. Web Techs.*, 848 F.3d at 994; *ActiveVideo*, 694 F.3d at 1327; *ADT LLC v. Vivent, Inc.*, IPR2022-00642, Paper 7 at 13-14 (PTAB Oct. 4, 2022).

B. There was no motivation to combine Halperin with Mankins.

Even if Petitioner had sufficiently described a combination of Halperin and Mankins (and it has not, *see supra* Section V.A), Petitioner has failed to make the necessary showing of “a reason ... to combine the elements *in the way the claimed new invention does*.” *KSR*, 550 U.S. at 418 (emphasis added).

First, Petitioner has identified no problem in Halperin that would motivate a skilled artisan to seek out the inclusion of Mankins. *See, e.g., In re Nuvasive, Inc.*, 842 F.3d 1376, 1382, 1384 (Fed. Cir. 2016) (overturning a finding of obviousness because the PTAB failed to “articulate a reason why a PHOSITA would combine the prior art references”).

To be sure, a problem generally known in the art *can* be sufficient motivation to suggest to a POSITA to combine an additional reference. For example, in *Intel Corp. v. PACT XPP Schweiz AG*, 61 F.4th 1373, 1381 (Fed. Cir. 2023), the Federal Circuit held that a combination was motivated because “there was a known problem of cache coherency in the art, that Bauman’s secondary cache helped address that issue, and that combining the teachings of Kabemoto and Bauman wasn’t beyond the skill of an ordinary artisan.”

But Petitioner has identified no general problem that would be relevant to a motivation to combine Halperin with Mankins. Petitioner merely contends that “additional steps from Mankins would optimize the system’s ability to identify unknown malware.” (Petition 24.) That conclusory assertion of unspecified improvement is insufficient on its face. *See TQ Delta, LLC v. CISCO Sys., Inc*, 942 F.3d 1352, 1359, 1361 (Fed. Cir. 2019) (reversing PTAB decision of obviousness because arguments were “[u]ntethered to any supporting evidence”). Moreover, it is not a reason to select a *specific* reference such as Mankins. Petitioner’s cursory reasoning would support adding *any* other malware-detection solution to Halperin to achieve the same effect. Petitioner has not identified a reason why a POSITA would have selected Mankins out of the tremendous number of other malware-detection solutions, nor why Mankins would be particularly apt for this combination.

WBIP, LLC v. Kohler Co., 829 F.3d 1317, 1337 (Fed. Cir. 2016) (“Whether a skilled artisan would be motivated to make a combination includes whether he would select particular references in order to combine their elements”).

Petitioner's failure to identify why its proposed combination is unique is a fatal flaw. *In re Cyclobenzaprine Hydrochloride Extended-Release Capsule Pat. Litig.*, 676 F.3d 1063, 1070-71 (Fed. Cir. 2012) (“Where, however, a defendant urges an obviousness finding by merely throw[ing] metaphorical darts at a board in hopes of arriving at a successful result, but the prior art gave either no indication of which parameters were critical or no direction as to which of many possible choices is likely to be successful, courts should reject hindsight claims of obviousness.” (cleaned up))

Elsewhere, Petitioner contends that “a POSITA would have recognized that using Mankins's mask approach in a distributed system such as Halperin's would improve the system by reducing false positives.”¹ (Petition 34.) Besides the fact that this (again) is not a reason to select Mankins *in particular*, that proposed

¹ Dr. Medvidovic explains that a false positive is an undesirable result that occurs in malware detection when the system indicates the presence of malicious software when no such software is present. (EX-2015 ¶ 93.)

motivation is irreconcilable with the combination Petitioner actually proposes. According to Petitioner's argument, Mankins's process begins operating only after Halperin "*does not identify malware* based on the behavioral data received." (Petition 23 (emphasis added).) In a situation in which Halperin does not identify malware, it has not made a positive identification of malware. Accordingly, it cannot have made a *false* positive identification of malware. Mankins cannot help with improving false positives if it runs only when Halperin has not yielded any result at all. (EX-2015 ¶¶ 93.) Petitioner's alleged motivation to combine the references is objectively unattainable in the manner in proposes combining the references.

Second, Petitioner's motivation arguments fall apart on even cursory examination. Petitioner contends that "incorporating Mankins's Bloom filter data object into Halperin's malicious activity detection approach would have been nothing more than a predictable use of a prior art element according to its established function." (Petition 21.) But the use of Mankins that Petitioner proposes is neither predictable nor in line with Mankins's established function.

Petitioner suggests that a relevant artisan "would have recognized that the examples of 'target behavior' described in Halperin ... would be detected by monitoring system calls." (*Id.*) That argument does not consider how Halperin and Mankins actually work. Halperin's sensors collect reports of activity that, on an

individual basis, may not raise suspicions, but that *in the aggregate when performed at or near the same time* suggest malware activity. (EX-1006 at [0047]-[0056]; EX-2015 ¶¶ 89-90.)

Mankins's Bloom filter, meanwhile, is not designed to report on *non-anomalous* individual system calls. A *predictable* use of Mankins's method would be to do what it teaches—to use its Bloom filter to identify and report only the occurrence of “anomalous” individual system calls, which are particular system calls that fall outside the expected behavior of an individual program. (EX-1008 5:62-65; EX-2015 ¶¶ 94-95.) That technique—designed to pinpoint an individual action that is indicative of malware—is a poor fit for Halperin's detection system, wherein any *individual* action is ignored unless “[a] certain percentage of the computers” report a similar action at or near the same time. (EX-1006 [0048]-[0056]; EX-2015 ¶¶ 94-95.) Petitioner does not explain how a skilled artisan would overcome this mismatch in approaches—whether, for example, its proposed combination would require a number of computers to report actions exceeding Bloom filters in accordance with Halperin's teaching or would instead treat Mankins's system as something different from a standard Halperin sensor. Petitioner accordingly has not explained why that artisan would turn to Mankins when modifying a Halperin system.

Petitioner also contends that it would be “an obvious modification to collect system call sequences as part of the data” Halperin receives. (Petition 22.) But that contention contradicts Petitioner’s explanation of how the prior art’s teachings map onto the ’250 patent’s limitations. Petitioner contends that *Halperin alone* teaches “receiving data”; it does not rely on Mankins in combination with Halperin for that limitation. (Petition 25.)

Nor does Petitioner’s contention that Mankins “may be implemented in a distributed fashion” move the needle. (See Petition 22.) Merely pointing out that a Mankins system *is capable of* operating in the same kind of environment as a Halperin system is not motivation to graft a Mankins system onto a Halperin system. Pointing out that a fork will fit into a light socket, even if literally true, ought not to be misconstrued as motivation to introduce the one into the other.

In short, Petitioner does not explain why a POSITA would have sought to combine these particular references. Indeed, there is every reason to believe that a skilled artisan modifying a Halperin system (for a reason Petitioner does not specify) would *reject* Mankins. Petitioner suggests that Mankins would run as some sort of backstop for Halperin (“if the [Halperin] system does not identify malware based on the behavioral data received ... Mankins’s process would begin”) (Petition 23), such that Mankins would presumably capture malware Halperin missed. But a known

weakness of Mankins's Bloom filters is that they *produce false negatives*—misidentifying malicious activity as benign. (EX-1008 8:61-64; EX-2015 ¶ 97.) A POSITA would be disinclined to layer *two* programs that *both* tend to err in favor of giving a pass to malicious behavior. (EX-2015 ¶ 97.)

And ***third, Halperin teaches away from Mankins.*** Halperin explains that some prior art systems “employ[] heuristic techniques”—they analyze software behavior, “characteriz[e] legitimate software behavior and then identify[] any deviation from such behavior.” (EX-1006 [0006]; EX-2015 ¶ 98.) Halperin teaches that those techniques are not desirable because they “often result[] in a false alarm ... leading computer users to disable such software or set the sensitivity of such software so low to the point where new viruses are often not identified.” (EX-1006 [0006]; EX-2015 ¶ 98.)

Mankins “employs heuristic techniques” (EX-1006 [0006]) of the kind Halperin disparages: it builds a Bloom filter by observing “normal system-call behavior” and then “detect[s] intrusions ... by monitoring the behavior of computer programs and comparing the monitored actual behavior with predefined Bloom filter data to identify any anomalous behavior.” (EX-1008 3:21-25 & 40-43; *compare id.* with EX-1006 [0006]; *see* EX-2015 ¶ 99.) Because Halperin teaches away from Petitioner's proposed modification of Halperin with Mankins, a skilled artisan would

not have been motivated to perform that modification. *See DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 567 F.3d 1314, 1326 (Fed. Cir. 2009) (“An inference of nonobviousness is especially strong where the prior art's teachings undermine the very reason being proffered as to why a person of ordinary skill would have combined the known elements.”).

C. There was no reasonable expectation of success in combining Halperin with Mankins.

Given Petitioner's failure to adequately describe a combination *at all*, it is unsurprising that Petitioner has also failed to describe a combination of Halperin with Mankins that would offer a POSITA a reasonable expectation of success in constructing the claimed invention in the '250 patent.

As mentioned above, Petitioner proposes running the two pieces of software in sequence, contending that “if the system does not identify malware based on the behavioral data ... Mankins's process would begin.” (Petition 23.) But this proposed sequencing is nonsensical because Halperin never affirmatively indicates that it has not identified malware. Halperin's continuous and rolling analysis of network behavior offers no point at which a skilled artisan would conclude that Halperin has “not identified malware” such that, according to Petitioner, “Mankins's process would begin.” (EX-2015 ¶ 84.) Nor does Petitioner indicate whether

Halperin's process would continue or stop after handing over the reins to Mankins.
(EX-2015 ¶ 84.)

Further, Petitioner's run-Halperin-then-run-Mankins approach contradicts its passing mention of "incorporat[ing] Mankins's Bloom filter data object" into Halperin as "a sensor that uses a behavioral approach." (See Petition 21.) Petitioner's attempt to map the purported combination's teachings to the '250 patent's limitations further confirms that Petitioner does not conceive of Mankins as a Halperin-type "sensor." During the steps of the '250 patent's process to which Petitioner contends Halperin's sensors are analogues, Petitioner cites Halperin *alone* as teaching the limitations.² (Petition 24-29 (limitations [1P]–[1C]).) Whether this is a failure by Petitioner to accurately describe its combination or a failure to show that a POSITA would have been motivated to combine its collection of disparate references with a reasonable expectation of success, it is fatal to the petition. See *Elektro*, 81 F.4th at 1374.

Petitioner has improperly used the '250 patent as a blueprint onto which it has mapped inapt teachings from the prior art. See *Echolochem, Inc. v. S. Cal. Edison Co.*, 227 F.3d 1361, 1371-72 (Fed. Cir. 2000) ("tak[ing] the inventor's disclosure as

² As discussed further below, Petitioner is incorrect. See *infra* Section VI.A.

a blueprint for piecing together the prior art to defeat patentability[] [is] the essence of hindsight”). As Dr. Medvidovic explains, what Petitioner has provided is not a motivation to combine Petitioner’s references with a reasonable expectation of success, but a defective defense-in-depth strategy. (EX-2015 ¶¶ 85-86.) Defense-in-depth requires coordination between and among defensive layers—otherwise, the defense strategy may leave uncovered holes while simultaneously expending unnecessary resources erecting redundant defenses elsewhere. (EX-2015 ¶¶ 85-86.)

Petitioner’s hindsight-motivated strategy is evident from the face of the prior art; nothing in Halperin or Mankins suggests that combining them would provide additional protection. *See Juniper Networks, Inc. v Correct Transmission, LLC*, IPR2021-00682, Paper 26 at 35 (PTAB Oct. 3, 2022) (holding claims non-obvious because petitioner did not “identify a common problem, a single critical parameter, or explicit guidance” tying the references together and “mere allegation of similarities in two references” does not establish a reasonable expectation of success). To the contrary, the systems in Halperin and Mankins are likely to produce differing results, leaving a skilled artisan with less confidence that either system’s results are correct. To be sure, a skilled artisan might be capable of reconciling a situation in which, for example, Halperin sounds the all-clear but Mankins suggests the presence of malware—but *Petitioner nowhere explains* how such an artisan

would go about that task in Petitioner's undercooked combination. *Pfizer, Inc. v. Apotex, Inc.*, 480 F.3d 1348, 1365 (Fed. Cir. 2007) ("to have a reasonable expectation of success, one must be motivated to do more than ... try each of numerous possible choices until one possibly arrived at a successful result, where the prior art gave [] no ... direction as to which of many possible choices is likely to be successful.").

D. Adding Kirby further destabilizes Petitioner's combination.

Petitioner's Halperin-Mankins-Kirby combination fails because its Halperin-Mankins combination fails, as just discussed. But Petitioner's perfunctory argument for incorporating Kirby into that failed combination (Petition 46-47) is also separately defective.

First, Halperin teaches away from Kirby just like it teaches away from Mankins. Kirby expressly employs a "behavioral heuristic" that "identifies actions known to be common to undesirable activity." (EX-1007 3:7-9.) But Halperin teaches away from "software that employs heuristic techniques." (EX-1006 [0006].) Indeed, because Kirby combines multiple methods of malware detection, it would be even more likely to generate the "false alarm[s]" Halperin warns of. (EX-1006 [0006]; EX-2015 ¶ 105.)

Second, as with the Halperin-Mankins combination, Petitioner fails to explain how the Halperin-Mankins-Kirby trio would operate. Petitioner adverts vaguely to “incorporating Kirby’s signature matching techniques to aid in the classification of objects”—but does not explain into *what* a skilled artisan would incorporate those techniques, or *how* that artisan would go about such an integration. (EX-2015 ¶¶ 103-106.) For example, Petitioner does not rely on Kirby for limitations [1C] or [13C], the step at which the ’250 patent classifies objects as malware. (See Petition 48.)

Even less well explained is Petitioner’s contention about “Kirby’s teachings regarding identifying as malware objects that ‘touch’ malware.” (Petition 47.) Nowhere in Halperin’s system does Halperin teach identifying any object as malware, and so the notion that Halperin could incorporate Kirby’s teaching that objects “touch[ing]” malware likely are malware is at best a non sequitur. The suggestion is similarly nonsensical to directing someone to reach, blind, into a Scrabble letter-tile bag and select the “Q” tile—and any tiles touching that tile.

And, **third**, Petitioner cites no evidence that a POSITA would have had a reasonable expectation of success in combining Kirby with its Halperin-Mankins combination. (See Petition 46-47.) Thus, even if it had made the separate preponderance showing of a motivation to combine the references, Petitioner’s

Kirby combination would fail. *Juniper Networks, Inc.*, IPR2021-00682, Paper 26 at 35-36 (holding claims non-obvious because petitioner did “not sufficiently evidence[] a reasonable expectation of success” in combining the references); *Teva Pharms. USA, Inc. v. Corcept Therapeutics, Inc.*, 18 F.4th 1377, 1383 (Fed. Cir. 2021) (affirming no reasonable expectation of success; though petitioner “describe[d] the prior art as ‘disclos[ing] the problem, ... the solution, ... and the way to find the solution’” “[i]n doing so, it ignore[d] the reasonable-expectation-of-success requirement”).

**VI. PETITIONER FAILS TO SHOW THAT ITS COMBINATIONS
RENDER OBVIOUS EVERY LIMITATION OF THE CHALLENGED
CLAIMS**

Petitioner offers two grounds—one based on Halperin in view of Mankins, and a second based on Halperin in view of Mankins and Kirby, to contend that the '250 patent is unpatentable:

Ground 1	Section 103(a)	Obvious over Halperin in view of Mankins	Claims 1, 2, 12-14, 24
Ground 2	Section 103(a)	Obvious over Halperin in view of Mankins and Kirby	Claims 1-30

Petitioner has not made a sufficient showing of unpatentability under either ground.

In its first ground, Petitioner relies on Halperin alone for claim limitations [1P]–[1C] and on Halperin in view of Mankins for limitations [1D]–[1I]. For the reasons set forth below, Halperin fails to disclose [1C]; and Halperin in view of Mankins fails to disclose each element of limitations [1D] and [1I]. In its second ground, Petitioner again relies on Halperin alone for several analogous limitations in independent Claim 25, and those contentions in Ground 2 fail for the same reason as they do in Ground 1.

And as discussed above in Section V, Petitioner identifies neither a motivation to combine Halperin with Mankins—much less with Mankins *and* Kirby—nor a reasonable expectation of success in yielding the invention in the '250 patent from those unmotivated combinations. The '250 patent and Halperin teach fundamentally different methods because each sought to solve a different problem. The inclusion of Mankins and Kirby does nothing to overcome this glaring incompatibility. The '250 patent is directed to an efficient, object-centric determination and classification of a particular object as malware or not. (Section II.B, *supra*; EX-2015 ¶ 47.) Halperin teaches detecting patterns of suspicious, collective behavior of multiple computers that may be indicative of a malware infection and preventing further spread of that possible infection within a network. (Section IV.A, *supra*; EX-2015 ¶ 70.)

Dr. Newman does not dispute that the two systems’ approaches are different—Halperin “look[s] at the behavior of a system as a whole and determine[s] whether it may be infected with malware,” and the ’250 patent “look[s] at a specific object and determine[s] whether the behavior of that object indicates that it’s malware.” (EX-2016 at 45:20-46:14; EX-1004 at ¶ 40 (“either [an unknown program] or the system as a whole may be monitored”).) Because Halperin’s method is not directed to an object-specific analysis, Halperin fails to disclose multiple limitations discussed below—particularly, those relating to object-specific data or the computer object itself.

A. [1C], [13C] Petitioner fails to show that Halperin discloses or renders obvious “*classifying the computer object as malware*”

Halperin does not disclose classifying a computer object as malware based on the received data. Halperin’s detection of potential viral *activity* and initiation of an alarm or containment actions does not mean Halperin discloses or renders obvious *classifying* any particular *object* as malware.

Claim Limitation	Claim Language
[1C]	“ <i>classifying the computer object as malware when the data indicates that the computer object is malware</i> ”

[13C]	<i>“the base computer being constructed and arranged to classify the computer object as malware when the base computer determines that the data indicates that the computer object is malware”</i>
-------	--

1. Halperin does not disclose or render obvious “classifying the computer object as malware.”

Petitioner fails to account for the fundamental disconnect between Halperin's and the '250 patent's systems. Halperin does not disclose “classifying [a] computer object,” whether as malware or otherwise. Instead, Halperin teaches a protocol for “detecting malicious *activity* in a computer network”—the behavioral symptoms indicating the possible presence of malware—and responding to those symptoms by halting traffic or alerting a human being. (EX-1006 Abstract (emphasis added), EX-2015 ¶ 120.) Nothing in that disclosure teaches (or requires) the “classifying a computer object as malware.” Halperin's system detects malware *symptoms*—what the Abstract accurately describes as “detecting malicious activity”—and prevents that malware's *further spread*. And as Halperin makes clear, spotting symptoms and halting the contagion may be accomplished without “classifying [any] computer object as malware.” (See EX-1001 20:19.)

Halperin discloses a system and method of “detection and prevention of malicious computer programs.” (EX-1006 [0002].) But neither “detect[ing]” nor “prevent[ing]” malware is itself a process of *classifying* a particular object as malware. For example, as Halperin recognizes, malware may be “detected” and its spread “prevented” by a signature-based system based on a *previous* classification of a piece of code as malware. (*E.g.*, EX-1006 [0005] (describing the “detect[ion] of known computer viruses”).)

Detecting symptoms of malware, preventing malware’s spread, and classifying an object as malware are distinct processes, as Dr. Medvidovic explains: At the airport, a trained dog can be used to *detect* the presence of illicit substances in luggage and alert authorities to *prevent* the luggage’s owner from boarding an aircraft. (EX-2015 ¶ 121.) But, until the person’s belongings are searched, it is impossible to *classify* the object responsible for causing the dog to alert. The substance might be drugs, explosives, or simply a false positive like a dog treat. The mere act of *preventing* the person from boarding the plane may increase safety, but it does not *classify* the object responsible. In the same way, Halperin may detect and prevent the spread of malware, but it does not classify any computer objects. (EX-2015 ¶¶ 122-123.)

Additionally, just because Halperin employs containment actions that take place “when malicious software *has been identified*” does not mean Halperin discloses a classification of any computer program. (EX-1006 [0073].) Halperin treats the “malicious software’s” identification as a *fait accompli*, not as a step in Halperin’s process—Halperin nowhere teaches *how* to identify (or classify) any object.

Halperin’s use of the passive voice in “has been identified” further indicates that Halperin does not disclose how to identify the malicious software. Petitioner recognizes that a similar passive-voice construction—“defined” instead of “identified”—elsewhere in Halperin does not indicate that Halperin has taught a relevant artisan how to perform the “defining”: with regard to claim [1E], Petitioner contends that “[i]n Halperin, ‘target behavior profiles’ of acceptable and/or unacceptable behavior are ‘defined,’ ... but Halperin does not state explicitly how the ‘target behavior profiles’ are generated.” (Petition 30.) So, too, with the identification of malicious software: the passage in Halperin on which Petitioner relies merely assumes that “software has been identified”; it does not “state explicitly how” software is “identified.”

Instead of teaching how to identify any particular piece of “malicious software” in a way that could be considered “classifying a computer object as

malware,” Halperin teaches only how to identify the signs that *some* malicious software may be operating on a network. But Halperin never points to any particular piece of software as the culprit. “In the absence of ... other supporting evidence to enable a skilled artisan to make the claimed invention, a standalone § 103 reference must enable the portions of its disclosure being relied on.” *Raytheon Techs. Corp. v. Gen. Elec. Co.*, 993 F.3d 1374, 1381 (Fed. Cir. 2021). Here, Petitioner relies *only* on Halperin’s non-enabled disclosure of prior identification of malware—it proffers no “other supporting evidence” that a skilled artisan could make the claimed invention. *See id.*; Petition 28-29. Petitioner accordingly has not shown by a preponderance of the evidence that these claims are unpatentable.

2. Petitioner’s arguments that Halperin teaches classifying a “computer object” are flawed.

Petitioner proffers two arguments for this limitation. Both fail.

Petitioner’s *first* argument asserts that Halperin teaches classifying the computer object as malware because the base computer determines whether correlated target behavior matches a predefined suspicious behavior pattern. (Petition 28.) In support, Petitioner string-cites six paragraphs from Halperin’s specification without any indication of what activity in Halperin purportedly corresponds to a classification. But as explained just above, nothing in Halperin teaches such a classification. *See supra* Section VI.A.1. Instead, Petitioner’s cited

paragraphs merely show Halperin's *behavior*-detection method for identifying malware *symptoms*.

Petitioner's attempt to augment its argument with two parenthetical quotes does not remedy this deficiency. (Petition 28 (quoting EX-1006 [0087], [0061]).) The first of Petitioner's quotes—read in the context of its surrounding paragraphs—merely confirms that “identif[ication of] malicious code” is referring to the detection of **activity** that may indicate the presence of a virus before initiating an alarm or containment actions. (EX-1006 [0087]; EX-2015 ¶ 124.) To support the interpretation that Halperin's paragraph [0087] discloses this limitation, Petitioner relies on several unsupported leaps from Dr. Newman, including arguing that Halperin's “*retrospective external* input” could be “informed by Halperin's system” itself. (EX-2016 128:19-129:3 (emphasis added).) This is illogical. To support that argument, Dr. Newman relies on an interpretation that in the '250 patent, an “object” is an “event.” (EX-2016 108:18-109:8; 137:18-138:4.) Not only is this interpretation not preserved anywhere in the petition, but, as explained above in Section II.C.2, an event cannot be an object.

Similarly, the second quote confirms that Halperin is directed to detecting “virus **activity**” (with or without a subsequent check by an operator as to whether this activity was legitimate or potentially virus-related). (EX-1006 [0061]; EX-2015

¶ 125.) Detecting virus activity is not the same as classifying the virus itself. Petitioner relies on its expert's contention that "target *behavior*" or "virus *activity*" may themselves be "suspect object[s]." (Petition 28; EX-1004 ¶ 81 (emphasis added).) But as Dr. Newman conceded in his deposition, that reasoning is based on his erroneous and unpreserved interpretation of "object" to include "event[s]." (EX-2016 108:18-109:8; 137:18-138:4.) That interpretation is wrong. *See supra* Section II.C.2. Ultimately, neither of Petitioner's parenthetical quotes discloses determining whether a particular object is malware, much less classifying it as such. (EX-2015 ¶¶ 124-126.)

Moreover, Petitioner's assertion that Halperin discloses classification of a particular object as malware as a result of detecting a suspicious behavior pattern is incorrect. After Halperin collects target behavior, Halperin determines whether the collective network behavior corresponds to a predefined pattern of suspicious behavior—*e.g.*, whether a certain percentage of computers in the network are behaving in a certain way. (EX-1006 [0047]-[0056].) That is, Halperin detects potential "virus **activity**"—suspicious actions taken **at the computer or network level**. (EX-1006 [0061].) This determination "indicat[es] that a computer virus may have infected those computers." (EX-1006 [0047].) "Upon detecting a suspicious behavior pattern, [the] server [] may initiate one or more virus containment actions."

(EX-1006 [0057].) Halperin's detection of potential viral activity on a network and actions to contain the spread of a potential virus do not disclose a determination of whether a *particular object* is malware or classification of such object. (EX-2015 ¶¶ 126-127.)

A skilled artisan would understand that Halperin's omission of object classification is by design, because nothing about Halperin's subsequent containment disclosures contemplates—much less requires—classifying any particular object as malware. (EX-2015 ¶ 127.) Instead, those containment actions either halt network traffic to or from affected computers or notify a human that symptoms of malware may have been detected. (See EX-1006 [0023]-[0030].) The containment actions do not themselves constitute “classifying.” And even if the human Halperin's system alerts were to manually search for and identify the object that caused Halperin's sensors to trigger, that would not teach the '250 patent's method: as discussed above, the plain and ordinary meaning of “classifying” in the '250 patent claims excludes human classification. See *supra* Section II.C.1.

None of Halperin's exemplary containment actions require a prior classification of a particular object as malware. (See EX-1006 [0023]-[0030]; EX-2015 ¶¶ 129-131.) Petitioner's reliance on language explaining containment actions a system might take “when malicious software *has been identified*” is unavailing

because—as discussed above—it merely treats identification as something already accomplished without teaching how to accomplish it. (See Petition 28 (citing EX-1006 [0073])(emphasis added).) Similarly, Petitioner's citation to Halperin's references to “identify[ing] malicious code (from here on referred to as ‘viruses’)” merely repeats the same non-program-specific method Halperin employs throughout—using sensors to find *symptoms* of viral activity on a network. (Petition 28 (citing EX-1006 at [0087]).) Something more than imprecise language broadly (and incorrectly) categorizing the system's operation as “identify[ing] malicious code” is necessary to teach that procedure. (*Cf.* Petition 30 (conceding that Halperin's statement that “target behavior profiles” are “defined” “does not state explicitly how the ‘target behavior profiles’ are generated”).)

Petitioner's *second* argument asserts that a POSITA “would have further understood” Halperin's “virus alarm” or “virus containment actions” to occur “because the system has classified the computer object as malware.” (Petition 28-29.) This appears to be an attempt at an inherency argument. But Petitioner fails to explain or support its conclusory assertions or explain why initiating an alarm or containment action would necessarily mean that an object was classified as malware. (See *id.*; EX-1004 ¶¶ 84-85.) Such assertions are inadequate. See, e.g., *Lumenis Be Ltd v. BTL Healthcare Techs. A.S.*, IPR2021-01280, Paper 8 at 22 (PTAB Mar. 9,

2022) (declining to institute because petitioner's argument lacked explanation and the Board was unwilling to speculate); *Roland Corp. v. Inmusic Brands, Inc.*, IPR2018-00335, Paper 14 at 17 (PTAB July 2, 2018); *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999).

Halperin's initiation of a virus alarm or containment actions does not mean that Halperin's system has identified malware—much less classified it. (EX-2015 ¶¶ 129-131.) Dr. Medvidovic provides an apt analogy. (EX-2015 ¶ 130.) When a patient calls his doctor and says he is experiencing coughing, sneezing, and a runny nose, the doctor may recommend, among other things, that the patient stay home from work or school to avoid potentially infecting others. (EX-2015 ¶ 130.) But the fact that the doctor detected symptoms and prescribed containment actions does not mean that he also necessarily diagnosed the patient as having *any* virus—much less that the doctor identified (*i.e.*, classified) the *particular* virus responsible for the patient's symptoms. (EX-2015 ¶ 130.)

Similarly, here, when Halperin detects malware symptoms, it initiates actions to prevent further spread from potentially infected systems. (EX-2015 ¶ 131.) But Halperin's disclosure of actions taken to *contain* a possible virus on a network does not inherently disclose or indicate to a POSITA that Halperin has *classified* any particular object as malware—particularly where none of the remedial containment

actions Halperin discloses requires identifying a particular object as malware. (See EX-1006 [0048]-[0056]; EX-2015 ¶ 131.)

Other than as discussed above, Dr. Newman's declaration merely repeats the same conclusory assertions contained in the petition and is, therefore, entitled to little weight. *Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 9 at 15 (PTAB Aug. 24, 2022); 37 C.F.R. § 42.65(a). Accordingly, Petitioner does not establish a reasonable likelihood of success in meeting its burden to show that Halperin discloses "*classifying ... the computer object as malware.*"

B. [1D] Petitioner fails to show that Halperin in view of Mankins discloses or renders obvious "*initially classifying the computer object as not malware.*"

Neither Halperin nor Mankins teaches "classifying the computer object as not malware." In the '250 patent, this affirmative initial classification of non-malware *as* non-malware is important because it allows the program to continue to run and precedes generating an object-specific mask—a resource-intensive undertaking. (See EX-1001 4:21-27; EX-2015 ¶ 134.)

As with the other limitations discussed above, Halperin does not disclose or render obvious *classifying* any particular *object* as malware. Halperin's corresponding lack of reaction during its continuous scanning when that scanning

finds no symptoms of malware does not amount to a classification of any object as non-malware.

Claim Limitation	Claim Language
[1D]	<i>“when the determining does not indicate that the computer object is malware, initially classifying the computer object as not malware”</i>
[13D]	<i>“wherein the base computer is constructed and arranged to automatically generate a mask for an object that is initially classed not as malware”³</i>

³ Petitioner contends that claim 13 is “materially identical to claim 1, except that claim 13 is an apparatus version” and raises only the same challenges to that claim as it raised to claim 1. (Petition 42, 50.) Petitioner thus cannot rely on any difference in the claim language to support the obviousness of either of those claims if the other is nonobvious. *See Netflix*, 84 F.4th at 1382 (arguments not raised in the petition are forfeited).

Petitioner begins by stating that “[b]oth Halperin and Mankins disclose or render obvious this limitation,” leaving Patent Owner and the Board to guess as to its proposed manner of combining the two references.

With regard to Halperin, Petitioner suggests this limitation is “simply the converse” of [1C] and that Halperin therefore renders it obvious because “from Halperin’s disclosure of classifying an object as malware,” a relevant artisan would have understood “that the same object would be classified as not malware if the determining step generated the opposite result.” (Petition 29.) But as discussed above, Halperin does not teach classifying *any* object as *anything*—whether malware or not malware. *See supra* Section VI.A.

And even if Halperin—counterfactually—*did* classify objects as malware, Halperin’s continuous, rolling examination of behavior without finding any suspicious behavior at some given point in time is not the same as affirmatively classifying anything as not malware. Without this step, the purported Halperin-Mankins combination does not know for which object it should “automatically generat[e] a mask.” (EX-1001 20:34.)

To put it another way, Petitioner’s contention is that *remaining silent* about whether symptoms of malware are present on a system is the same as *affirmatively stating* that all objects on the system *are not* malware—that absence of evidence (of

malware) is evidence of absence (of malware). That reasoning is a fallacy. *See In re Rail Freight Fuel Surcharge Antitrust Litig.*, 725 F.3d 244, 254 (D.C. Cir. 2013) (“[A]bsence of evidence is not evidence of absence.”); *New England Power Generators Ass’n, Inc. v. FERC*, 707 F.3d 364, 370 n.3 (D.C. Cir. 2013) (“‘ $P \supset Q$ ’ does not mean ‘ $\neg P \supset \neg Q$ ’”—i.e., *P* is evidence of *Q* does not mean the absence of *P* evidences the absence of *Q*).

Petitioner then argues that Mankins separately discloses this limitation (Petition 30-32), but Mankins does not fill Halperin’s gap. Petitioner points to Mankins’s training phase—the period in which a Mankins system builds a Bloom filter—and contends that that phase is classifying the computer object as not malware. But that contention ignores the other half of the claim limitation, which requires that the system classify the computer object as non-malware based upon a “determining” that “does not indicate that the computer object is malware.” Mankins does not perform any “determining” in its learning phase. (EX-2015 ¶¶ 135.) Indeed, because the Bloom filter according to Mankins is a data object representing a program’s *benign* behavior, running the program during Mankins’s training phase indicates a *preexisting* classification of the program as non-malware—not that the program is being “classified” as non-malware *during* that phase, when such a classification would come too late. (EX-1008 at 6:60-7:7; EX-

2015 ¶ 135.) Thus, Mankins does not remedy Halperin’s shortcoming with regards to this limitation and neither discloses nor renders obvious this limitation.

C. [1I] Petitioner fails to show that Halperin in view of Mankins discloses or renders obvious “*reclassifying the computer object as malware*”

As addressed above, Petitioner fails to identify a motivation to combine Halperin with Mankins to arrive at limitation [1I] or [13I]. Moreover, these references do not disclose the limitation.

Claim Limitation	Claim Language
[1I]	<i>“reclassifying the computer object as malware when the actual monitored behaviour extends beyond that permitted by the mask.”</i>
[13I]	<i>“wherein the base computer is constructed and arranged [such that] ... the object is reclassified as malware when the actual monitored behaviour extends beyond that permitted by the mask.”</i>

In the ’250 patent, the crux of this limitation is that the computer object, after an initial classification—as in limitation [1D]—is then examined again and *re*-classified. The ’250 patent explains that this “provides an efficient and effective way of monitoring the behaviour of an object that has been classified or regarded as

good and allows the object to be reclassified quickly as malware if the behaviour of the object warrants it.” (EX-1001 at 4:27-30.)

The two distinct steps of *classifying* and *reclassifying* an object in the '250 patent are crucial because they allow the operator to “continually assess whether the object continues to be safe or not.” (EX-1001 at 15:10-11.) If an object previously classified as safe “attempts to connect to a known unsafe website, database or to take action that is known as generally being action only taken by unsafe programs, then again the object may be considered to be unsafe.” (EX-1001 at 15:22-25.) An object that is initially classified as not malware based on its behavior is not guaranteed to be safe forever, because the object may change its behavior or because what is considered safe has changed. The purpose of the reclassification is so that the system can recognize the object as unsafe when it is seen again in the future. (EX-1001 at 8:4-33 (explaining that a community database is populated over time with information about which objects are safe or unsafe, with information relating to each object run on all of the connected remote computers).) Without the ability to initially classify and then *reclassify*, the methods and systems of the '250 patent would be stagnant and thus vulnerable. (EX-2015 ¶ 144.)

Petitioner relies on the same disclosure in Halperin for this limitation as it does for limitation [1C]. In trying to show that Halperin discloses the initial

classification of the computer object in [1C], Petitioner argues that “when Halperin takes actions responsive to the detection of a virus, such as to ‘activate a virus alarm’ or ‘initiate virus containment actions,’ it is because the system has classified the computer object as malware.” (Petition 28-29.) Petitioner then points to the same disclosure to argue that Halperin also discloses a *reclassification*. (Petition 42 (“Halperin’s disclosure of ‘virus containment actions’ ... include reclassifying the computer object as malware.”).) Petitioner’s recycling of the same disclosure for two separate limitations reveals the weakness in its argument. As explained above, Halperin does not teach *any* classification of an object. But even if the disclosures on which Petitioner relies *did* teach classifying an object, that classification happens *only once*.

The claim requires the *reclassification* of a specific computer program: the one that was *initially classified* as non-malware and whose mask was later exceeded. Thus, Petitioner’s argument fails on its face because neither Halperin nor Mankins ever performs an initial classification in step [1D], as described in Section VI.B above.⁴ Without an initial classification, there can be no *reclassification*.

⁴ Petitioner does not argue Mankins discloses the initial classification in step [1C].

Even setting aside the fact that neither Halperin nor Mankins discloses a second classification, the actions that Petitioner identifies in Halperin and Mankins as teaching “reclassifying” do not constitute classifying *at all*, as required by the ’250 patent. The ’250 patent’s reclassification is used to mark as unsafe objects that were once considered safe, so that when they are seen again the user is aware that they may no longer be considered benign. (EX-1001 8:4-16 (explaining that the base computer stores a database used by the remote computers to determine if an object is safe or unsafe).) The disclosures upon which Petitioner relies do not achieve this result.

Petitioner contends that “if suspicious behavior is detected, ‘the [Halperin] system can ... initiate virus containment actions.’” (Petition 42.) But the seven “virus containment actions” Halperin identifies do not require—or even contemplate—identifying a specific computer object. (EX-1006 [0024-0030], EX-2015 ¶ 145.) Dr. Newman proposes that it would have been “obvious to a POSITA that in the combined Halperin/Mankins system, that taking appropriate ‘virus containment actions’ as described in Halperin would have included reclassifying a program as malware.” (EX-1004 ¶ 112.) But this hand-waving assertion does not explain how a POSITA could have used Halperin’s object-agnostic suspicious-

behavior alert to determine which specific computer object triggered the behavior the Halperin system identified. (EX-2015 ¶ 145.)

Petitioner relies on a similarly vague sentence in Mankins that says that the system may take “predetermined actions designed to protect the computer.” (EX-1008, 11:8-11.) But Mankins does not elaborate on what “predetermined actions designed to protect the computer” may be. Like in Halperin, the predetermined actions may be *preventative* as opposed to a specific classifying action. Further, Mankins does not suggest—and nothing requires—that the responsive action be specific to the program whose system call Mankins’s Bloom filter has flagged as suspicious. (EX-2001 ¶ 146.) For example, Mankins might simply shut the computer down to prevent further malware activity or propagation. (EX-2001 ¶¶ 146-147.) Doing so would not (re-)classify any specific computer program. Because Mankins’s “predetermined actions designed to protect the computer” do not expressly or obviously disclose taking a classifying action with the specific program whose Bloom filter was exceeded, Mankins does not disclose nor render obvious claim [1I].

Petitioner’s shotgun approach to this limitation, collating a list of various *preventative* measures in either of Halperin or Mankins—and leaving the Board to determine the manner of Petitioner’s combination—is ineffective. Petitioner does

not identify a single *reclassification* of a computer object in either of Halperin or Mankins.

- D. [25P] Petitioner fails to show that Halperin in view of Mankins discloses or renders obvious “[a] method of providing data about a computer object from a remote computer to a base computer so that a comparison can be made at the base computer with similar data received from other remote computers, the method comprising.”**

As explained above in Section II.C.3, the preamble of claim 25 is limiting. Petitioner has failed to show that Halperin discloses the essential steps described in the preamble of the claim.

Claim Limitation	Claim Language
[25P]	<i>“A method of providing data about a computer object from a remote computer to a base computer so that a comparison can be made at the base computer with similar data received from other remote computers, the method comprising:</i>

Petitioner's adoption by reference of its argument regarding the [1P] limitation for the preamble of claim 25 is nonsensical. Nothing in its [1P] argument explains how Halperin teaches “a method of providing data about a computer object from a remote computer to a base computer so that a comparison can be made at the base computer with similar data received from other remote computers.” (See

Petition 24-25.) Petitioner has presented no argument to show that that Halperin teaches the preamble of claim 25. *See Netflix, Inc. v. DivX, LLC*, 84 F.4th 1371, 1377 (Fed. Cir. 2023) (noting that Board may not “find[] a claim unpatentable based on theories not presented in the petition” and “should ... not have to decode a petition to locate additional arguments beyond the ones clearly made”).

E. [2], [14], [25B] Petitioner fails to show that Halperin in view of Mankins discloses or renders obvious “*wherein the data about the computer object that is sent from the plural remote computers to the base computer includes one or more of: ... the current name of the object.*”

Petitioner contends that these claims are obvious over the combination of Halperin and Mankins.⁵ (Petition 50, 60.) But Petitioner does not rely on any disclosure from Mankins (or from Kirby in Ground 2) for the additional limitations in these dependent claims. Petitioner's argument that Halperin alone teaches these claims fails, because Halperin does not include the current name of any computer object when sending data from the remote computers to the base computer.

⁵ Although Petitioner also addresses limitations [2] and [14] in Ground 2, its Ground 2 argument merely adopts by reference its Ground 1 argument. (Petition 48, 50.) Petitioner's Ground 2 argument for [25B] similarly adopts only its Ground 1 arguments for claim limitation [1A] and claims 2 and 14. (Petition 61.)

Claim Limitation	Claim Language
[2]	<i>“A method according to claim 1, wherein the data about the computer object that is sent from the plural remote computers to the base computer includes one or more of: ... the current name of the object”</i>
[14]	<i>“Apparatus according to claim 13, the data includes one or more of: ... the current name of the object”</i>
[25B]	<i>“the data including information about the behaviour of the object running on one or more remote computers and including one or more of: ... the current name of the object;”</i>

Petitioner contends that Halperin teaches these claims because one of the detection “sensors” of Halperin is triggered by “computers in the network sending the same file attachment via a message in the last 15 minutes.” (Petition 54.) But even if Petitioner were correct that a POSITA would have understood that the name of file attachments is included as part of the information sent to the base computer in Halperin, Petitioner does not explain why the file attachment described in Halperin is the same as the “computer object” in question in the ’250 patent, and so does not explain why the name of the file attachment is the “current name of the object.”

These limitations of the '250 patent, in conjunction with their respective independent claims, require that the “data about a computer object” received at the base computer include *both* “information about the behaviour of the object” and “the current name of the object.” (EX-2015 ¶ 149.) That is, the “name of the object” in claims 2 and 14 points to the object responsible for “the behaviour” in limitations [1A] and [13A]. The name of the object does not refer to the filename of an *attachment* sent via a message that was *perhaps* sent by the computer object in question. Put simply, Petitioner does not explain how a file attachment sent via a message could also be responsible for sending that message, and there is no indication in Halperin that the two are related. Petitioner's expert concedes that the portions of Halperin that he cited do not indicate how the filename would identify a specific computer object. (EX-2016 147:21-148:6 (“Q: ... But Halperin doesn't expressly say anything about identifying the name of a specific computer object. Correct? [Objection] [Dr. Newman:] I don't cite anything there that says that.”).)

In fact, it would be illogical to assume that the filename of an attachment sent in a message is identical to the filename of the computer object responsible for sending it. Both experts agree on this point. (See EX-2016 at 60:11-15; EX-2015 ¶ 150) As Dr. Medvidovic explains, computer malware is designed to evade detection. (EX-2015 ¶ 150.) An author of computer malware that propagates via

file attachment would select an innocuous name for the file attachment, such as “resume.pdf” or “client_invoice.xlsx” in order to entice the user to open the attachment to trigger the execution of the embedded malware. (EX-2015 ¶ 150.) However, the malware itself—which is downloaded onto and executed on the user’s machine—is often a different file from the attachment used to spread it. (EX-2015 ¶ 151.) Thus, a POSITA would not find it obvious that “the current name of the computer objects would be sent to the base computer as part of determining whether they are the ‘same file attachment.’” (Petition 43.) This is because the computer objects for sending the file attachments and the file attachments themselves are two different objects. (EX-2015 ¶ 151.)

Halperin alone does not disclose these claim limitations, and Petitioner does not identify any disclosures in Mankins (or Kirby, in Ground 2) that would overcome Halperin’s deficiency. (See Petition 43, 50.) Petitioner therefore has not proved by a preponderance of the evidence that either claim 2 or claim 14 is unpatentable.

VII. CONCLUSION

For the foregoing reasons, Petitioner has failed to show the ’250 patent is unpatentable.

Patent Owner's Preliminary Response to
Petition for *Inter Partes* Review
Patent No. 8,418,250

Date: January 2, 2024

Respectfully submitted,

KING & SPALDING LLP

/Brian Eutermoser/

Brian Eutermoser
Registration No. 64,058

Attorney for Patent Owner
Open Text Inc.

CERTIFICATION OF WORD COUNT

The undersigned hereby certifies that the portions of the above-captioned **PATENT OWNER RESPONSE** specified in 37 C.F.R. § 42.24 have 13,964 words, in compliance with the 14,000 word limit set forth in 37 C.F.R. § 42.24(a)(1)(i) & 42.24(b)(2). This word count was prepared using Microsoft Word 2022.

Date: January 2, 2024

Respectfully submitted,

KING & SPALDING LLP

/Brian Eutermoser/

Brian Eutermoser
Registration No. 64,058

Attorney for Patent Owner
Open Text Inc.

CERTIFICATE OF SERVICE

The undersigned hereby certifies that the foregoing **PATENT OWNER RESPONSE** and all supporting **EXHIBITS 2015-2017** were served by filing the documents through the Patent Trial and Appeal Case Tracking System (P-TACTS), on January 2, 2024 and courtesy copies were also served via electronic mail, in their entireties on the following:

Benjamin Hershkowitz
(Reg. No. 41,833)
GIBSON, DUNN & CRUTCHER LLP
200 Park Avenue
New York, NY 10166-0193
Tel: (212) 351-2339
bhershkowitz@gibsondunn.com

*Counsel for Petitioners
Sophos Ltd. and Sophos Inc.*

Josh Krevitt (*pro hac vice*)
GIBSON, DUNN & CRUTCHER LLP
1881 Page Mill Road
Palo Alto, CA 94304-1211
Tel: (212) 351-2490
JKrevitt@gibsondunn.com

Brian A. Rosenthal (*pro hac vice*)
Katherine Q. Dominguez (*pro hac vice*)
Vivian Lu
GIBSON, DUNN & CRUTCHER LLP
200 Park Avenue
New York, NY 10166-0193
Tel: (212) 351-2339
BRosenthal@gibsondunn.com
KDominguez@gibsondunn.com
Vlu@gibsondunn.com

Timothy P. Best (*pro hac vice*)
GIBSON, DUNN & CRUTCHER LLP
333 South Grand Avenue
Los Angeles, California 90071-1504
Tel: (213) 229-7659
TBest@gibsondunn.com

*Counsel for Petitioners
Sophos Ltd. and Sophos Inc.*

Patent Owner's Response to
Petition for *Inter Partes* Review
Patent No. 8,418,250

Date: January 2, 2024

KING & SPALDING LLP

/Brian Eutermoser/

Brian Eutermoser
Registration No. 64,058

Attorney for Patent Owner
Open Text Inc.