

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TREND MICRO, INC.

Petitioner,

v.

OPEN TEXT INC.,

Patent Owner.

Case IPR2023-00655

Patent No. 8,418,250

PATENT OWNER'S PRELIMINARY 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

I.	INTRODUCTION	1
II.	THE BOARD SHOULD EXERCISE ITS DISCRETION TO DENY INSTITUTION UNDER 35 U.S.C. § 314(a)	3
A.	Factor 1: The District Court Has Not Granted And Will Not Grant A Stay.....	6
B.	Factor 2: A District Court Trial Is Expected Near—And Probably Before—The Statutory Deadline.	11
C.	Factor 3: The Parties Will Have Invested Significant Resources In The District Court Proceeding.	14
D.	Factor 4: The Issues And Parties Overlap Significantly.....	15
E.	Factor 5: Both Parties Are Parties In The Parallel Litigation.	16
F.	Factor 6: The Petition Does Not Present “Compelling Evidence Of Unpatentability.”	18
III.	THE ’250 PATENT	19
A.	Inadequacy Of The Prior Art & The Need For Improved Malware Detection	19
B.	Innovations Of The ’250 Patent	21
C.	Claim Construction.....	24
D.	Level Of Ordinary Skill In The Art.....	25
IV.	OVERVIEW OF THE RELEVANT PRIOR ART	26
A.	Overview Of Halperin	26
B.	Overview Of Mankins	31
C.	Overview Of Inoue.....	33

V.	THE CHALLENGED CLAIMS ARE PATENTABLE	35
A.	A POSITA Would Have Had No Motivation to Combine Halperin with Mankins or Inoue	36
1.	A POSITA would not have been motivated to combine Halperin with Mankins	36
2.	A POSITA would not have been motivated to combine Halperin with Inoue	40
B.	Petitioner Fails To Establish That Halperin In View Of Mankins or Inoue Renders Obvious The Challenged Claim Limitations.....	42
1.	1[b] Petitioner fails to show that Halperin “discloses” “receiving data about a computer object”	44
2.	1[c] Halperin does not “disclose[]” “determining ... whether the data ... indicates that the computer object is malware”	49
3.	1[d] Petitioner fails to show that Halperin “discloses” “classifying the computer object as malware”	53
4.	1[f] Petitioner fails to show that Halperin in view of Mankins “discloses” “automatically generating ... acceptable behavior for the computer object”	57
5.	1[f] Petitioner fails to show that Halperin in view of Inoue “discloses” “automatically generating ... acceptable behavior for the computer object”	64
C.	Patent Owner Reserves Its Right To Contest Portions Of The Petition Not Addressed.....	66
D.	Objective Indicia Of Nonobviousness Further Support The Challenged Claims’ Patentability.....	67
VI.	CONCLUSION.....	67

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Apple Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 11 (PTAB Jun. 1, 2021).....	<i>passim</i>
<i>Apple Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 15 (PTAB May 13, 2020)	11
<i>Belden Inc. v. Berk-Tek LLC</i> , 805 F.3d 1064 (Fed. Cir. 2015)	59
<i>Cizion, LLC v. Kerr Machine Co.</i> , PGR2020-00065, Paper 10 (Dec. 3, 2020).....	12
<i>Commscope Techs. v. Dali Wireless, Inc.</i> , IPR2022-01242, Paper 23 (PTAB Feb. 27, 2023).....	21, 22
<i>Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.</i> , 800 F.3d 1375 (Fed. Cir. 2015)	36
<i>Harmonic Inc. v. Avid Tech., Inc.</i> , 815 F.3d 1356 (Fed. Cir. 2016)	36
<i>Intellectual Ventures I LLC v. Hewlett Packard Enterprise Co.</i> , No. 6:21-cv-00596-ADA (W.D. Tex. Dec. 22, 2022).....	7
<i>Lumenis Be Ltd. v. BTL Healthcare Techs. A.S.</i> , IPR2021-01280, Paper 8 (PTAB Mar. 9, 2022).....	36
<i>Luxshare Precision Industry v. Amphenol Corp.</i> , IPR2022-00132, Paper 10 (PTAB May 3, 2022)	11
<i>mCom IP, LLC v. Cisco Sys., Inc.</i> , No. 6:22-00261-ADA (W.D. Tex. Oct. 20, 2022).....	7
<i>Motorola Mobility LLC v. Ironworks Patents, LLC</i> , IPR2021-00421, Paper 11 (PTAB July 22, 2021).....	11

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

<i>Nintendo Co. v. Gamevice, Inc.</i> , IPR2020-01197, Paper 13 (PTAB Jan. 12, 2021)	11
<i>Nokia of Am. Corp. v. Godo Kaisha IP Bridge</i> , IPR2022-00755, Paper 9 (PTAB Nov. 2, 2022).....	17
<i>NXP USA, Inc. v. Impinj, Inc.</i> , PGR2022-00005, Paper 20 (PTAB Aug. 25, 2022)	12
<i>Samsung Elecs. Co. v. Clear Imaging Research LLC</i> , IPR2020-01399, Paper 13 (PTAB Feb. 3, 2021).....	11
<i>Sonrai Memory Ltd. v. Kingston Tech. Co., Inc.</i> , No. 6:21-cv-1284-ADA (W.D. Tex. Oct. 18, 2022)	7
<i>Sophos, Ltd. v. Open Text Inc.</i> , IPR2022-00699, Paper 1 (PTAB Mar. 8, 2023)	13
<i>UCB v. Actavis Labs. UT, Inc.</i> , 65 F.4th 679, 687 (Fed. Cir. 2023)	56
<i>Vizio, Inc. v. Maxell, Ltd.</i> , IPR2022-01459, Paper 7 (PTAB Apr. 11, 2023)	16, 17
Statutes	
35 U.S.C. § 312(a)(3).....	35
35 U.S.C. § 314(a)	3, 18
37 C.F.R. § 42.22(a)(2).....	35
AMERICA INVENTS ACT, Congress. H. Rept. 112-98 Part 1 (June 1, 2011).....	1
Other References	
Memo, <i>Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation</i> (June 21, 2022)	4, 12, 13, 18

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)

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

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. Trend Micro, Inc.</i> , Case No. 6:22-cv-239-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. 53 (W.D. Tex. Sept. 7, 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)

Open Text Inc. ("Patent Owner") submits this Preliminary Response to the Petition of Trend Micro, Inc. ("Petitioner") seeking *inter partes* review ("IPR") of the U.S. Patent No. 8,418,250 ("the '250 patent").

I. INTRODUCTION

The Board should decline to institute IPR of the '250 patent because an IPR would duplicate the parties' district court litigation efforts and the petition presents weak invalidity arguments based on fundamentally dissimilar prior art that does not disclose multiple limitations.

The Board should deny institution under *Fintiv* because instituting this IPR would undermine Congress's stated goal in creating IPR proceedings: to "limit unnecessary and counterproductive litigation costs." H.R. Rep. No. 112-98, pt. 1, at 40 (2011, 2011 U.S.C.C.A.N. 67, 69 (post-grant reviews were meant to be "quick and cost effective alternatives to litigations")). The '250 patent's validity has been challenged in *four* parallel Western District of Texas ("WDTX") cases, now consolidated, and those cases are scheduled to proceed to trial two months *before* a final written decision on this petition would be due. *See Webroot v. AO Kaspersky Lab*, No. 6:22-cv-00243-ADA-DTG (WDTX) (lead case) ("Kaspersky"); *Webroot v. Trend Micro Inc.*, No. 6:22-cv-00239-ADA-DTG (WDTX); *Webroot v. Sophos Ltd.*, No. 6:22-cv-00240-ADA-DTG (WDTX); *Webroot v. CrowdStrike Inc.*, No.

6:22-cv-00241-ADA-DTG (WDTX); *see also Webroot v. Forcepoint LLC*, No. 6:22-cv-00342-ADA-DTG (WDTX) (consolidated with the other cases, but '250 patent is not at issue).

Here, moreover, there is there no reasonable possibility of a district court stay. The district court proceedings involve thirty-nine patents and four additional defendants-counterclaimants, three of whom will litigate the validity of the '250 patent based on the same primary reference asserted here regardless of whether this petition is instituted. The district court has construed claim terms for the '250 patent, and by the time the Board issues its institution decision (in October 2023), the parties will be deep into fact discovery. Instituting this petition will at a minimum generate expensive and needless duplication of effort by Patent Owner, the Board, and the district court, and as with all duplicative proceedings it poses the risk of inconsistent decisions.

The Board should deny institution for the additional reason that Petitioner fails to demonstrate a reasonable likelihood that it will prevail on the merits. The '250 patent uses a mask to detect and classify computer objects—in essence, *individual programs*—as malware. But all of Petitioner's grounds rely on U.S. Patent Application Publication No. 2004/0111632 to Halperin ("Halperin") as a primary reference. Halperin evaluates *aggregate computer and network-level behavior* to

find symptoms that may indicate malware has infected a computer or network—it does not analyze the operation of a single program or associate activity with specific programs. Even when Halperin's system determines malware may be present in the system, Halperin uses containment actions—such as suspending a computer from sending messages—and does not “*classif[y]*” the malware as recited in the '250 patent claims. In short, these are entirely different systems that provide very different approaches to solve different problems.

Petitioner also fails to identify a plausible motivation to combine Halperin's network-intrusion-response system with the fundamentally different system in 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, a POSITA would have had no motivation to add kernel-level, program-specific

Patent Owner respectfully requests that the Board deny the petition.

II. THE BOARD SHOULD EXERCISE ITS DISCRETION TO DENY INSTITUTION UNDER 35 U.S.C. § 314(a)

In view of the parallel litigations, in which a district court will adjudicate—prior to the Board—the same grounds raised in this petition, the Board should exercise its discretionary denial authority under 35 U.S.C. § 314(a).

Because Congress's goal in enacting post-grant proceedings like IPRs was "to ... limit unnecessary and counterproductive litigation costs," Director Vidal has instructed the Board not to institute IPRs that would "increase, rather than limit, litigation costs" being incurred in parallel litigation. Director's Memo, *Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation* ("Guidance") at 1 (June 21, 2022).

Instituting this IPR would cause just such an increase. The reasons are easy to see: a district court stay is out of the question, and the district court will consider the very same arguments presented in the petition. Accordingly, for the next year, the Board and the district court will labor to decide the very same issues with the district court well ahead of—and scheduled to reach a final decision months before—the Board. The year-long parallel tracks of two adjudicatory bodies will "increase, rather than limit, litigation costs," create "potential conflict between the ... (PTAB) and district court proceedings," and generally increase "the potential for inefficiency and gamesmanship." Guidance at 1. This is precisely the situation that Director Vidal instructed the Board to avoid when she instructed the Board to take "a holistic view of whether efficiency and integrity of the system are best served by denying or instituting review." *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 6 (PTAB Jun. 1, 2021).

Confirming this conclusion, every one of the *Fintiv* factors weighs for discretionary denial:

(1) A stay of the co-pending district court litigation—involving thirty-nine patents and five defendants—is vanishingly unlikely. Petitioner cannot avoid that trial because it has petitioned for IPR of only some of the patents asserted against it. And this IPR will similarly not narrow the issues on which discovery and trial must proceed. The '250 patent is currently asserted against three other defendants, one of whom has not petitioned for IPR of the '250 patent. Moreover, *only one* of Petitioner's codefendants has agreed not to litigate the validity of the '250 patent in the district court regardless of any institution decisions by the Board. Claims have been construed, and discovery is well underway.

(2) The district court trial is expected two months *before* any Board final written decision.

(3) The parties and district court have already invested heavily in carrying that parallel litigation past claim construction—*six months* before any anticipated institution decision—and well into discovery.

(4) The issues raised in the petition are the same as those the district court has before it, and Petitioner has offered no stipulation that it will not pursue the very

same arguments in both proceedings. Petitioner's co-defendants, moreover, have raised the same prior art before the district court.

(5) Petitioner will get its chance to make its arguments before the district court because Petitioner is a defendant in the parallel litigation.

(6) As shown below, the petition's grounds of unpatentability are weak. Thus, every *Fintiv* factor here weighs for discretionary denial. The district court is well on its way to an efficient resolution of the disputes between Patent Owner, Petitioner, and Petitioner's codefendants. Both the goals of the AIA and the general welfare of the patent system would be best served by denying institution.

A. Factor 1: The District Court Has Not Granted And Will Not Grant A Stay.

The first *Fintiv* factor asks "whether the court granted a stay or evidence exists that one may be granted if a proceeding is instituted." *Fintiv*, IPR2020-00019, Paper 11 at 6. This factor weighs strongly in favor of denial because the answer to both questions is **no**: no stay has been requested or ordered, and Petitioner has identified no evidence that one will (or even might) be. To the contrary, specific evidence indicates that a stay is vanishingly unlikely.

First, the district court considers issue simplification “the most important factor” in deciding whether to stay patent litigation¹—and here, a stay would not simplify anything. As circumstances stand now, the validity of the '250 patent will be litigated in the consolidated action whether or not this petition is instituted, because the patent is asserted against a defendant that has not sought Board review and two other defendants that have sought Board review have not agreed that they will not pursue in the district court the same grounds raised in this petition.

¹ See, e.g., EX-2003 [*Sonrai Memory Ltd. v. Kingston Tech. Co., Inc.*, No. 6:21-cv-1284-ADA, slip op. at *4-5 (W.D. Tex. Oct. 18, 2022)] (denying a stay following IPR institution because issue simplification was the “most important factor” and the Court had already held a *Markman* hearing and “expended significant resources”); EX-2004 [*mCom IP, LLC v. Cisco Sys., Inc.*, No. 6:22-00261-ADA, slip op. at *5 (W.D. Tex. Oct. 20, 2022)] (finding “simplification is not likely” where the IPR did not challenge all of the claims asserted in the district court, since the other claims would still require a trial); EX-2005 [*Intellectual Ventures I LLC v. Hewlett Packard Enterprise Co.*, No. 6:21-cv-00596-ADA, slip op. at *7 (W.D. Tex. Dec. 22, 2022)](denying a stay following IPR institution , as even IPR estoppel would not simplify the invalidity case, given the defendants’ reliance on prior art systems).

What is more, it will be litigated alongside *thirty-eight other* patents asserted by and against Petitioner and *four other* defendants—issues that the district court is highly unlikely to delay deciding in favor of letting the Board proceed with *this* IPR. The case against Petitioner alone involves seventeen patents—eleven asserted by Patent Owner and six asserted in counterclaims by Petitioner.

The table below illustrates the patents, parties, and IPR petitions involved in this highly complex litigation:

**Table 1. Patent Owner's Asserted Patents
Against Each WDTX Defendant and Resulting Review Petitions**

Patent	Kaspersky	CrowdStrike	Forcepoint	Sophos	Trend Micro
8,418,250	Asserted; no IPR	IPR2023- 00289	<i>N/A</i>	IPR2023- 0699	IPR2023- 00655
8,726,389	Asserted; no IPR	IPR2023- 00556	Asserted; no IPR	IPR2023- 00633	Asserted; no IPR
10,599,844	Asserted; no IPR	Asserted; no IPR	Asserted; no IPR	IPR2023- 00528	IPR2023- 00662
9,413,721	<i>N/A</i>	<i>N/A</i>	Asserted; no IPR	IPR2023- 00491	<i>N/A</i>
9,578,045	IPR2023- 01199	IPR2023- 00124	<i>N/A</i>	Asserted; no IPR	IPR2023- 00677
10,257,224	IPR2023- 01011	IPR2023- 00126	<i>N/A</i>	Asserted; no IPR	IPR2023- 00657
10,284,591	IPR2023- 00895	IPR2022- 01522	<i>N/A</i>	Asserted; no IPR	IPR2023- 00692
8,856,505	Asserted; no IPR	IPR2023- 01159	<i>N/A</i>	Asserted; no IPR	Asserted; no IPR
8,181,244	<i>N/A</i>	IPR2023- 01158	<i>N/A</i>	Asserted; no IPR	Asserted; no IPR
8,201,243	Asserted; no IPR	IPR2023- 01052	Asserted; no IPR	IPR2023- 00731	Asserted; no IPR
8,719,932	Asserted; no IPR	IPR2023- 01051	Asserted; no IPR	IPR2023- 00732	Asserted; no IPR
11,409,869	IPR2023- 01160	IPR2023- 01053	Asserted; no IPR	PGR2023- 00031	Asserted; no IPR
8,763,123	<i>N/A</i>	IPR2023- 1050	<i>N/A</i>	Asserted; no IPR	<i>N/A</i>
8,438,386	<i>N/A</i>	<i>N/A</i>	IPR2023- 00784	<i>N/A</i>	<i>N/A</i>
10,025,928	<i>N/A</i>	<i>N/A</i>	IPR2023- 00786	<i>N/A</i>	<i>N/A</i>

Because of the large number of claims and issues unrelated to the validity of the '250 patent, institution of this IPR will have no effect on the district court's stay calculus.

Second, the district court has been clear in public comments that it will not stay cases where, as here, the court can resolve the case as quickly as the Board can:

[T]here's no unfairness to either side by not staying a case for the PTAB when I can try the case as quickly as they can resolve it.... And so far, I think that's proven to be true.

(EX-2006 at 3.) Indeed, as commentators have observed, "Judge Albright almost never stays proceedings to let the PTAB make its decision first." (*Id.* at 2.)

Judge Albright's public comments apply squarely to the situation here. The median time-to-trial in WDTX is 28.2 months. On that timeline, trial is expected in this case in mid-July 2024. Trial is currently scheduled for August 2024. That Judge Albright has made public comments indicating that his practice is to deny a stay in precisely the circumstances presented here is further evidence (if any were needed) that he would not stay the WDTX case in response to institution of this petition. Given Judge Albright's stated policies and the complexity of the parallel proceedings, imagining circumstances in which Judge Albright would cast aside his stated preferences and policies to grant the extraordinary remedy of a stay of *this* parallel case in particular requires far-reaching speculation.

Even where no stay has been requested, the Board has found Factor 1 to favor discretionary denial where (as here) “Petitioner present[ed] no evidence that it has requested a stay [] or that a stay is likely.” *Luxshare Precision Industry v. Amphenol Corp.*, IPR2022-00132, Paper 10, at 9 (PTAB May 3, 2022). And such a decision is particularly appropriate where the circumstances show that “a stay ... is unlikely.” *Nintendo Co. v. Gamevice, Inc.*, IPR2020-01197, Paper 13, at 11-12 (PTAB Jan. 12, 2021); *Samsung Elecs. Co. v. Clear Imaging Research LLC*, IPR2020-01399, Paper 13 at 11-12 (PTAB Feb. 3, 2021) (holding Factor 1 may weigh for denial where there is evidence that the district court is unwilling “to avoid duplicative efforts and await the PTAB’s final resolution”); *Motorola Mobility LLC v. Ironworks Patents, LLC*, IPR2021-00421, Paper 11 at 7-8 (PTAB July 22, 2021).

Given the full circumstances present here, it is not plausible that the district court will stay its multi-patent, multi-defendant case so that the Board can reach a *later* decision on *just one* patent. Factor 1 supports discretionary denial. *Samsung*, IPR2020-01399, Paper 13 at 11-12; *Motorola*, IPR 2021-00421, Paper 11 at 7-8.

B. Factor 2: A District Court Trial Is Expected Near—And Probably Before—The Statutory Deadline.

Factor 2—the implications of the district court and IPR schedules—also supports denial. “[A] parallel proceeding in an advanced state implicates considerations of efficiency and fairness, which can serve as an independent reason

to apply discretion to deny institution.” *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 15 at 11-12, 17 (PTAB May 13, 2020) (*informative*).

Here, institution would “increase, rather than limit, litigation costs.” Guidance at 1. Specifically, it would double the parties’ costs and obligations to litigate the ’250 patent’s validity in separate forums. The IPR trial would overlap almost entirely with the parallel district court litigation. WDTX’s median time-to-trial as of March 2023 is 28.2 months (EX-2007 at 37), indicating a trial by mid-July 2024—a date consistent with the currently scheduled trial date of August 19, 2024. (EX-2008 at 9.) Should the Board institute, its final written decision deadline would be October 17, 2024, nearly two months after the scheduled trial date. In other words, the IPR proceedings will overlap almost entirely with the parallel litigation and likely will not produce a result before the trial.

But the likelihood that the Board’s decision will not come until months after the district court’s is only part of the story. The Board should consider “the duplicative efforts that will occur during the [] duration of the inter partes review.” *NXP USA, Inc. v. Impinj, Inc.*, PGR2022-00005, Paper 20 at 7 (PTAB Aug. 25, 2022). And here, even in the unlikely event that the Board reaches decision several months before the deadline, the parties will spend nearly a year expending precisely

the “duplicative,” “unnecessary[,] and counterproductive litigation costs” that the AIA was designed to avoid. *Fintiv*, IPR2020-00019, Paper 11 at 10; Guidance at 1.

Nor is even that all. Petitioner's co-defendant Sophos filed a petition for IPR of the '250 patent raising a substantively identical ground based on the Halperin-Mankins combination. *See generally Sophos, Ltd. v. Open Text Inc.*, IPR2022-00699, Paper 1 (PTAB Mar. 8, 2023). Accordingly, if it institutes, the Board faces the specter of *three* parallel proceedings, including two before the Board, all addressing the same arguments. Nothing about such a glut of duplicative litigation furthers the AIA's purposes.

In support of institution, Petitioner offers only the bare assertion that “trial is currently set for August 19, 2024, so there is time to complete inter parte [*sic*] review.” (Petition 84.) But events have moved on. Given that no institution decision is due on this petition until *October* of 2023, the Board would be highly unlikely to “complete inter parte[s] review” before the district court concludes its parallel litigation.

Because the district court will not stay the litigation, the litigation schedule would overlap the IPR trial schedule almost entirely, and the district court's judgment will likely issue *before* any Board decision here, the second *Fintiv* factor also favors discretionary denial.

C. Factor 3: The Parties Will Have Invested Significant Resources In The District Court Proceeding.

Factor 3—the resources invested in the parallel district court proceeding—strongly suggests denial.

To support its position, Petitioner merely states that “it is early in the parallel litigation as fact discovery has not started.” (Petition 84). But in addition to misrepresenting the state of the case *now*, that contention will only become less true as the parties and Board approach the date of any expected decision on institution.

The parties have invested significant resources in the parallel litigation over the last fifteen months. For example, the parties long ago exchanged infringement and invalidity contentions (in July 2022 and September 2022 respectively), and fact discovery is underway. Claim construction, too, is long since complete (EX-2009), “indicat[ing] that the court and parties have invested sufficient time in the parallel proceeding to favor denial.” *Fintiv*, IPR2020-00019, Paper 11 at 10.

And while *Fintiv* directs consideration of resource investment at the time of institution, the overlapping schedules here counsel the Board to further consider the redundant resources that *will be* wasted should IPR be instituted. *Id.* at 6 (requiring “a holistic view of whether efficiency and integrity of the system are best served by denying or instituting review”).

Petitioner did not rush to file this petition. Instead, it took 357 of the allowable 365 days following the March 10, 2022 service of Patent Owner's complaint. Moreover, the petition came eight months after Petitioner received Patent Owner's infringement contentions and six months after Petitioner served its own invalidity contentions. (EX-2010 at 1-2; EX-2011 at 2.) Petitioner offers no explanation for its delay, and that delay supports denial. *Fintiv*, IPR2020-00019, Paper 11 at 11-12.

D. Factor 4: The Issues And Parties Overlap Significantly.

Factor 4—the degree of overlap between the issues in the petition and those before the district court—also strongly supports denial.

Regarding this fourth factor, *Fintiv* explains that “concerns of inefficiency and the possibility of conflicting decisions [are] particularly strong” where “the Board [is] presented with substantially identical prior art arguments ... at issue in district court.” *Fintiv*, IPR2020-00019, Paper 11 at 12. “Accordingly, if the petition includes the same or substantially the same claims, grounds, arguments, and evidence as presented in the parallel proceeding, this fact has favored denial.” *Id.*

Here, the petition identifies substantially the same prior art arguments that Petitioner and co-defendants Sophos and CrowdStrike have put before the district court. (EX-2012 at 12, EX-2013 at 5-7.) Moreover, Kasperksy identifies substantially similar grounds of purported invalidity based on the Halperin

reference. (EX-2014, App'x A at 1.) Of Petitioner's co-defendants, only Sophos has offered any sort of stipulation that it will not pursue entirely duplicative litigation. And Petitioner itself has not offered *any* stipulation that it will not simultaneously pursue in the district court *precisely the same arguments* based on *precisely the same grounds* asserted in its petition during the pendency of any IPR instituted from that petition. (*See* Petition 84.)

Factor 4 counsels denial.

E. Factor 5: Both Parties Are Parties In The Parallel Litigation.

Factor 5—the overlap of the parties between the petition and the parallel litigation—suggests denial because (1) Petitioner is a defendant in the parallel litigation and, as such, will have a full, fair opportunity to litigate the petitioned grounds, and (2) the issues Petitioner raises are the same ones its co-defendants have *also* raised in the parallel litigation.

Under *Fintiv*, “[i]f a petitioner is unrelated to a defendant in an earlier court proceeding, the Board has weighed this fact against exercising discretion to deny institution.” *Fintiv*, IPR2020-00019, Paper 11 at 13-14. Here, by contrast, Petitioner is not just related; it *is* a defendant in the litigation. Accordingly, the circumstances raise no fairness concerns that denial would deprive Petitioner of its “day in court.” *See Vizio, Inc. v. Maxell, Ltd.*, IPR2022-01459, Paper 7 at 21-22 (PTAB Apr. 11,

2023) (“Because Petitioner is the litigation defendant, factor 5 weighs in favor of discretionary denial”); *Nokia of Am. Corp. v. Godo Kaisha IP Bridge*, IPR2022-00755, Paper 9 at 13 (PTAB Nov. 2, 2022) (holding that Factor 5 weighed for discretionary denial where Petitioner was defendant in parallel proceeding).

That other defendants are also litigating the petitioned issues at the district court further suggests denial. *Fintiv* holds that, “[e]ven when a petitioner is unrelated to a defendant, however, if the issues are the same as, or substantially similar to, those ... about to be litigated, or other circumstances weigh against redoing the work of another tribunal, the Board may, nonetheless, exercise the authority to deny institution.” *Id.* at 14. Because of this consideration, *Fintiv* goes on to require that even “[a]n unrelated petitioner ... discuss why addressing the same or substantially the same issues would not be duplicative of the [other] case even if the petition is brought by a different party.” *Id.*

Petitioner offers no such explanation here. That other defendants are currently litigating the same issues raised in the petition raises serious concerns of duplicative work by adjudicators, and any public-policy concerns regarding patent validity will be thoroughly addressed by the parallel litigation. Thus, that *other* defendants in the district court proceedings are *also* litigating the petitioned grounds weighs yet further for discretionary denial.

F. Factor 6: The Petition Does Not Present “Compelling Evidence Of Unpatentability.”

Factor 6—the merits of the petition—also counsels denial because, as explained below, the petition fails to show a “reasonable likelihood” that Petitioner would prevail on any challenged claim. *See* 35 U.S.C. § 314(a). Petitioner’s proposed combination is an awkward, Frankensteinian assemblage of incompatible references that Petitioner tries to justify only with a thin and unconvincing motivation-to-combine argument. And, just as importantly, Petitioner contends that Halperin “discloses” several key claim limitations concerning the *classification* of *individual computer objects*—limitations that Halperin, with its object-agnostic focus on aggregate, computer-level behavior on a network simply does not teach. Because the petition fails to show a reasonable likelihood of success on the merits, the petition may not be instituted.

Even if the petition did pass the threshold of “reasonable likelihood” (it does not), it does not even approach the higher threshold of a “compelling” unpatentability challenge that could preclude discretionary denial. Guidance at 4. “[T]he compelling merits standard is a higher standard than the standard for institution set by statute.” *Commscope Techs. v. Dali Wireless, Inc.*, IPR 2022-01242, Paper 23, 3 (PTAB Feb. 27, 2023). That higher standard requires determining that Petitioner is not just *reasonably* but rather “*highly* likely” to

succeed on the merits. *Id.* at 4. Petitioner cannot meet that standard. As shown below, Petitioner's grounds are particularly weak.

In sum, all six *Fintiv* factors support denial. Patent Owner therefore respectfully requests that the Board exercise its discretion to deny institution.

III. THE '250 PATENT

A. Inadequacy Of The Prior Art & The Need For Improved Malware Detection

"Malware" generally refers to a computer file or object that is or contains malicious code, *e.g.*, viruses, trojans, worms, etc. (EX-2001 ¶28.)

Much of the prior art anti-malware software focused on scanning and stopping known malware. (EX-1001 1:18-2:18.) For example, some prior art anti-virus software analyzed objects for "signatures"—something within the object that was known to be indicative of malware. (EX-1001 1:18-28.) Whenever new malware was detected, service providers would craft new signatures and release them as updates to their anti-malware programs. (*Id.*) Other prior art systems checked 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 prior art 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-2001 ¶¶30-32.)

Other anti-malware systems sought to protect from 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, where a detailed analysis of the file would be performed. (EX-1001 2:4-11.) That analysis typically required human involvement and as a result could “still take days.” (*Id.*) Other prior art systems similarly relied on a human administrator to “make a better informed though still ‘manual’ decision as to whether or not the file is safe to run.” (*E.g.*, EX-1001 2:18-27.) 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-2001 ¶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-2001 ¶¶29-31.)

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 cls. 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 time-consuming need for preliminary detailed 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-2001 ¶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. A key differentiator from the signature-based prior art is that the '250 patent further discloses automatically generating a mask to monitor behavior of the object to determine whether reclassification is needed. (EX-1001 2:65-3:7; EX-2001 ¶45.) This method can be used in the exemplary system illustrated below:

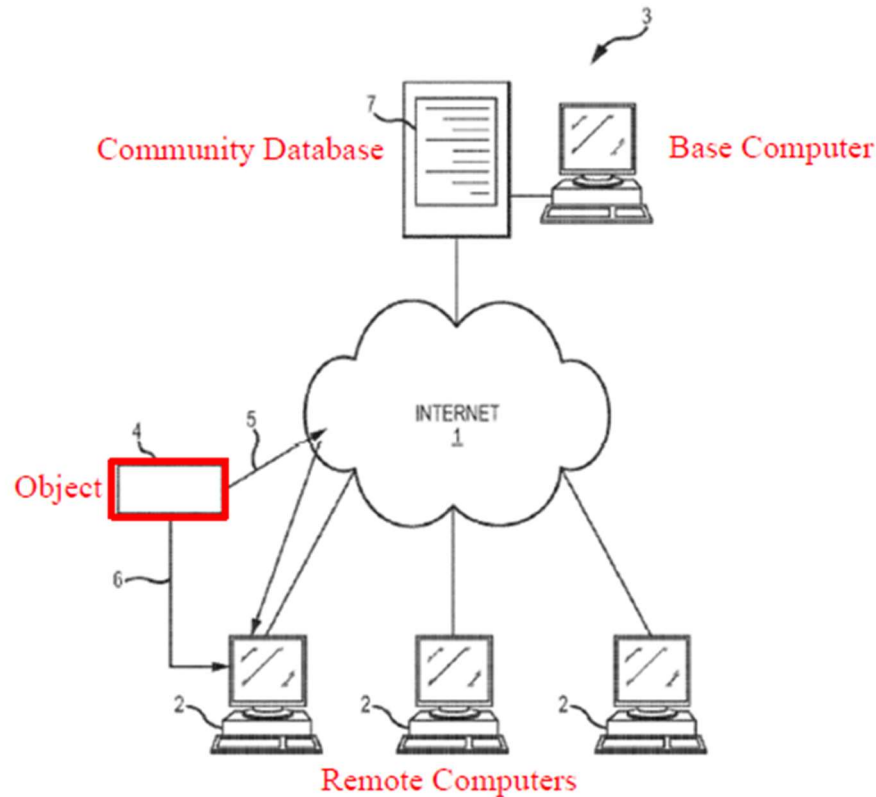


FIG.1

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

When an unknown object is run at a remote computer, data related to the object is passed to the base computer for storing 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” allowing 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

marked “safe” or “unsafe” according to its designation in the database, and the system provides signature, a message, or both to the remote computer (allowing 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 sent that might instruct the remote computer not to run the object or inform the user of the risk and offer the choice of running it. (EX-1001 9:17-25, Fig. 2 (steps 30-31).)

To ensure protection even where an object is allowed to run on a remote computer, 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 indicate, for example, details of any ports or interfaces that might be required to be opened to allow such communications, any databases that may be interrogated by that object. (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-2001 ¶47.)

If the behavior of the object is 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, this behavior is flagged, and the process not allowed to run. (EX-1001 15:47-52.) By continuously monitoring a process each time it is run, any behavior that differs from previous behavior can be noted and the mask updated, or the process reclassified 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 system decreased reliance on signature verification and negated the need for manual, time-consuming "deep analysis" of an object. (*Id.*; EX-2001 ¶49.)

C. Claim Construction

Petitioner concedes that no dispute exists over the construction of claim terms for the purposes of its petition. (Petition 26-27.) On March 16, 2023, the district court concluded that no terms of the '250 patent were indefinite and construed all

terms as having their plain and ordinary meaning.² (EX-2009 3-5, 8.) Accordingly, Patent Owner and its expert apply the plain and ordinary meaning of all claim terms. (EX-2001 ¶51.)

D. 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 would have had 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-2001 ¶¶52-56.) More advanced degrees or training in a related discipline or both can compensate for shorter work experience. (*Id.*)

² The court provided additional clarifying remarks regarding certain terms' plain and ordinary meaning. For brevity and because those clarifications are not critical to the issues addressed in this preliminary response, Patent Owner does not reproduce those clarifying remarks here and instead includes the district court's claim construction order as an exhibit.

IV. OVERVIEW OF THE RELEVANT PRIOR ART

A. Overview Of Halperin

Halperin is directed to detecting patterns of collective computer behavior on a network that suggest the presence of viral activity and taking containment actions to halt further spread. (EX-1003 Abstract, [0036]-[0057].)

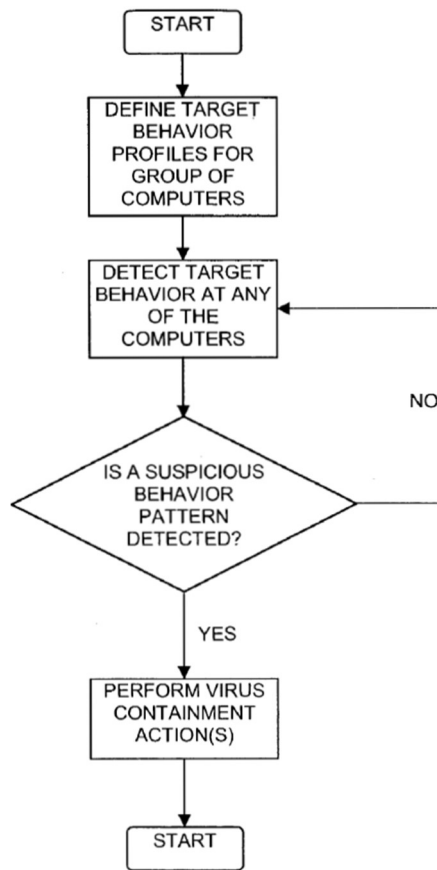
Halperin observed that some prior art antivirus software identified virus behavior by characterizing a program's legitimate behavior and identifying any deviations from such behavior. (EX-1003 [0006].) Halperin criticized this type of object-specific technique as often resulting in false alarms due to the fact that computer user behavior tends to vary over time and between users. (EX-1003 [0006]; EX-2001 ¶58.) Halperin found 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. (*Id.*)

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-1003 [0003].) Some viruses were even capable of spreading on their own “with little or no intervention on the part of the computer user.” (EX-1003 [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-1003 [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-1003 [0037]; EX-2001 ¶60.) Rather than monitor or analyze a particular computer object—like the '250 patent—Halperin's proposed solution detects suspicious patterns of “target behavior” across a group of computers on a network (indicating the presence of a potential virus in the network) and takes one or more of what Halperin calls “virus containment” actions. (EX-1003 Abstract, [0038]-[0057]; EX-2001 ¶60.) Figure 6 illustrates this overall process:



(EX-1003, Fig. 6.)

Halperin starts by defining one or more “target behavior profiles” for a group of computers on a network. (EX-1003 [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-1003 [0039]-[0046].) That is, whereas the ’250 patent looks at behavior

information for the *same or similar objects*—for example, an individual computer program—across multiple computers on a network, Halperin looks at the behavior of multiple computers on a network exhibiting the *same computer behavior*. (EX-1003 [0037]-[0038]; EX-2001 ¶61.)

Halperin then employs 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-1003 [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-1003 [0047]; EX-2001 ¶62.)

The exemplary patterns of suspicious behavior Halperin discloses all relate to a “certain percentage of the computers in the network” all doing the same thing, most often within a relatively short amount of time. (EX-1003 [0048]-[0056]; EX-2001 ¶63.) 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,” a virus may be present on the network. (EX-1003 [0048].) That is, whereas the ’250 patent relies on data that is specific to a particular computer object to determine whether that object is malware, Halperin relies on data that is not tied to a particular object—instead, it seeks to determine whether a certain percentage of computers on the

network are behaving unexpectedly. The system does not have—or need—knowledge of any particular computer object or objects that may be causing the target behavior by the computers on the network. (EX-2001 ¶¶63-64.)

Halperin teaches that if the server detects a suspicious behavior pattern, it may initiate “virus containment actions.” (EX-1003 [0057], [0023]-[0030].) Because the system proffers no means of identifying a particular computer object—such as a piece of software—as the malware culprit or determining whether an object is or is not malware, Halperin’s proposed solutions amount to either telling a human that there might be a problem (*id.* [0025]–[0027]) or halting the flow of information across the network (*id.* [0024], [0028]–[0030]; *see also* EX-2001 ¶64). Halperin does not teach removing the offending software. (*See* EX-1001 [0024]–[0030]; EX-2001 ¶64.)

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 the *network* shows signs of a malware infection. (EX-1003 [0037]–[0038]; EX-2001 ¶65.)

B. Overview Of Mankins

Mankins is directed to detecting intrusion in a computer system by monitoring system calls within the computer's operating system (OS) kernel. (EX-1004 Abstract.) 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-2001 ¶67.) Mankins treats Bloom filters as within the knowledge of a relevant artisan. (See EX-1004 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-2001 ¶67.) 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 false positives—values that are not within the set but that nonetheless return a positive result. (*Id.*)

Mankins describes a Bloom filter data object as "a set of hash functions and a bit vector ... of a given length." (EX-1004 8:19-22.) As relevant here, 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-1004 7:44-46.) Doing so allows the system to “compactly store data representative of predefined normal program behavior.” (EX-1004 7:51-52.)

Mankins also notes that Bloom filters generate false positives, but do so at a known rate correlated to the bit vector's length. (EX-1004 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-1004 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-1004 8:49-52).

Mankins teaches monitoring a computer program's system calls within the OS kernel³ to define the program's expected behavior—including, optionally,

³ Mankins defines the kernel as the portion of the operating system “configured to interface computer programs ... with shared resources.” (EX-1004 4:11-13.)

monitoring system calls while running the program under training conditions. (EX-1004 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-1004 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-1004 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. According to one embodiment of the system, the comparison may be used to detect anomalous behavior. (EX-1004 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-1004 11:11:8-11.)

C. Overview Of Inoue

“Anomaly Intrusion Detection in Dynamic Execution Environments” by Hajime Inoue and Stephanie Forrest (“Inoue,” EX-1005⁴) is a paper allegedly published in *Proceedings of the 2002 Workshop on New Security Paradigms* on

⁴ Petitioner's Exhibit 1005 consists of two documents: Inoue and the Declaration of Ingrid Hsieh-Yee, Ph.D., which purports to authenticate Inoue as prior art. Petitioner's Exhibit list does not refer to the Declaration of Dr. Hsieh-Yee.

September 23, 2002. (*See* EX-1005-10, Declaration of Ingrid Hsieh-Yee, Ph.D., ¶ 27.)

Inoue describes what it claims is an improvement to Dynamic Execution Environments, such as the Java Virtual Machine (“JVM”): “dynamic sandboxing.” (EX-1005-1; EX-2001 ¶71.) It teaches “two activities: sandbox generation and sandbox execution” in the context of the JVM. (EX-1005-3; EX-2001 ¶71.)

Sandboxes are security mechanisms integrated into the JVM that allow it to insulate underlying hardware from malicious attacks. (EX-2001 ¶73.) Sandboxes are also program-specific: “each sandbox is customized to a given program and context in which it is executed.” (*Id.*) Sandbox generation involves “running [a] program with an instrumented [Java Virtual Machine].” During sandbox generation, “profiling information is recorded to the sandbox profile” by “accumulating records for each unique behavior” of a program. (*Id.*) “Each time a new method is invoked, its signature is written to a log.” (*Id.*) “This log constitutes the sandbox profile.” (*Id.*)

Then, “[d]uring sandbox execution, behavior that isn’t in the profile is considered anomalous.” (*Id.*) Like Mankins’s Bloom filters, therefore, Inoue’s dynamic sandbox is a means of automatically distinguishing routine (and thus

allowable) program behaviors from novel and anomalous (and thus disallowed) program behaviors. (*Compare id.* with EX-1004 10:66-67.)

Inoue suggests that the “response” to such “anomalous” behavior is “potentially very flexible,” and “could range from logging the anomaly to the application termination.” (EX-1005-3.) The sole preexisting response within the Java security model is “to disallow the attempted behavior.” (*Id.*)

V. THE CHALLENGED CLAIMS ARE PATENTABLE

“In an inter partes review, the burden of persuasion is on the petitioner.... Failure to prove the matter as required by the applicable standard means that the party with the burden of persuasion loses on that point—thus, if the fact trier of the issue is left uncertain, the party with the burden loses.” *Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378-79 (Fed. Cir. 2015); *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (noting petitioners have the “burden from the onset to show with particularity why the patent it challenges is unpatentable”). Accordingly, a petition must include “a detailed explanation of the significance of the evidence.” *See* 37 C.F.R. § 42.22(a)(2); *see also* 35 U.S.C. § 312(a)(3); *Lumenis Be Ltd. v. BTL Healthcare Techs. A.S.*, IPR2021-01280, Paper 8 at 22 (PTAB Mar. 9, 2022) (finding petitioner had not met its burden and declining

to institute because petitioner's argument lacked explanation and the Board was unwilling to speculate).

A. A POSITA Would Have Had No Motivation to Combine Halperin with Mankins or Inoue

1. A POSITA would not have been motivated to combine Halperin with Mankins

A POSITA would not have been motivated to combine Halperin with Mankins. Halperin and Mankins are directed to different detection methodologies, so a POSITA would have had no motivation to combine them. Where Halperin detects malware symptoms by correlating high-level computer actions across a network, Mankins's focus is far narrower: detecting an individual program's intrusion attempts by monitoring that program's system calls within a single computer's OS kernel. (EX-2001 ¶80.) Because of these differences, integrating Mankins with Halperin would have required modifying both references in a way that contradicts each reference's teachings. Indeed, likely recognizing the inoperability of their combination, Petitioner's only proposed combination is that "Mankins can be used as an active quarantine system" to Halperin. (EX-1002 ¶112.) This proposition also entirely disregards that Halperin does not teach how to identify "suspect software" to be quarantined, as discussed in response to Petitioner's arguments related to limitations 1[b] and 1[c] below. (EX-1002 ¶426; EX-2001 ¶88.)

Halperin teaches away from Mankins. The first and most obvious reason a POSITA would not have tried to combine Halperin with Mankins is that Halperin expressly teaches away from using a heuristic system of the type Mankins proposes. (EX-2001 ¶81.) Halperin states that “antivirus software that employs heuristic techniques to identify typical virus behavior by characterizing legitimate software behavior and then identifying any deviation from such behavior Often results 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-1003 [0006].) Halperin, in other words, recognizes that Mankins-style heuristic-detection systems may produce *both* false positives and false *negatives*.

Mankins, for its part, expressly acknowledges that its system produces just the kind of false negatives Halperin suggests are a problem. (*See* EX-1004 at 8:62-64.) A Bloom filter according to Mankins can provide only a *likelihood* that a given action is within the expected behavior of a given computer program. (*See* EX-1004 at 8:62-64; EX-2001 ¶67.) In other words, some percentage of the time (defined by the size of the bit vector in its Bloom filter), the Mankins system approves behavior that it should flag as suspicious—generating a false negative. (EX-2001 ¶83.)

A POSITA understanding the Mankins system's shortcomings would therefore not be motivated to incorporate Mankins-style Bloom filters into a system

according to Halperin, which expressly warns about the very problems Mankins acknowledges. (EX-2001 ¶¶83-84.)

Petitioner's proposed motivation is based on false assumptions. Petitioner argues that “a POSITA would have an expectation of success in making this combination because both references solve the same problem in a complementary manner.” (Petition 28.) This reasoning is flawed on its face because even if both references work to solve the same problem (which they do not), it does not guarantee that a combined implementation will be synergistic. (EX-2001 ¶¶86-87.) Petitioner attempts to support its flawed reasoning by advocating that both Halperin and Mankins are based on observing normal behavior. (Petition 28.) However, combining Halperin with Mankins is not straightforward and risks eliminating the benefits that Petitioner suggests Mankins would offer to Halperin. For example, Petitioner does not explain how a POSITA would determine *which* object or objects are suspicious and should be “wrapped” according to the system in Mankins. (EX-2001 ¶88.) Additionally, Petitioner's conclusory statement that there would have been an expectation of success is based on the equally conclusory position of Dr. Nielson. Without providing any support, including even explaining why or how, Dr. Nielson suggests that “because of the ease of compatibility, the overlapping domains, and the use of standard functionalities, a POSITA would have had a strong

expectation of success in combining these two references.” (EX-1002 ¶ 113.) Certainly, if their combination were so straightforward, Petitioner and Dr. Nielson would have offered more than their *ipse dixit* conclusions. However, as Dr. Medvidovic explains, Petitioner's suggestion is not a compatible system nor a proper combination. (EX-2001 ¶¶87-90.)

A POSITA would have understood that running distributed Bloom filters provides no benefit over either system because it is in fact simply running each system separately. Petitioner's expert proposes implementing Mankins “as an active quarantine system” to Halperin. (EX-1002 ¶112.) Dr. Nielson contends that when Halperin recognizes suspicious activity, “it would have been obvious to wrap the suspect software into a system that can continue to monitor it.” (*Id.*) Petitioner contends that this would offer “an improvement to the accuracy-limiting trade off disclosed in Halperin as a choice between false alarms and malware detection.” (Petition 28.) But running Mankins's Bloom filters on each remote computer in addition to Halperin's network-level behavior detection system is not a combination of the two systems but is instead a “defense-in-depth” strategy involving two distinct pieces of software. (EX-2001 ¶¶92-93.) While a defense-in-depth strategy may provide benefits when the systems are carefully coordinated, here a POSITA would have understood that any such coordination would be meaningless because it is not

necessary to—and might in fact impede—the operation of either system. (EX-2001 ¶93.) Halperin, for its part, provides no indication about *which* object or objects should be analyzed using a Mankins-style Bloom filter. (EX-2001 ¶88.) And where the systems' results disagree, that disagreement would not provide a POSITA with additional clarity and would only cast doubt on both systems' results. (EX-2001 ¶97.)

Distributed Bloom filters also run contrary to Halperin's teaching of detecting aggregate behavior. Halperin teaches that, for detection of malware symptoms to be meaningful, a threshold number of computers must display the same behavior. But a Mankins Bloom filter, by its very nature, essentially reduces that threshold number to *one*—eliminating any possible benefit of Halperin's aggregate-behavior detection. (EX-2001 ¶¶94-95.)

2. A POSITA would not have been motivated to combine Halperin with Inoue

Petitioner's similarly stunted and conclusory motivation-to-combine argument for the purported Halperin-Inoue combination suffers from precisely the same deficiencies as its Halperin-Mankins motivation. (*See* Petition 71; EX-2001 ¶98.)

Halperin teaches away from Inoue. With its dynamic sandbox technique, Inoue first profiles a program's typical behavior by logging records of each behavior

during training, and then uses that profile to classify “behavior that isn’t in the profile [as] ... anomalous.” (EX-1005-3.) Thus, Inoue’s dynamic sandboxing—like Mankins’s Bloom filters—teaches “heuristic techniques to identify typical virus behavior by characterizing legitimate software behavior and then identifying any deviation from such behavior.” (EX-1003 [0006]; EX-2001 ¶99.) Halperin expressly teaches away from that behavior because of its unreliability when user interaction is taken into account. (*See id.*) Inoue, like Mankins, expressly recognizes this very flaw in its own system, pointing out that “[i]nteractive applications, which are closer to the user” might “creat[e] larger numbers of false positives” because of the user’s propensity to “invoke[]” certain “functionality ... at infrequent intervals.” (EX-1005-7.) Thus, for the same reasons it teaches away from the similar Mankins system, Halperin teaches away from Inoue. (EX-1003 [0006]; EX-2001 ¶99.)

Petitioner’s arguments do not solve the problem. Dr. Nielson again offers what is apparently his own boilerplate conclusion that “because of the ease of compatibility, the overlapping domains, and the use of standard functionalities, a POSITA would have had a strong expectation of success in combining these two references.” (EX-1002 ¶ 427; *cf. id.* ¶ 113). Again, though, challenging the rigor of that supposed “ease of compatibility,” Petitioner’s only proposed combination is that “Inoue can be used as an active quarantine system” in Halperin’s system, which as

Dr. Medvidovic explains, is not a compatible system nor a proper combination. (EX-1002 ¶ 426.) This proposition also entirely disregards that Halperin does not teach how to identify “suspect software” to be quarantined, as discussed with respect to limitations 1[b] and 1[c] below. (EX-1002 ¶426; EX-2001 ¶100.)

Fundamentally, Petitioner's purported motivation to combine Halperin with Inoue is at best a motivation to use each separately at around the same time. (EX-2001 ¶101.) Nothing about Petitioner's supposed motivation suggests any interaction between the Halperin system and the Inoue system, or that running the systems together offers any synergistic improvement over running the systems separately. (*Id.*) Petitioner offers no reason that a POSITA would have taken the time to weld together some bits of Inoue (which Petitioner does not identify) with other, similarly unidentified bits of Halperin to arrive at anything like the invention in the '250 patent, instead of simply running the two systems separately to leverage what Petitioner claims are their “different ... capabilities.” (Petition 70.)

B. Petitioner Fails To Establish That Halperin In View Of Mankins or Inoue Renders Obvious The Challenged Claim Limitations

Petitioner does not demonstrate a reasonable likelihood of success in proving any challenged claim unpatentable because the petition fails to establish that Petitioner's prior art discloses or renders obvious every limitation of the independent claims 1, 13, and 25 of the '250 patent:

Ground 1	§ 103(a)	Obvious in view of Halperin and Mankins	1-5, 12-17, and 24
Ground 2	§ 103(a)	Obvious in view of Halperin, Mankins, and Arnold	6 and 18
Ground 3	§ 103(a)	Obvious in view of Halperin, Mankins, Kirby, and Carpentier	7-11, 19-23, and 25-30
Ground 4	§ 103(a)	Obvious in view of Halperin and Inoue	1-5, 12-17, and 24

The ’250 patent and Halperin present fundamentally different methods because each sought to solve a distinct problem. The inclusion of Mankins, Inoue, or Petitioner’s other secondary references does nothing to correct this glaring incompatibility. The ’250 patent is directed to an efficient, object-centric determination of whether a particular object is malware. (Section III.B, *supra*; EX-2001 ¶¶34-49.) Halperin is directed to a method for detecting patterns of suspicious collective behavior of multiple computers—*symptoms* of a malware outbreak—and preventing further spread of that possible outbreak in a network environment. (Section IV.A, *supra*; EX-2001 ¶¶57, 60.) 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 object itself. (EX-2001 ¶¶108-109.)

Petitioner contends that Halperin alone “discloses” key limitations in each of independent claims 1, 13, and 25, and relies on Halperin in view of Mankins or Halperin in view of Inoue to “disclose” others. Accordingly, because Petitioner fails

to establish that Halperin alone “discloses” those limitations of the independent claims and because Petitioner cannot show that a POSITA would have been motivated to combine Halperin with Mankins or Inoue, Petitioner also cannot show that any of its four grounds render obvious dependent claims 2-12, 14-24, and 26-30.⁵

**1. 1[b] Petitioner fails to show that Halperin “discloses”
“receiving data about a computer object”**

Petitioner contends that Halperin alone “discloses” this limitation. Because Petitioner does not explain how Halperin operates on individual objects, Petitioner fails to show that Halperin “discloses” “*receiving data about a computer object.*” (Petition 29-30.)

⁵ Because Patent Owner's arguments related to the Halperin, Mankins, and Inoue references are dispositive as to all Petitioner's contentions regarding the independent claims, Patent Owner need not address Petitioner's other secondary references (Kirby, Arnold, and Carpentier). Patent Owner reserves the right to raise additional arguments based on those reference and a lack of motivation to combine them with Halperin, Mankins, or both.

Claim Limitation	Claim Language
1[b]	<i>“at a base computer, receiving data about a computer object from each of plural remote computers on which the object or similar objects are stored, the data including information about the behaviour of the object running on one or more remote computers”</i>

As explained above, the ’250 patent is directed to determining whether a specific computer object—an executable computer file—is malware. Its object-focused approach distinguishes it from Halperin, which examines aggregate, collective computer behavior, not individual programs, as possible symptoms of malware infection somewhere on a network. (EX-2001 ¶¶110-111.) Because Halperin is not directed at examining individual objects, Petitioner’s contentions that it discloses or renders obvious object-centric portions of the ’250 patent’s claims are nonsensical.

Halperin does not collect data about what objects perform what actions. Instead, Halperin operates more like a tripwire, logging each instance of when a certain event is performed by a computer, as captured by a sensor. (EX-2001 ¶111.) Halperin teaches that “[i]n the detecting phase the system logs every event of a

sensor identifying a suspicious behavior on a specific computer,” which “preferably includes the type of event, the time it occurred and on which computer it occurred (if relevant).” (EX-1003 [0090].) Halperin uses these logs only to determine how many occurrences happened within a predetermined time period. (*Id.*) For example, Halperin teaches that “if in the past 5 minutes there were 3 events of accessing the Windows Registry, the system checks what was the measured probability of having 3 or more computers with a Windows Registry even in the learning phase.” (*Id.*; *see also, e.g.*, EX-1003 [0037] (correlating percentage of computers performing a certain action).) According to Halperin’s detection methodology, there is no need to collect information identifying what objects performed what actions. (EX-2001 ¶111.)

Despite the nature of Halperin’s explicit disclosure, Petitioner obfuscates the difference between the ’250 patent’s object-centric activity and Halperin’s aggregate behavior detection by claiming that “Halperin discloses that ... descriptions of the suspicious behavioral activity *from software* ... on these computers are shared across the network.” (Petition 29.) But this assertion—that “suspicious behavioral activity” must be “from software”—is question-begging.

Indeed, Petitioner's expert, Dr. Nielson, goes a step further and characterizes the "target behavior" as "data about a virus."⁶ (EX-1002 ¶ 126.) In the same breath, Dr. Nielson concedes that "the data received by the central server is behavioral" (not object-specific) but nevertheless attempts to support his prior characterization based on an entirely different mechanism Halperin employs—the decoy emails. (EX-1002 ¶ 128.) Halperin's "target behavior profiles" (EX-1003 [0039]-[0046]) and associated "suspicious behavior patterns" (EX-1003 [0048]-[0056]) are used in technically and functionally distinct embodiments from the decoy emails. (EX-1003 [0086] (separately identifying "**detectors** (e.g. file deletion and Windows Registry access) and **decoys** (e.g. e-mail decoy)."); EX-2001 ¶113) Target *behavior* is not "data about a virus" and relying on Halperin's decoy e-mail sensor does not make it so. (EX-2001 ¶113.)

In the same vein, the decoy e-mail sensor itself is not "data about a virus" nor "data about a computer object," as the limitation requires. (EX-2001 ¶114.) An e-

⁶ Recall that, in the Halperin system, "target behavior" does not refer to *behavior of a target*, but rather to *behavior that the system is targeting*. (EX-1003 [0047].) Petitioner's argument implies the former reading, but Halperin's specification makes clear that only the latter reading is correct.

mail to a decoy address is more akin to a “target behavior” than it is to “data about a computer object,” because there is no way to tell from the email itself which computer object sent that e-mail. (EX-2001 ¶114). Indeed, Halperin explains that a system administrator is required “to determine whether or not the messages were indeed sent by a computer virus.” (EX-1003 [0058].) If the Halperin system could detect the computer object responsible for sending the e-mail based on the decoy e-mail alone, this step would not be necessary. (EX-2001 ¶114). Dr. Nielson’s bare assertion that a “POSITA would have understood ‘infected’ messages to typically include the virus itself” is contradictory to the disclosure of Halperin. (EX-1002 ¶ 130.)

Dr. Medvidovic provides an example elucidating the important distinction between Halperin’s *behavior* detection (whether by “target behavior profiles” or decoy e-mails) and receipt of data “about an object.” (EX-2001 ¶115). Halperin describes one suspicious behavior pattern as “a certain percentage of computers in the network deleting more than 10 files in the last 1 minute.” (EX-1003 [0050].) Dr. Medvidovic notes that several users on different computers all cleaning their hard drives around the same time—for example, after the expiration of a litigation hold—would trigger Halperin’s predefined suspicious behavior pattern. (EX-2001 ¶115). But a report of that human-triggered, non-malware behavior is not “data

about [any one] computer object,” much less evidence that a given “object is malware.” Accordingly, even though the false alarm is, according to Halperin, “an indication that a computer virus may have infected those computers,” it is not an indication that any computer object is malware. (EX-2001 ¶115).

“Target behavior,” “predefined suspicious event[s],” and “decoys” refer broadly to the behavior attributed to a computer in the network, not to data about a computer object. (EX-1003 [0038]-[0046]; EX-2001 ¶116). Accordingly, Petitioner fails to show a reasonable likelihood of success in meeting its burden to prove that Halperin discloses “*receiving data about the computer object.*”

2. 1[c] Halperin does not “disclose[]” “*determining ... whether the data ... indicates that the computer object is malware*”

The '250 patent requires determining that “the data ... indicates that the computer object is malware.” In Halperin, when “target behavior” matches “predefined suspicious behavior patterns” the conclusion is that “a computer virus *may* have infected those *computers.*” (EX-1003 [0047].) Halperin does not suggest the identity of the virus, or even identify an object that *might* be a virus. (EX-2001 ¶117.) Petitioner nevertheless relies on this same false equivalence between objects and computers to support limitation 1[c] in each ground in its petition that addresses that limitation, as well as the analogous limitation in claim 13. (Petition 30, 46, 71.)

Claim Limitation	Claim Language
1[c]	“determining in the base computer whether the data about the computer object received from the plural computers indicates that the computer object is malware;”

Petitioner contends that Halperin “discloses this limitation” by teaching that “a predefined suspicious behavior pattern ... is an indication that a computer virus may have infected ... computers.” (Petition 30.) Petitioner’s bait-and-switch promise that Halperin teaches identification of *individual objects* while pointing to Halperin’s examination of *network-level activity* infects its arguments regarding this limitation, too.

Here, Petitioner contends that Halperin “discloses identifying portions of emails and software ...that are suspicious or malicious.” (Petition 30.) But Halperin does no such thing. Halperin identifies *behavior* that appears to be a symptom of suspicious or malicious software residing somewhere on a network. (Halperin, ¶ 47; EX-2001 ¶118.) It may then *quarantine* emails or otherwise pause network traffic to prevent the spread of the suspected malware, wherever it may be located. (EX-1003 [0058]-[0059].) But, as discussed in detail above in claim 1[b], at no point does it “identify ... emails [or] software”—much less “portions of” either—that are themselves “suspicious or malicious.” (EX-2001 ¶118.) To the contrary, whenever

Halperin talks about determining about whether a particular piece of software might be a virus, it teaches that such “identif[ication]” is *somebody else’s problem*. (EX-2001 ¶114.) Moreover, it teaches that the *somebody else* is a human being—not, for example, a different malware-detection system. (E.g., EX-1003 [0058] (“The server may ... notify a system administrator ... who may then check the quarantine[] to determine whether or not the messages were indeed sent by a computer virus”). As discussed above, if Halperin were capable of determining that particular computer objects are malware, it would not require this extra step of notifying a system administrator to check the quarantine. (EX-2001 ¶114.)

Petitioner’s expert adds no more context to Petitioner’s insubstantial argument. Dr. Nielson posits only that because Halperin teaches detecting when “a computer virus *may* have infected those *computers*”—which, again, is not determining that a specific computer object is malware—then it must be the case Halperin’s virus containment actions are acting on a specific computer object. (EX-1002 ¶ 135.) Further, this *non sequitur* completely ignores that none of Halperin’s “virus containment actions” are tied to a specific object (e.g., “suspending ... all messages sent by computer 100,” “notifying a system administrator that a virus *may* have been detected,” “notifying a user at computer 100 of the suspicious message activity”). (EX-1003 [0024]-[0030]; EX-2001 ¶121.) And again Halperin farms out

any actual virus detection to a third party—teaching that “messages that are addressed to a decoy address” must be “[f]orward[ed] ... to a third party for analysis, such as a company or other body that produces anti-virus software.” (EX-1003 [0025].)

Dr. Medvidovic again provides an example to illustrate the importance of this distinction between the '250 patent and Halperin. (EX-2001 ¶122.) When a patient calls his doctor and says he is experiencing coughing, sneezing, and a runny nose, the doctor may recommend, among other things, that the patient stay home from work or school to avoid passing on the presumed infection to others. But the fact that the doctor detected symptoms and took containment actions does not mean that he also necessarily diagnosed (or classified) the patient as having *any* virus—much less that the doctor identified the *particular* virus responsible for the patient's symptoms. (EX-2001 ¶122.)

Similarly, here, when Halperin detects malware symptoms, it initiates actions to prevent further spread from potentially infected systems. (EX-2001 ¶123.) 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 knowledge that a particular object might be or is

in fact malware. (See EX-1003 [0048]-[0056]; EX-2001 ¶123.) Thus, Dr. Nielson cannot—and does not—explain how any of these containment actions require determining that a specific computer object is responsible for the suspicious behavior.

Petitioner’s cited support for Halperin’s purported disclosure of this limitation instead merely teaches determining that, in the aggregate, suspicious behavior has occurred or is occurring on the network. Petitioner thus falls short of meeting its burden to show that Halperin discloses or renders obvious “*determining ... whether the data ... indicates that **the computer object** is malware.*”

**3. 1[d] Petitioner fails to show that Halperin “discloses”
“*classifying the computer object as malware*”**

Halperin’s teaching of quarantining e-mails sent to decoy addresses does not show that Halperin discloses or renders obvious classifying a particular object as malware. Petitioner relies on the same analysis of Halperin to support limitation 1[d] in each ground in its petition that addresses that limitation, as well as the analogous limitation in claim 13. (Petition 31, 46, 71.)

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

Petitioner's analysis of Halperin with respect to this limitation can be summarized this way: because Halperin discloses quarantining suspicious messages, "the software is quarantined" and thus "classif[ied]." (Petition 31.).

First, as discussed in 1[b] and 1[c] above, Petitioner's sleight-of-hand misdirection from email messages to "the software" is not justified either by Halperin's disclosure or by a skilled artisan's common sense. Here, that problem comes into greater focus. A relevant artisan would have no way of knowing, from the email itself, which object on a remote computer was responsible for sending the email, so the email could not in any relevant sense be considered "data about the object" that sent it. (EX-2001 ¶126.)

Moreover, even assuming *arguendo* that the email *could* be tied back to the object that sent it, nothing about quarantining *the email* suggests that the *object that sent it* has been classified. Thus, while Petitioner argues that quarantining is the same as classifying, the thing being quarantined here is *the wrong thing*—not "the object" but a symptom of the object's actions. (EX-2001 ¶126.)

And in any event, mere quarantining is not classifying. Dr. Medvidovic offers an analogy to elucidate this point. (EX-2001 ¶127.) Halperin's disclosure is akin to suspecting that a pandemic exists because doctors all over the world have found that millions of people all over the world are simultaneously exhibiting signs of illness

(shortness of breath, fevers, etc.). Classifying malware is akin to a determination by one doctor that one individual has COVID-19. The two scenarios are not dependent on each other: a doctor can diagnose an individual patient regardless of the existence of a pandemic and it is possible to suspect that a pandemic exists without knowing the specific virus responsible for it. Similarly, Halperin deduces that a virus is likely responsible for the suspicious symptoms, but does not disclose detecting the specific computer object responsible for creating those suspicious symptoms (actually classifying the computer object). (EX-2001 ¶127.)

At no point does Petitioner or its expert explain how Halperin explicitly “discloses” why a quarantine—holding a message for some period of time—constitutes a classification or otherwise achieves the goals of classifying the computer object.

Relying on the “virus containment actions” described in Halperin, Petitioner’s expert asserts that “[a]ll of these virus containment elements identify potentially malicious computer objects such as software.” (EX 1002 ¶ 140.) As discussed in 1[c] above, these responsive actions largely do not operate on a specific computer object. (EX-1003 [0024]-[0030]; EX-2001 ¶128.)

And in the only instance in which an action might even be arguably tied to a *collection* of objects, Halperin teaches that any identification of malware must be

done by a remote human being. (*EX-1003* [0025].) Dr. Nielson adds that a “[a] POSITA would have also understood that the messages being analyzed *may* contain the malware.” (*EX 1002* ¶ 140 (emphasis added).) Whether a reference “may” disclose a limitation is not the appropriate standard. *E.g., UCB v. Actavis Labs. UT, Inc.*, 65 F.4th 679, 687 (Fed. Cir. 2023) (“To anticipate a claim, a single prior art reference *must* disclose each and every element recited in the claim.”) (cleaned up; emphasis added). And in any event, it is not what Petitioner has argued. (Petition 31 (contending that “Halperin discloses” this limitation.)

Finally, Petitioner relies on Dr. Nielson’s bare conclusion that a “POSITA would have understood ‘infected’ messages to typically include the virus itself” to argue that a quarantined infected message is a classification of the responsible computer object. (*EX-1002* ¶130.) However, as discussed in 1[b], this contradicts the disclosure of Halperin, because if the Halperin system could detect the computer object responsible for sending the e-mail based on the decoy e-mail alone, then no system administrator would be required “to determine whether or not the messages were indeed sent by a computer virus.” (*EX-1003* [0058]; *EX-2001* ¶114.) Recall Dr. Medvidovic’s analogy in 1[c]: as a doctor that recommends his patient to stay home is not necessarily diagnosing the patient with any particular virus, when Halperin detects malware symptoms, Halperin’s disclosure of *containment* actions

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 knowledge of which particular object might be malware. (See EX-1003 [0048]-[0056]; EX-2001 ¶115.)

A skilled artisan would understand that Halperin's front-end omission of object classification is by design, because nothing about Halperin's back-end containment disclosures contemplates—much less requires—classifying any particular object as malware. (EX-2001 ¶126.) 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-1003 [0023]-[0030].) None of Halperin's exemplary containment actions require a prior classification of a particular object as malware. (See EX-1003 [0023]-[0030]; EX-2001 ¶128.) 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.*”

4. 1[f] Petitioner fails to show that Halperin in view of Mankins “discloses” “*automatically generating ... acceptable behavior for the computer object*”

Petitioner's characterization of Halperin's teachings once again suffers from the failure to disambiguate a computer object from a computer network.

Additionally, Petitioner's combination of Halperin and Mankins fails because (1) it is predicated on the patently false assumption that Halperin discloses receiving "a software's system calls from the remote computers" and (2) their combination would defeat any possible benefits of reducing false positives. (Petition 33). Petitioner relies on the same analysis of Halperin and Mankins to challenge limitation 25[e] as it relies on to challenge limitation 1[f]. (Petition 65.)

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

Petitioner contends that Halperin's disclosure of the collection of "behavioral information in a learning database" while in a learning phase teaches "automatically generating ... acceptable behavior for the computer object." (Petition 32.) Petitioner's expert adds that Halperin "discloses automatically monitoring system events" in a learning phase to determine "potentially unacceptable behavior (related to the virus) and acceptable behavior (when no viruses are present)." (EX-1002 ¶ 149.) Yet, regarding "events," Halperin explains "[t]hat information is logged in a central database that logs for each event the **type of event, time of event and on**

which computer it took place.” (EX-1003 [0086] (emphasis added).) Notably, events are not correlated to any particular computer objects, but rather to time and to computers. (*Id.*; EX-2001 ¶132.) Halperin specifically discloses the type of information it collects during its learning phase—conspicuously missing is any object-identifying information. (See EX-1003 [0076] (“Virus behavior may be monitored ... in terms of numbers of message per unit of time, shapes of utilization graphs, such as for disk storage access or CPU usage, ... histograms of e-mail length, ... recipients per message, etc.”).) Petitioner’s argument fails because Petitioner cannot and does not explain how the “events” disclosed in Halperin are correlated to specific computer objects. (EX-2001 ¶132.) Instead, as discussed above, Halperin tracks and correlates only computer or network behavior. Petitioner does not include Mankins for this portion of the limitation, nor would Mankins cure this deficiency. (EX-2001 ¶132.)

Petitioner relies on Mankins in this claim limitation to teach the “mask.” (Petition 32.) To enable this combination, Petitioner argues that it would have “been obvious for a [POSITA] to generate the Bloom filter of normal behavior disclosed in Mankins on the central server of Halperin after receiving information about a software’s system calls from the remote computers as disclosed by Halperin.” (Petition 33.) There are three glaring flaws with this combination: *first*, it is not a

true combination but rather two systems acting independently (and antagonistically); *second*, Halperin does not disclose “receiving information about a software’s system calls from the remote computers.” (Petition 33); *third*, Mankins teaches away from its system being based around a central computer—or, at the least, centralizing Mankins’s Bloom filter data objects on the server would eliminate Petitioner’s purported justification for combining Mankins with Halperin, namely reducing false positives. (EX-2001 ¶133.)

In the first case, running Mankins’s Bloom filters on each remote computer in addition to Halperin’s network-level behavior-detection system is not a combination of the two systems but is instead a “defense-in-depth” strategy involving two distinct pieces of software. (EX-2001 ¶134.) When care is not taken and multiple security strategies are ineffectively combined, the result is not added security but instead redundancy, waste of resources, and frequently an increase in false detections or decreased protection. Here, a POSITA would not have considered running Halperin and then Mankins as two separate malware detection layers to be a useful defense-in-depth combination because the two layers would be unable to share or benefit from each other’s data and might even contradict each other. (EX-2001 ¶¶134-135.)

For example, the proposed combination presents the significant risk that Halperin and Mankins contradict each other. (EX-2001 ¶135.) Halperin teaches that a threshold number of computers must display the same behavior for Halperin to trigger an alert. (EX-1003 [0047]-[0057].) Halperin warns that activity observed on just one computer “cannot be confidently attributed to a virus.” (EX-1003 [0037].) For this reason, a POSITA would understand that Halperin aggregates sensors from different remote computers until the threshold number of computers is reached before suspecting a virus in the network. (EX-2001 ¶135.) When Halperin operates as designed, a single sensor triggering—or even multiple sensors triggering, as long as the number is below the threshold—will not prompt an alert. (*Id.*) A POSITA would not then choose to perform the process of Mankins, which triggers upon a *single* Bloom filter violation, as an extra test in Halperin. (*Id.*) Doing so would contradict Halperin’s teaching that behavior must be evaluated in the aggregate. (*Id.*) A POSITA would not find it obvious to connect two cybersecurity solutions with conflicting methodologies, because it would lead to difficulty in interpreting threat risk in the computer system or network. (*See* Ex-2001 ¶97.) More specifically, if Halperin’s predefined suspicious behavior patterns do not detect viral activity, then it is because Halperin is working as intended and there is no viral outbreak to detect. (*See* Ex-2001 ¶94.) A POSITA would not view this as an opportunity to change

strategies altogether to now move into an object-focused intrusion detection routine as disclosed in Mankins. (*Id.*)

In the second case, Halperin does not mention the words “system calls” or ever refer to an individual computer’s system-call behavior. Petitioner’s expert does not correct this mischaracterization at any point in his declaration. The combination of Halperin and Mankins as defined by Petitioner for this limitation is unsubstantiated. Accordingly, Petitioner does not establish a reasonable likelihood of success in meeting its burden to show that Halperin in view of Mankins discloses “*automatically generating ... acceptable behavior for the computer object.*” (EX-2001 ¶136.)

In the third case, a POSITA would not have combined Halperin and Mankins in a manner requiring the Mankins system to be based on a central server because combining the two in this way would eliminate the very benefit Petitioner suggests would have motivated a POSITA to combine Halperin with Mankins—reducing false positives. (EX-2001 ¶137.) Mankins is not designed to operate in the distributed environment Halperin contemplates. (EX-2001 ¶137.) To the contrary, Mankins teaches away from this central-server arrangement—it discourages operating system-call-based detections outside of the kernel itself. Mankins explains that such arrangements are inefficient:

Unfortunately, several shortcomings are apparent in existing system-call-based intrusion detection techniques. For example, significant delays are inherent in these techniques and may make them impracticable for use with complex computer programs that are processed at high speeds. In particular, it takes time to compare system calls tracked in the OS kernel with data of a profile database stored outside of the OS because communications must be sent back and forth between the OS kernel and the profile database.

EX-1004 2:14-26. Accordingly, a POSITA would not have sacrificed Mankins's efficiency by adapting the Bloom filter data objects to Halperin's central computer 502 as advocated by Petitioners. (EX-2001 ¶137.)

Additionally, when it operates on a network, Mankins teaches that reducing the false positives inherent in Bloom filters *requires* generating Bloom filter data objects at *each* of the multiple remote computers so that the results at those separate computers may be checked against one another:

[W]hen several network devices generate separate Bloom filter data objects 180, are monitored together, and are subject only to attacks that affect each of the network devices within a small amount of time, it is likely that at least one of the network devices would detect a false positive that is produced by another device.

EX-1004 10:57-62. In short, the generation of multiple Bloom filter data objects at each of the remote computers operates as a sanity check for each of the filters—and that sanity check is eliminated if only a single Bloom filter data object is generated at a central computer. Accordingly, contrary to Petitioner's argument that a POSITA would have combined Halperin with Mankins to reduce false positives, a POSITA

would have recognized that combining Halperin with Mankins such that Mankins's Bloom filter data objects are generated at Halperin's central computer would eliminate the means by which Mankins reduces false positives. (EX-2001 ¶¶137.) A POSITA would not have been able to combine Halperin with Mankins in order to reduce false positives. (*Id.*)

Further, Petitioner does not explain *how* Mankins (which monitors individual computer programs) would even operate in conjunction with Halperin (which is program-agnostic). (EX-2001 ¶138.) Petitioner is vague about for which program the system would “automatically generat[e] a mask” because neither reference provides that answer. The only technically feasible solution is to run Mankins *on every piece of software running on every computer in the network*. (EX-2001 ¶88.) Petitioner implausibly contends that this plainly resource-intensive procedure would somehow be “obvious to efficiently implement.” (Petition 34; EX-2001 ¶88.)

5. 1[f] Petitioner fails to show that Halperin in view of Inoue “discloses” “*automatically generating ... acceptable behavior for the computer object*”

Inoue is functionally equivalent to Mankins. Petitioner argues that limitation 1[f] is obvious over Halperin in view of Inoue in much the same way Petitioner argues that the limitation is obvious over Halperin in view of Mankins. Indeed, Dr. Nielson repeats, largely verbatim, the same language to describe the two purported

combinations' operation and purported benefits. (*Compare* EX-1002 ¶¶ 107-113 *with id.* ¶¶ 421-427). For example, Dr. Nielson suggests that each of the two purported combinations offers “defense in depth,” with the secondary (Inoue or Mankins) reference being used as “another alternative to a simple false-alarm/less-detection tradeoff” and operating “as an active quarantine system.” (*Id.* ¶¶ 108, 111, 112, 422, 425, 426.)

That repetition is no surprise, because Mankins and Inoue in fact perform largely similar functions in largely similar fashions. (EX-2001 ¶141.) Each system heuristically trains a model (a Mankins Bloom filter or an Inoue dynamic sandbox) with a program's expected behavior. (EX-1004 7:8-11; EX-1005-31; EX-2001 ¶141.) Each system then monitors that individual program's behavior by using the trained model to distinguish novel (and thus bad) behavior from previously observed (and thus good) behavior. (*Compare* EX-1004 12:1-14 *with* EX-1005-3; EX-2001 ¶141.) The only real difference in Inoue's operation over that of Mankins is that Inoue runs in a Java virtual machine instead of the OS kernel, and thus uses *method invocations* instead of *system calls* as its unit of examination. (*See* EX-1005-3; EX-2001 ¶142.)

Halperin in view of Inoue does not disclose 1[f] for the same reasons that Halperin in view of Mankins does not disclose 1[f]. Thus, for the same reasons as

described above (Section V.B.4), Petitioner's challenge based on Halperin and Inoue fails. Like Mankins, Inoue teaches a heuristically trained model that disambiguates allowable system calls from those that are suspicious. (EX-2001 ¶143.) Thus, because of the fundamental similarities between the Inoue system and the Mankins system, the Halperin-Inoue combination fails for the same reasons the Halperin-Mankins system does: *First*, Halperin teaches away from heuristic behavior detection strategies like Inoue and Mankins (EX-2001 ¶143); *second*, as disjointed layers of a defense-in-depth strategy, running Halperin and Inoue offers neither system any benefit and would instead result in conflicting malware-detection reports (EX-2001 ¶144); and, *third*, the fundamental dissimilarity between the operation, purpose, and focus of Halperin from those of Inoue mean that a POSITA would not understand how Inoue would even operate with Halperin (EX-2001 ¶¶145-146). (See *supra* Section V.B.4.)

C. Patent Owner Reserves Its Right To Contest Portions Of The Petition Not Addressed.

At this preliminary stage, Patent Owner addresses only those deficiencies in the prior art necessary to clearly show a lack of reasonable likelihood of the petition's success. Patent Owner's decision not to address certain limitations or secondary prior art references at this preliminary stage in no way reflects Patent

Owner's agreement with any unaddressed contention in the petition or lack of dispute as to additional limitations or prior art.

D. Objective Indicia Of Nonobviousness Further Support The Challenged Claims' Patentability.

Objective indicia of nonobviousness exist, and Patent Owner reserves its right to provide evidence on this point, should the Board institute. (EX-2001 ¶33.) (But for the foregoing reasons, the Board should not.)

VI. CONCLUSION

For the foregoing reasons, the Board should deny institution.

Date: July 17, 2023

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 PRELIMINARY RESPONSE** specified in 37 C.F.R. § 42.24 have 13,501 words, in compliance with the 14,000 word limit set forth in 37 C.F.R. § 42.24. This word count was prepared using Microsoft Word 365 v.2022.

Date: July 17, 2023

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 PRELIMINARY RESPONSE** and all supporting **EXHIBITS 2001-2014** were served by filing the documents through the Patent Trial and Appeal Case Tracking System (P-TACTS) on July 17, 2023 and courtesy copies were also served via electronic mail, in their entireties on the following:

Robert M. Hansen (Reg. No. 43,656)
Scott B. Amankwatia (Reg. No. 78,226)
Matthew J. Anderson (Reg. No. 78,257)
THE MARBURY LAW GROUP, PLLC
11800 Sunrise Valley Drive
15th Floor
Reston, VA 20191
Tel: (703)-391-2900
Fax: (703)-391-2900
Email: rhansen@marburylaw.com
Email: samankwatia@marburylaw.com
Email: manderson@marburylaw.com
Email: pat-docketing@marburylaw.com

*Counsel for Petitioner
Trend Micro Inc.*

Date: July 17, 2023

KING & SPALDING LLP

/Brian Eutermoser/

Brian Eutermoser
Registration No. 64,058

Attorney for Patent Owner
Open Text Inc.