

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TREND MICRO, INC.
Petitioner

v.

OPEN TEXT INC.
Patent Owner

Case IPR2022-00692
Patent No. 10,284,591

PATENT OWNER'S PRELIMINARY RESPONSE TO PETITION

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

TABLE OF CONTENTS.....	i
EXHIBIT LIST	viii
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 Before The Projected Statutory Deadline.....	11
C. Factor 3: The Parties Have Invested Significant Resources In The District Court Proceeding.....	13
D. Factor 4: The Issues And Parties Overlap Significantly.....	15
E. Factor 5: Both Parties Are Parties In The Parallel Litigation.	17
F. Factor 6: The Petition Does Not Present “Compelling Evidence Of Unpatentability.”	19
III. BACKGROUND FUNDAMENTALS	20
A. Programming Languages.....	20
B. Function Calls.....	20
C. Call Stacks	21
IV. INADEQUACY OF THE PRIOR ART & THE NEED FOR IMPROVED ANTI-EXPLOIT TECHNIQUES.....	24
V. THE ’591 PATENT’S NEW ANTI-EXPLOIT TECHNOLOGY	26
A. Overview of the ’591 Patent.....	26

B.	Claim Construction	28
C.	Level of Ordinary Skill in the Art	28
VI.	OVERVIEW OF THE PRIOR ART	29
A.	Overview of Dalcher	29
B.	Overview of Pappas.....	33
VII.	PETITIONER FAILS TO SHOW A REASONABLE LIKELIHOOD THAT IT WOULD PREVAIL WITH RESPECT TO ANY CHALLENGED CLAIM.....	36
A.	Petitioner Does Not Adequately Articulate a Combination of Dalcher’s and Pappas’s Teachings	38
B.	There Is No Motivation to Combine the Teachings of Dalcher and Pappas	46
1.	Petitioner fails to show why a POSITA would have been motivated to combine Dalcher with Pappas to practice the claimed invention.....	46
2.	Petitioner fails to establish a reasonable likelihood of success.....	48
3.	A POSITA would not have been motivated to combine the teachings of Dalcher and Pappas.	51
C.	Petitioner Fails to Show That the Combination of Dalcher and Pappas Renders Obvious Any Independent Claim	55
1.	Petitioner fails to show that the combination of Dalcher and Pappas renders obvious performing a memory check before an address of an originating caller function is reached (1.e, 9.f, 17.e).....	56
2.	Petitioner fails to show that the combination of Dalcher and Pappas renders obvious triggering an alert when the	

memory check detects one of the enumerated elements (1.e, 9.f, 17.e).....	60
3. Petitioner fails to show that the combination of Dalcher and Pappas renders obvious preventing execution of a payload (1.f, 9.g, 17.f).....	63
D. Petitioner Fails to Show a Reasonable Likelihood of Success as to Any of Challenged Dependent Claims.....	66
VIII. OTHER LIMITATIONS NOT ADDRESSED	66
IX. THERE ARE SECONDARY CONSIDERATIONS OF NON- OBVIOUSNESS.....	67
X. CONCLUSION.....	68
CERTIFICATION OF WORD COUNT	1
CERTIFICATE OF SERVICE	2

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>ActiveVideo Networks, Inc. v. Verizon Comms., Inc.</i> , 694 F.3d 1312 (Fed. Cir. 2012)	<i>passim</i>
<i>ADT LLC v. Vivint, Inc.</i> , IPR2022-00642, Paper 7 (PTAB Oct. 4, 2022)	40, 45, 46, 55
<i>Apple Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020)	<i>passim</i>
<i>Apple Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 15 (PTAB May 13, 2020) (<i>informative</i>)	11
<i>Belden Inc. v. Berk–Tek LLC</i> , 805 F.3d 1064 (Fed. Cir. 2015)	46
<i>Cisco Sys., Inc. v. Monarch Networking Solutions LLC</i> , IPR2020-01227, Paper 11 (PTAB Mar. 4, 2021)	14
<i>Commscope Techs. v. Dali Wireless, Inc.</i> , IPR 2022-01242 (PTAB Feb. 27, 2023)	19, 20, 21, 22
<i>Duo Security Inc. v. StrikeForce Techs., Inc.</i> , No. IPR2017-01064, 2017 WL 4677238 (PTAB Oct. 16, 2017)	46, 59, 62, 66
<i>Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.</i> , 800 F.3d 1375 (Fed. Cir. 2015)	37
<i>Eclinical Works, LLC v. Decapolis</i> LLC, IPR2022-00229, Paper 10 (PTAB Apr. 13, 2022)	13
<i>Ericsson Inc. v. Collision Comms., Inc.</i> , IPR2022-01233, Paper 12 (PTAB Jan. 19, 2023)	15
<i>Google LLC v. EcoFactor, Inc.</i> , IPR2021-00454, Paper 9 (PTAB July 22, 2021)	17

Patent Owner’s Preliminary Response to
Petition for Inter Partes Review
Patent No. 10,284,591

<i>Harmonic Inc. v. Avid Tech., Inc.</i> , 815 F.3d 1356 (Fed. Cir. 2016)	37
<i>In re Huang</i> , 100 F.3d 135 (Fed. Cir. 1996)	67
<i>InfoBionic Inc. v. Braemer Mfg., LLC</i> , IPR2015-01705, Paper 11 (PTAB Feb. 16, 2016).....	37
<i>Intellectual Ventures I LLC v. Hewlett Packard Enterprise Co.</i> , No. 6:21-cv-00596-ADA, slip op. (W.D. Tex. Dec. 22, 2022)	7
<i>In re: Keller</i> , 642 F.2d 413 (CCPA 1981)	38
<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	44, 48
<i>Leo Pharm. Prods., Ltd. v. Rea</i> , 726 F.3d 1346 (Fed. Cir. 2013)	67
<i>Liberty Mutual Ins. Co. v. Progressive Cas. Ins. Co.</i> , CBM2012-00003 Paper 8 (PTAB Oct. 25, 2012)	45
<i>Lowe’s Home Centers, LLC v. Epistar Corp.</i> , IPR2018-00932, Paper 7 (PTAB Nov. 1, 2018).....	38
<i>Lumenis Be Ltd. v. BTL Healthcare Techs. A.S.</i> , IPR2021-01280, Paper 8 (PTAB Mar. 9, 2022)	37
<i>Luxshare Precision Industry v. Amphenol Corp.</i> , IPR2022-00132, Paper 10 (PTAB May 3, 2022)	10
<i>Lyft, Inc. v. Quartz Auto Techs., LLC</i> , IPR2020-01450, Paper 7 (Mar. 4, 2021)	39, 40, 45
<i>mCom IP, LLC v. Cisco Sys., Inc.</i> , No. 6:22-00261-ADA, slip op. (W.D. Tex. Oct. 20, 2022).....	7
<i>Microsoft Corp. v. FG SRC, LLC</i> , 860 Fed. Appx. 708 (Fed. Cir. 2021).....	44

Patent Owner’s Preliminary Response to
Petition for Inter Partes Review
Patent No. 10,284,591

<i>Motorola Mobility LLC v. Ironworks Patents, LLC</i> , IPR2021-00421, Paper 11 (PTAB July 22, 2021)	11
<i>NHK Spring Co., Ltd. v. Intri-Plex Techs., Inc.</i> , IPR2018-00752, Paper 8 (PTAB Sept. 12, 2018) (<i>precedential</i>).....	12
<i>Nintendo Co. v. Gamevice, Inc.</i> , IPR2020-01197, Paper 13 (PTAB Jan. 12, 2021)	10
<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. 24, 2022)	14
<i>Palo Alto Networks, Inc. v. Centripetal Networks, Inc.</i> , IPR2021-01153, Paper 10 (PTAB Jan. 24, 2022)	45
<i>Pers. Web Techs., LLC v. Apple, Inc.</i> , 848 F.3d 987 (Fed. Cir. 2017)	<i>passim</i>
<i>Regeneron Pharms. Inc. v. Novartis Pharma AG</i> , IPR2020-01317, Paper 15 (PTAB Jan. 15, 2021)	14, 15
<i>Samsung Elecs. Co. v. California Institute of Tech.</i> , IPR2023-00131, Paper 10 (PTAB May 4, 2023)	12
<i>Samsung Elecs. Co. v. Clear Imaging Research LLC</i> , IPR2020-01399, Paper 13 (PTAB Feb. 3, 2021).....	10, 11
<i>Securus Techs., Inc. v. Global Tel*Link Corp.</i> , IPR2015-00155, Paper 30 (PTAB Apr. 7, 2016)	41, 59, 62, 66
<i>Sonrai Memory Ltd. v. Kingston Tech. Co.</i> , No. 6:21-cv-1284-ADA, slip op. (W.D. Tex. Oct. 18, 2022).....	7
<i>Sotera Wireless, Inc. v. Masimo Corp.</i> , IPR2020-01019, Paper 12 (PTAB Dec 1, 2020)	17
<i>T-Mobile US, Inc. v. TracBeam LLC</i> , IPR2015-01687, Paper 10 (Feb. 12, 2016)	45

Patent Owner’s Preliminary Response to
Petition for Inter Partes Review
Patent No. 10,284,591

<i>TikTok Inc. v. Advanced Coding Techs. LLC</i> , IPR2022-01546, Paper 9 (PTAB June 15, 2023)	39
<i>Twinstrand Biosciences v. Guardant Health, Inc.</i> , IPR2022-01115, Paper 14 (PTAB Dec. 20, 2022)	66
<i>Vizio, Inc. v. Maxell, Ltd.</i> , IPR2022-01459, Paper 7 (PTAB Apr. 11, 2023)	17
<i>WBIP, LLC v. Kohler Co.</i> , 829 F.3d 1317 (Fed. Cir. 2016)	51
Statutes	
35 U.S.C. § 314(a)	3, 19, 36
Other Authorities	
37 C.F.R. § 42.22(a)(2)	37

EXHIBIT LIST

Exhibit	Description
2001	Declaration of Professor Nenad Medvidovic, Ph.D in Support of Patent Owner’s Preliminary Response To Petition
2002	<i>Curriculum Vitae</i> of Professor Nenad Medvidovic, Ph.D.
2003	<i>Webroot, Inc. and Open Text Inc., v. Trend Micro Inc.</i> , Case no. 6:22-cv-00239, Complaint for Patent Infringement, Dkt. 1 (W.D. Tex., March 4, 2022)
2004	<i>Sonrai Memory Ltd. v. Kingston Tech. Co. and Kingston Tech. Corp.</i> , Case No. 6:21-cv-1284-ADA, Order Denying Kingston Tech. Co. and Kingston Tech. Corp.’s Opposed Motion to Stay Pending Resolution of <i>Inter Partes</i> Review, Dkt. 94 (W.D. Tex., Oct. 18, 2022)
2005	<i>mCom IP, LLC v. Cisco Systems, Inc.</i> , Case No. 6:22-cv-00261-ADA, Order Denying Defendant’s Motion to Stay Pending <i>Inter Partes</i> Review, Dkt. 42 (W.D. Tex., Oct. 20, 2022)
2006	<i>Intellectual Ventures I LLC et al. v. Hewlett Packard Enterprise Co.</i> , No. 6:21-cv-00596-ADA, Order Denying Hewlett Packard Enterprise Company’s Opposed Motion to Stay Pending Resolution of <i>Inter Partes</i> Review, Dkt. 104 (W.D. Tex., Dec. 22, 2022)
2007	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab. et al.</i> , Case No. 6:22-cv-243-ADA-DTG, Dec. 9, 2022 Transcript of Motions Hearing, Dkt. 133 (W.D. Tex. Dec. 15, 2022) (excerpted)
2008	Dani Kass, “Catching Up On Patent Litigation With Judge Al-bright” (Law360 Mar. 14, 2023); https://www.law360.com/pulse/articles/1582438/print?section=pulse/courts

Patent Owner’s Preliminary Response to
Petition for Inter Partes Review
Patent No. 10,284,591

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, Order Granting Fourth Amended Scheduling Order, Dkt. 160 (W.D. Tex. Jan. 22, 2023)
2010	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)”, available https://www.uscourts.gov/sites/default/files/data_tables/fcms_na_distprofile0331.2023.pdf
2011	<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)
2012	<i>Webroot, Inc. and Open Text Inc., v. Sophos Ltd.</i> , Case No. 6:22-cv-240-ADA-DTG, Defendant’s Preliminary Invalidity Contentions (W.D. Tex., Sept. 13, 2022)
2013	<i>Webroot, Inc. and Open Text Inc., v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.</i> , Case No. 6:22-cv-00241-ADA-DTG, Second Supplemental Preliminary Invalidity Contentions (W.D. Tex., Jul. 14, 2023)
2014	U.S. Pat. Appl. Pub. No. 2009/0049550 A1 to S. Shevchenko, filed Jun. 6, 2008 and issued Feb. 19, 2009 (“Shevchenko”)
2015	U.S. Pat. Appl. Pub. No. 2005/0108562 A1 to R. Khazan et al., filed Jun. 18, 2003 and issued May 19, 2005 (“Khazan”)
2016	U.S. Patent 8,510,596 B1 to S. Gupta and P. Shenoy, filed Jul. 6, 2007 and issued Aug. 13, 2013 (“Gupta”)
2017	<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)

Patent Owner's Preliminary Response to
Petition for Inter Partes Review
Patent No. 10,284,591

Exhibit	Description
2018	Erickson, J., Hacking: The Art of Exploitation, 2nd Ed. (2008)
2019	Kleymenov, A. and Thabet, A., Mastering Malware Analysis, Packt Pub. (2019)
2020	LWN.net, An Introduction to Last Branch Records (Mar. 23, 2016) (available at https://web.archive.org/web/20160413055733/https://lwn.net/Articles/680985/)
2021	U.S. Patent 9,251,373 to K. AlHarbi and X. Lin, filed Mar. 14, 2013 and issued Feb. 2, 2016
2022	U.S. Patent 8,291,381 to T. Lai et al., filed Sep. 27, 2007 and issued Oct. 16, 2012
2023	GeeksforGeeks, Stack Unwinding in C++
2024	IBM Documentation, Stack unwinding (C++ only)
2025	Microsoft Learn, Exceptions and Stack Unwinding in C++

Open Text Inc. ("Patent Owner") submits this Preliminary Response to the Petition of Trend Micro, Inc. ("Petitioner") seeking *inter partes* review ("IPR") of U.S. Patent No. 10,284,591 ("the '591 patent").

I. INTRODUCTION

The Board should decline to institute IPR of the '591 patent because an IPR would duplicate the parties' district court litigation efforts and because Petitioner's solitary Section 103 ground fails to adequately identify a combination of the prior art teachings.

First, the Board should deny institution under *Fintiv* because instituting this IPR would undermine Congress's stated goals in creating IPR proceedings: to provide "quick and cost effective alternatives to litigations" and "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. An IPR here would be neither quicker than, nor an alternative to, ongoing district court litigation. The Western District of Texas ("WDTX") is scheduled to proceed to trial on the '591 patent almost three months ***before*** a final written decision on the petition would be due—so an IPR is not quicker. And, the district court has indicated it will not issue a stay when, as here, it can resolve the same issues prior to the Board—so an IPR is not an alternative to litigation. Moreover, other defendants have asserted the same prior art raised in the

petition, which the district court must address regardless of whether the Board institutes an IPR. Therefore, instituting an IPR would generate the unnecessary and counterproductive litigation costs that Congress sought to avoid.

Second, the Board should deny institution because Petitioner failed to demonstrate a reasonable likelihood that it will prevail on the merits. The Federal Circuit has admonished that “[a] clear, evidence-supported account ***of the contemplated workings of the combination is a prerequisite*** to adequately explaining and supporting a conclusion that a relevant skilled artisan would have been motivated to make the combination and reasonably expect success in doing so.” *Pers. Web Techs., LLC v. Apple, Inc.*, 848 F.3d 987, 994 (Fed. Cir. 2017). While evaluating Dalcher and Pappas individually, nowhere in its petition does Petitioner explain how a POSITA would allegedly combine Dalcher’s and Pappas’s teachings or what such a combination would suggest to a POSITA. Because neither Dalcher nor Pappas individually disclose the claimed invention, the deficiencies in Petitioner’s solitary Section 103 ground are fatal and the petition is not reasonably likely to succeed.

Patent Owner respectfully requests that the Board deny institution.

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

The '591 patent is one of dozens of patents asserted in claims and counterclaims among six different parties in a series of cases pending in the Western District of Texas. The district court has already issued its claim constructions and is on track to adjudicate the same prior art raised in the petition months before the Board issues a final written decision. Institution of a parallel IPR proceeding lagging well behind the district court would needlessly increase the cost of resolving the underlying dispute. Under these circumstances, the Board should exercise its discretionary authority to deny institution 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." 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. A district court stay is out of the question, and the district court will consider the very same prior art 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 sought 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 Mar. 20, 2020).

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 review of only five of the eleven patents asserted against it. And this IPR will similarly not narrow the issues on which discovery and trial must proceed—the '591 patent is currently asserted against four defendants in a now-consolidated case. Moreover, none of Petitioner's codefendants have agreed not to litigate the validity of the '591 patent in the district court if the Board institutes this IPR.

(2) Trial in the district court is expected almost three months ***before*** a final written decision. The district court case remains on track. The claims of the '591 patent have been construed, and discovery is well underway.

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

(4) The prior art references raised in the petition are the same as those the district court has before it, and Petitioner has not offered *any* stipulation that it will not simultaneously pursue in the district court *precisely the same arguments* based on *precisely the same prior art* asserted in its petition during the pendency of any IPR instituted from that petition, irrespective of the Board's institution decision.

(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 [1.] the court granted a stay or [2.] evidence exists that one may be granted if a proceeding is instituted.” *Fintiv*, IPR2020-00019, Paper 11 at 6.¹ Here, the undisputed 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 unlikely.

¹ The decision in *Fintiv* addresses various scenarios that may indicate whether a stay may be granted. *Fintiv*, IPR2020-00019, Paper 11 at 6-9. However, nothing in *Fintiv* suggests that these are the only scenarios that a panel may consider. In particular, *Fintiv* does not hold that an actual denial of a stay is the only consideration relevant to this factor. Rather, *Fintiv* contemplates that any type of evidence may be relevant. For example, *Fintiv* holds that one way to “support the arguments that ... a stay may be less likely, and instituting would lead to duplicative costs” is to show the “work completed by the parties and court in the parallel proceeding.” *Fintiv*, IPR2020-00019, Paper 11 at 10.

First, the district court considers issue simplification “the most important factor” in deciding whether to stay patent litigation²—and here, a stay would not simplify anything. The validity of the ’591 patent will be litigated in the consolidated district court action, whether this petition is instituted or not, because the patent is asserted against three other defendants, none of whom have promised not to pursue invalidity in the district court based on this institution decision.

²*See, e.g.*, EX-2004, *Sonrai Memory Ltd. v. Kingston Tech. Co.*, 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-2005, *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-2006, *Intellectual Ventures I LLC v. Hewlett Packard Enterprise Co.*, No. 6:21-cv-00596-ADA, slip op. at *7-8 (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).

Moreover, the validity of the '591 patent will be litigated in the district court alongside issues regarding 38 *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 twenty patents—thirteen asserted by Patent Owner and seven 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
10,284,591	IPR2023-00895	IPR2022-01522	N/A	Asserted; no IPR	IPR2023-00692
9,578,045	IPR2023-01199	IPR2023-00124	N/A	Asserted; no IPR	IPR2023-00677
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
8,418,250	Asserted; no IPR	IPR2023-00289	N/A	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

Patent Owner's Preliminary Response to
Petition for Inter Partes Review
Patent No. 10,284,591

Patent	Kaspersky	CrowdStrike	Forcepoint	Sophos	Trend Micro
9,413,721	N/A	N/A	Asserted; no IPR	IPR2023- 00491	N/A
10,257,224	IPR2023- 01011	IPR2023- 00126	N/A	Asserted; no IPR	IPR2023- 00657
8,856,505	Asserted; no IPR	IPR2023- 01159	N/A	Asserted; no IPR	Asserted; no IPR
8,181,244	N/A	IPR2023- 01158	N/A	Asserted; no IPR	Asserted; no IPR
11,409,869	IPR2023- 01160	IPR2023- 01053	Asserted; no IPR	PGR2023- 00031	Asserted; no IPR
8,763,123	N/A	IPR2023- 1050	N/A	Asserted; no IPR	N/A
8,438,386	N/A	N/A	IPR2023- 00784	N/A	N/A
10,025,928	N/A	N/A	IPR2023- 00786	N/A	N/A

As U.S. Magistrate Judge Gilliland noted during a December 9, 2022 hearing, this is “a phenomenal case that I’m hoping one day maybe people will write songs about because it is about as complicated as you’re going to get with patent cases.” (EX-2007 52.)

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-2008 3.) Indeed, as commentators have observed, “Judge Albright almost never stays proceedings to let the PTAB make its decision first.” (*Id.* 2.)

Judge Albright's public comments apply squarely to the situation here. The median time-to-trial in WDTX is 28.2 months. (EX-2010 37.) On that timeline, trial is expected in this case in mid-July 2024. Trial is currently scheduled for August 2024. (EX-2009 6.) The deadline for a final written decision in an IPR here would not be until many months later (in November 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 that he would not stay the WDTX case in response to institution of this IPR.

Petitioner acknowledges that no stay has been entered in the parallel litigation. (Petition 64.) 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).

In sum, the district court has not granted a stay and there is no evidence suggesting that the district court would grant a stay of a multi-patent, multi-defendant case so that the Board can reach a decision months *after* the current trial date on *just one* patent. Factor 1 weights in favor of 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 Before The Projected Statutory Deadline.

Factor 2—the implications of the district court and IPR schedules—also weighs for 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*); Guidance at 1.

Here, an IPR would overlap almost entirely with the parallel district court litigation. WDTX’s median time-to-trial is 28.2 months (EX-2010 37), indicating a trial by mid-July 2024—a date consistent with the currently scheduled trial date of August 19, 2024. (EX-2009 6.) Should the Board institute, its final written decision deadline would be November 13, 2024, nearly three months *after* the scheduled trial

date. Petitioner asserts that “trial is currently set for August 19, 2024, so there is time to complete inter parte [sic] review” (Petition 64) but that is incorrect; applying *either* the trial date set by the Court or WDTX’s median time-to-trial, an *inter partes* review would not be complete before the ’591 patent’s validity is tried against the same prior art asserted in the district court.

In other words, if the Board institutes an IPR here, the IPR proceedings would not produce a result before the trial, and the parties would spend a year expending precisely the “duplicative,” “unnecessary[,] and counterproductive litigation costs” that the AIA was designed to avoid. *Fintiv*, IPR2020-00019, Paper 11 at 5 (citing *Magellan Midstream Partners L.P v. Sunoco Partners Mktg. & Terms. L.P.*, IPR2019-01445, Paper 12 at 10 (Jan. 22, 2020)); *see id.*, Paper 15 at 17 (discretionarily denying institution where district court trial was scheduled two months before the final decision deadline); *NHK Spring Co., Ltd. v. Intri-Plex Techs., Inc.*, IPR2018-00752, Paper 8 at 20 (PTAB Sept. 12, 2018) (*precedential*) (finding that institution of IPR when the district court trial would resolve validity before the PTAB “would not be consistent with an objective of the AIA...to provide an effective and efficient alternative to district court litigation”); *Samsung Elecs. Co. v. California Institute of Tech.*, IPR2023-00131, Paper 10 at 15, 24 (PTAB May 4, 2023) (discretionarily denying institution where “[r]egardless of whether we

consider the scheduled trial date or the trial date expected based on time-to-trial statistics, the trial in the underlying litigation would commence several months before the expected date of our final written decision”); *Eclinical Works, LLC v. Decapolis LLC*, IPR2022-00229, Paper 10 at 9 (PTAB Apr. 13, 2022) (discretionarily denying institution where WDTX trial was scheduled “roughly one to two months before any final decision would have been due”).

Because the district court will resolve validity before the Board reaches its final written decision, the second *Fintiv* factor also favors discretionary denial.

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

Factor 3—the resources invested in the parallel district court proceeding—favors denial “if, at the time of the institution decision, the district court has issued substantive orders related to the patent.” *Fintiv* at 9-10; *see also id.* at n.16 (collecting cases). *Fintiv* holds specifically that “claim construction orders may indicate that the court and parties have invested sufficient time in the parallel proceeding to favor denial.” *Id.* at 10.

The parties have invested significant resources in the parallel litigation over the last eighteen 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-2017.)

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. *Fintiv* at 6; *NXP*, PGR2022-00005, Paper 20 at 7 (Board should consider “the duplicative efforts that will occur during the [] duration of a [inter partes] review”); *Cisco Sys., Inc. v. Monarch Networking Solutions LLC*, IPR2020-01227, Paper 11 at 11-12 (PTAB Mar. 4, 2021) (finding that more work completed by the parties and the court suggests that institution would lead to duplicative costs).

The investment in this case is colossal. The district court's *Markman* order that included the '591 patent was twenty pages, without even considering the Court's substantive analysis or the two additional *Markman* orders issued in this case. (EX-2017.) The Court considered and construed seven claim terms of the '591 patent, alone. (*Id.* 12-17.) And, the parties have engaged in extensive discovery on the parties' contentions, during which Patent Owner has made approximately 50TB of source code available. The Court has held multiple discovery hearings. Furthermore, Petitioner's and Patent Owner's experts have already invested an extensive number of hours preparing for their reports by reviewing the patent, prior art, and Defendants' invalidity contentions, as well as traveling to review source code for more than two months. See *Regeneron Pharms. Inc. v. Novartis Pharma AG*,

IPR2020-01317, Paper 15 at 17 (PTAB Jan. 15, 2021) (denying institution where the parties had expended considerable resources in claim construction, substantial fact discovery, and preparation for expert reports).

Petitioner's delay in filing this petition further weighs for denial. Petitioner waited almost exactly twelve months after receiving Patent Owner's March 4, 2022 complaint, eight months after receiving Patent Owner's infringement contentions, and six months after serving its invalidity contentions before filing its petition. Petitioner offers no explanation for its delay, which weighs for denial. *Fintiv*, IPR2020-00019, Paper 11 at 11-12; *Ericsson Inc. v. Collision Comms., Inc.*, IPR2022-01233, Paper 12 at 15 (PTAB Jan. 19, 2023) (finding that where the parties invested significant resources and "Petitioner waited almost a year" to file its petition, "this factor weighs decidedly in favor of discretionary denial"); *Regeneron*, IPR2020-01317, Paper 15 at 17 (finding that even where the petition was filed "less than a month after learning of the complaints," the investment by the parties in the parallel proceeding "outweighs the effort expended so far in this proceeding").

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, Petitioner 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. Even if Petitioner had offered a stipulation, such a stipulation would carry no weight because it does not bar the other defendants in the parallel litigation from litigating the very same invalidity challenges before the district court. Petitioner's co-defendants rely in the litigation on the same prior art asserted in the petition.³ In light of the inevitable parallel tracks, instituting an IPR would not only multiply the burden on Patent Owner, it would also create the possibility of inconsistent decisions between the district court and the Board.

³ Both Petitioner and codefendant Sophos assert Dalcher. (EX-2011 17; EX-2012 17.) Codefendant CrowdStrike also asserts Pappas. (EX-2013 21.)

Because Petitioner makes no effort to “mitigate[] any concerns of duplicative efforts between the district court and the Board,” or “of potentially conflicting decisions,” Factor 4 weighs heavily in favor of discretionary denial. *Sotera Wireless, Inc. v. Masimo Corp.*, IPR2020-01019, Paper 12 at 19 (PTAB Dec. 1, 2020); *Google LLC v. EcoFactor, Inc.*, IPR2021-00454, Paper 9 at 12-14 (PTAB July 22, 2021).

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 prior art references 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. That is not the case here because Petitioner 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 prior art at the district court further suggests denial. *Fintiv* holds that, “[e]ven when a petitioner is unrelated to a defendant, ... 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 “[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 prior art 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 prior art 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 weighs against institution because, as detailed below, the petition fails to show even a “reasonable likelihood” that Petitioner would prevail on any challenged claim. *See* 35 U.S.C. § 314(a). The petition fails to articulate the combination with adequate clarity. For example, Petitioner fails to explain how a POSITA would have combined the teachings of Dalcher and Pappas or why a POSITA would have combined particular teachings from these references. Section VII, *infra*. 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 have a “reasonable likelihood” of succeeding (it does not), it certainly would not pass 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 “*highly likely*” to succeed on the merits. *Id.* at 4. The opposite is true here.

*

*

*

In sum, all six factors weigh for discretionary denial. Accordingly, Patent Owner respectfully requests that the Board exercise its discretion to deny institution.

III. BACKGROUND FUNDAMENTALS

A. Programming Languages

“A program is nothing more than a series of statements written in a specific language.” (EX-2018 6.) A computer understands “machine language,” which is difficult to work with. (*Id.* 7; EX-2001 ¶28.) Therefore, most programmers use high-level languages that are more intuitive and can be compiled into machine language. (EX-2018 7, 36-37; EX-2001 ¶28.) “A low level operation or instruction provides minimal abstraction from a computer’s instruction set architecture, for example either machine code or assembly language. The word ‘low’ refers to the small or nonexistent amount of abstraction between the programming language and machine language.” (EX-1001 3:47-52; EX-2001 ¶29.)

B. Function Calls

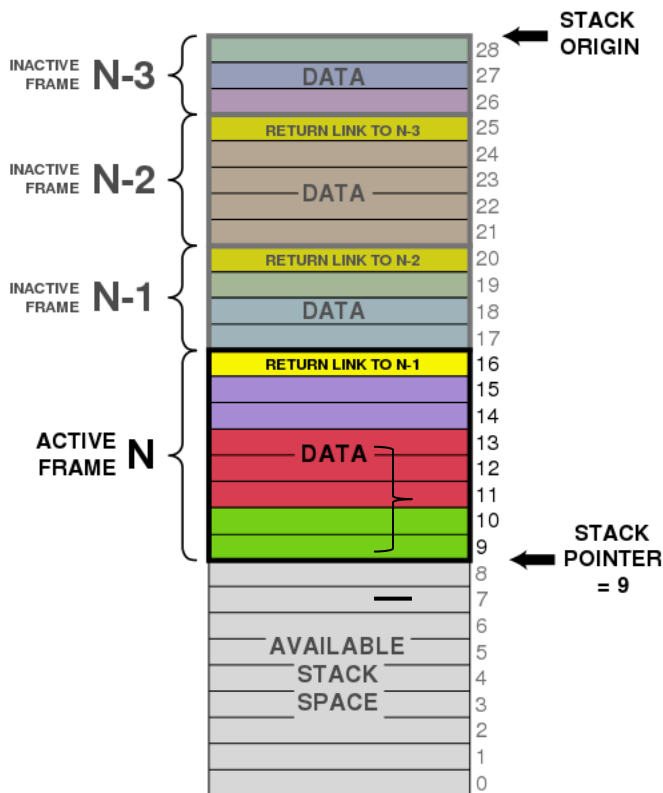
When a programmer knows that a set of program instructions will be used multiple times, she may group instructions into a smaller sub-program or subroutine, called a “function.” (EX-2001 ¶¶30-31.) The function may include sub-functions. (*Id.*) Once a function is established, a program can call the function instead of repeating the instructions. (*Id.* 31.) After calling the function, the instructions within

the function may be executed, after which execution returns to where the program was before it called the function. (*Id.*)

C. Call Stacks

A compiled program's memory (i.e., process memory) is often divided into five segments, each of which has a certain purpose. (EX-2001 ¶32.) The "text" segment stores the assembled machine language instructions of the program. (*Id.*) Because these instructions are not executed linearly, they include "branch, jump, and call instructions" to move the current instruction pointer to a different memory address. (*Id.*)

The "stack" segment of the process memory is used as a temporary scratch pad. (EX-2001 ¶33.) Because functions can contain variables and sub-functions, the stack may be used to remember information including the address at which execution should begin again after the function completes execution. (EX-2018 70.) Individual functions may be contained within their own stack frames (or call frames), and a stack may contain many stack frames. (EX-2001 ¶33.) Below is an illustration that represents an exemplary call stack with call frames N through N-3:



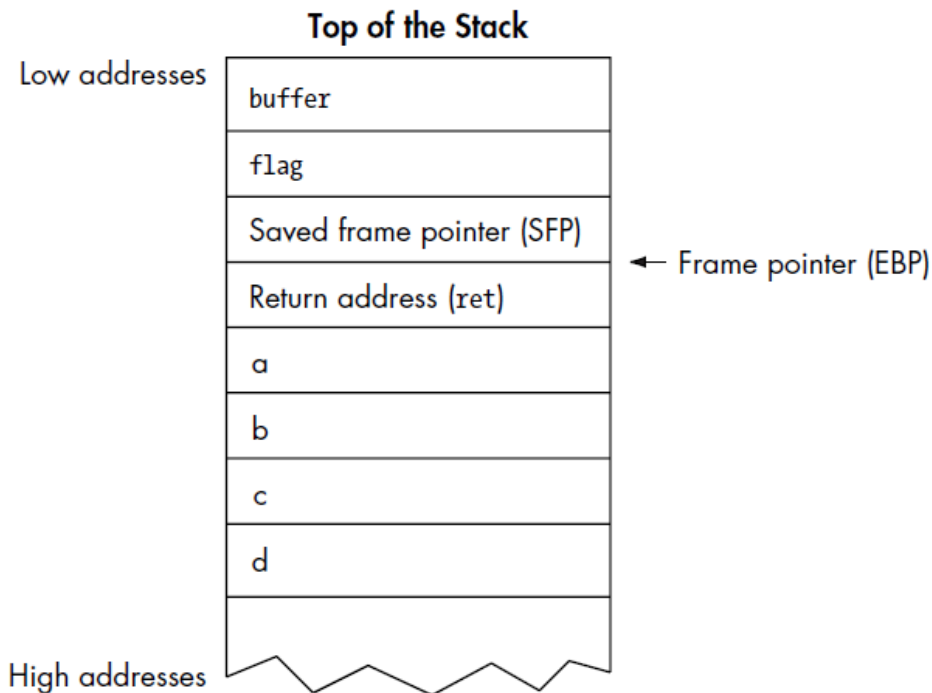
(Id.)

“When a function is **called**, several things are pushed [i.e., added] to the stack together in a stack frame.” (EX-2018 70 (emphasis added)⁴.) Some call stacks, e.g., the one illustrated above, grow downward towards lower-numbered memory addresses as stack frames are added. (EX-2018 70; EX-2021 1:44-45.) For example, Function N-3 is first pushed to the call stack in its own stack frame, and then Function N-2 (and its associated information) is pushed to the call stack in its own

⁴ Unless otherwise noted, all emphasis in quoted material has been added.

stack frame, and so on. (EX-2001 ¶34.) But call stacks may be oriented such that they grow upwards or downwards because a “stack” is merely an abstract data structure for tracking functions as they are called. (EX-2001 ¶35 (citing EX-2022 6:28-35; EX-2018 70).)

For example, the below call stack grows upwards, towards lower addresses:



(EX-2018 73.) Thus, any discussion of moving “up” or “down” a stack or whether a stack is growing towards lower or higher addresses must be considered within the context of a particular call stack. (EX-2001 ¶36.)

The stack follows a last-in, first-out ordering, which means the last item that is put into a stack is the first item to come out of it. (EX-2001 ¶34.) Stack unwinding is generally understood in the art to refer to the removal of function entries from a

call stack during run time. (EX-2001 ¶37.) As execution of each function is completed, the stack frame for that function is “popped off” (i.e., removed from) the stack, and execution returns to the previous calling function. (EX-2018 70, 75.)

IV. INADEQUACY OF THE PRIOR ART & THE NEED FOR IMPROVED ANTI-EXPLOIT TECHNIQUES

“Exploiting a program is simply a clever way of getting the computer to do what you want it to do, even if the currently running program was designed to prevent that action.” (EX-2018 115.) Thus, an “exploit” refers to a piece of code or sequence of commands that takes advantage of a vulnerability to cause unintended, potentially harmful behavior on a computer. (EX-1001 1:12-16, 1:21-23; EX-2001 ¶¶40-42.) One way exploit code may be executed (and thereby cause potentially harmful behavior) is for the program’s normal code flow to be hijacked and directed towards a zone of memory where the exploit code is currently stored. (*Id.* 1:17-21.)

For the exploit to accomplish its goal of starting a new process, the exploit must execute a complete execution stack, starting from the shellcode and running down through to a lowest level user function, e.g., a function that includes machine or assembly instructions. (EX-1001 4:24-28.) An exploit will commonly inject shellcode into a target process before or at the same time as it exploits a vulnerability. (EX-1001 4:30-32.) Shellcode includes a malicious payload, which is capable of any one of a number of malicious actions, e.g., executing another process, loading a

module into process memory, adding an admin account, or automatically removing lines from log files. (EX-1001 4:32-35, 3:34-37, 7:5-10, 7:61-62.).

Prevention of software exploits from executing code is a paramount security concern for system users and administrators. (EX-1001 1:23-25.) Before the '591 patent, early anti-exploit methods and security platforms typically relied on identifying signatures, i.e., unique identifiers, known code patterns, or corrupted memory, to detect malware. (EX-2001 ¶42.) But these approaches were extremely resource- and time-intensive to employ, as well as costly. (*Id.*)

Despite substantial use of resources and time, conventional systems were still unable to identify and prevent emerging “zero-day” threats from new or unknown malicious code. (EX-2001 ¶43.) By the early-to-mid-2000s, new threats escalated as network connectivity became widespread, and programs that mutate slightly with each new copy, e.g., polymorphic programs, appeared. (*Id.*) These events, among others, rendered traditional signature-based virus scan systems ineffective. (*Id.*)

Many prior exploit prevention technologies relied on the ability to prevent an exploit from ever starting its own shellcode, such that it could never execute the shellcode's malicious payload. (EX-1001 6:24-30, 7:56-62.) However, such methods were ineffective when the exploit was able to start its own shellcode. (EX-2001 ¶44.) Accordingly, an improved anti-exploit technology was needed.

V. THE '591 PATENT'S NEW ANTI-EXPLOIT TECHNOLOGY

A. Overview of the '591 Patent

The '591 patent claims an “anti-exploit” technique to prevent undesirable computer exploits from executing a malicious payload. (EX-1001 1:13-33.) Unlike prior art anti-exploit systems, which sought to prevent an exploit from ever even starting its own shellcode, the '591 patent describes an anti-exploit component that prevents shellcode from ultimately executing a malicious payload. (*Id.* 6:24-30.) Patent Owner's patented technology enables automatic and real-time analysis, identification, and neutralization of previously unknown malicious code. (EX-2001 ¶45.)

The novel anti-exploit techniques described and claimed in the '591 patent involve the loading of a component, e.g., a custom module, into the memory space of a monitored process. (EX-1001 4:45-47.) For example, the '591 patent advises monitoring memory space for functions comprising processes that attempt to execute another process or load a module into process memory. (*Id.* 4:58-64, cl. 1.)

When a monitored function is invoked, for instance as a result of receiving an API call, the system executes “stack walk processing.” (EX-1001 1:33-39.) Stack walk processing can walk a call stack “for the process (function) until an address of the first function of a called function is reached.” (*Id.* 5:26-29.) In one embodiment,

the anti-exploit technology may walk the stack frames starting from the lowest-level user mode function to an address where the call sequence started. (*Id.* 5:29-39.) In stack walk processing, a base pointer address and stack pointer address may be identified for a stack frame associated with the called function. (*Id.* 5:43-46.)

The anti-exploit method performs a memory check against a plurality of stack entries identified during stack walk processing and triggers an alert if the check detects any one of enumerated suspicious behaviors. (EX-1001 1:39-48, 2:34-38, cl. 1.) For example, claim 1 of the '591 patent lists six possible detections that might trigger an alert. (*Id.* cl. 1.) In one embodiment, a memory check is performed before the address of the originating caller function is reached. (*Id.* cl. 1, Fig. 2.) When an alert of suspicious behavior is triggered, the anti-exploit method prevents execution of a payload for the invoked function. (*Id.* 2:1-3, 6:24-30.)

The innovations of the '591 patent achieved a number of advantages over prior art anti-exploit methods, including “enhanced security protection, improved detection of potential security exploits, reduction in error rate in identification and marking of suspicious behavior (e.g., false positives), and improved usability and interaction with users who are not required to continuously monitor for security exploits, among other examples.” (EX-1001 2:44-51; EX-2001 ¶50.)

B. Claim Construction

Petitioner purports to apply the plain and ordinary meaning of all claim terms.
(Petition 9-10.)

On March 16, 2023, in parallel district court litigation, the Court construed seven terms of the '591 patent as having their plain and ordinary meaning. (EX-2017 12-17.) For the dependent claim term “low level user mode function,” the Court clarified that the plain and ordinary meaning is “user mode function that is low level with respect to its layer of abstraction.” (*Id.* 14.)

Patent Owner and its expert apply the plain and ordinary meaning of all claim terms consistent with the district court's *Markman* order. (EX-2001 ¶52.)

C. Level of Ordinary Skill in the Art

A POSITA at the time of the invention of the '591 patent would have had a bachelor's degree in an accredited program of electrical engineering, computer engineering, computer science, or a similar discipline, and 2-3 years of practical work or research experience in the fields of electrical engineering, computer science, networking, communications, and/or device and network security. More advanced degrees and/or training in a related discipline can compensate for shorter work experience. (EX-2001 ¶¶53-57 (noting that minor distinctions in Petitioner's and Patent Owner's definitions would not alter the analysis).)

VI. OVERVIEW OF THE PRIOR ART

A. Overview of Dalcher

In the field of software security, one traditional approach for analyzing malware is known as “behavioral monitoring and analysis.” (EX-1004 0001.) In this approach, the “security software may depend on the ability to widely monitor system activity.” (*Id.* 0002.) “For example, security software may monitor execution flow [of software] in order to determine if code involved in handling a monitored operation is trusted and expected to be involved in the handling of the operation.” (*Id.*)

Dalcher recognizes that the ability to monitor software execution can be limited by operating system restrictions (e.g., “the use of interface hooking may be prohibited in the kernel”) or by its “performance impact.” (*Id.* at 0003.) To this end, Dalcher discloses a system, method, and computer program product designed to monitor the execution flow of a single function executed on a computer. (EX-1004 0004.) To achieve its goal, Dalcher identifies and analyzes “data associated with a function . . . within a call stack.” (EX-1004 0017.)

A call stack is a data structure that stores the data associated with one or more functions. (EX-1004 0017, 0019; EX-2001 ¶¶32-37.) The call stack is made up of entries, which can include a return address (i.e., the address that identifies the

function that pushed the entry onto the call stack), local variables, reserved space for variables, and register values. (EX-1004 0018.)

Dalcher discloses a three-step approach: (1) identify the data associated with a particular function on the call stack; (2) determine a call stack frame from freed memory; (3) monitor an execution flow associated with the function using the frame from freed memory to determine a level of confidence. (EX-1004 0016, Fig. 3; EX-2001 ¶61.)

In the first step, Dalcher identifies “the data associated with a function.” (EX-1004 0017, 0020, Fig. 3.) Dalcher discloses a variety of means for identifying the data associated with a particular function, such as generating a test event to determine which functions on the call stack are associated with the test event or monitoring a call stack via hooking. (EX-1004 0019-0020.)

In the second step, Dalcher determines a call stack frame from freed memory—a technique that is central to Dalcher's claimed invention. (EX-1004 0021, cls. 1, 18-19, Fig. 3; EX-2001 ¶63.) Dalcher explains that “freed memory” is “memory in the stack that previously held an active stack entry.” (EX-1004 0021.) When execution of a stack frame is complete, that frame is “popped” from the call stack (i.e., it is no longer active). (EX-2001 ¶63.) The stack space that previously held that data (i.e., the “freed memory”) may then be overwritten. (EX-2001 ¶63.)

Dalcher explains that while the active stack pointer is moved to make the freed stack memory unavailable via normal stack usage when a stack frame is popped, freeing memory does not require overwriting the contents of the memory. (EX-1004 0038.)

Dalcher teaches that the stack space may still contain data until it is later overwritten—for example, by another function pushing a new call frame onto the stack. (EX-1004 0038; EX-2001 ¶63.) Thus, the call stack determined “from freed memory” may refer to a call stack frame outside of the active call stack (i.e., below or outside the active stack pointer). (*E.g.*, EX-1004 0021.) It may also refer to the call frame that is pushed onto the active call stack and which resides (at least partially) in the now-overwritten freed memory. (*E.g.*, EX-1004 0021; EX-2001 ¶63.)

In the third step, Dalcher discloses monitoring the execution flow associated with the function, using the call stack frame from the freed memory. (EX-1004 0023, Fig. 3.) Figure 4 illustrates using extended call stack analysis “to assemble a picture of an execution flow” for a function. (EX-1004 Fig. 4, 0027, 0048.) Dalcher initially determines the scope of its monitoring, for example, by generating a test event and limiting its monitoring to the handling of that event. (*E.g.*, EX-1004 0028, Fig. 4.) When a callback is received that falls within the monitored scope, Dalcher discloses

the use of stack analysis to assess related call frames. (EX-1004 0032-0033, Fig. 4 (steps 408-410).)

Figure 5 illustrates a method of call stack analysis in which Dalcher identifies a candidate stack frame (which may be identified from freed memory) and “determine[s] a level of confidence” in the candidate stack frame’s authenticity. (EX-1004 0035, 0039-0041, Fig. 5.)

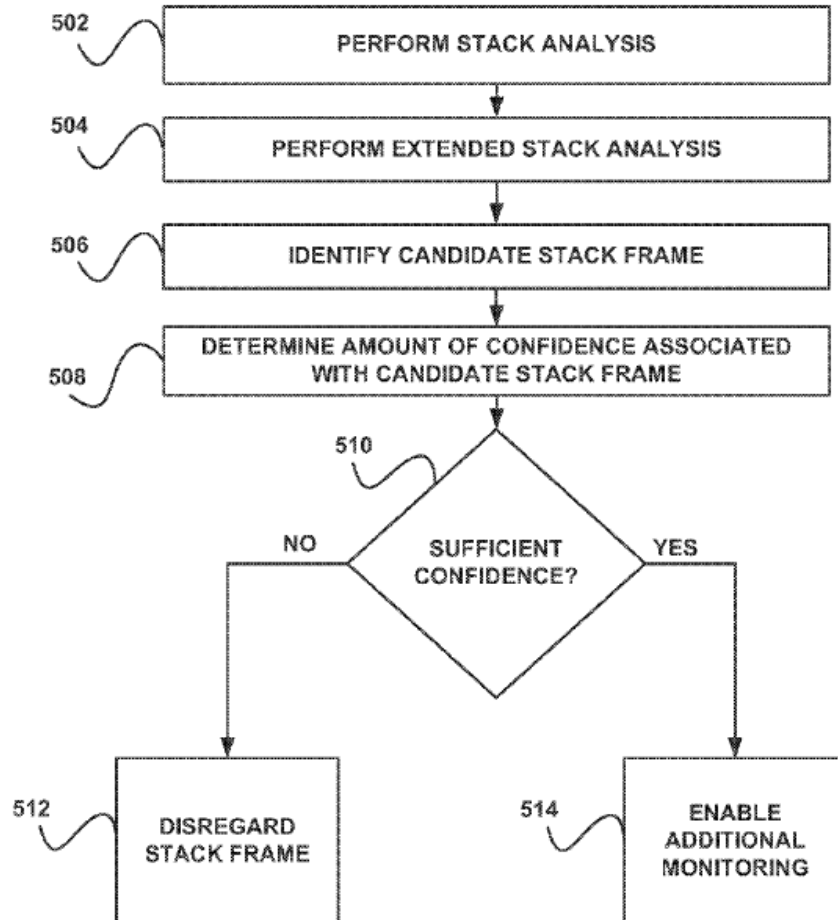


FIGURE 5

According to Dalcher, “confidence” is a measure of whether a stack frame is “genuine.” (EX-1004 0041.) A genuine stack frame would be one that, for example, contains “a suspected call return address [that] points to code, and the code includes a call to the candidate stack frame” or, as another example, one that contains a link to a verifiably genuine stack frame. (EX-1004 0041.) Based on whether the determined confidence level meets a predefined threshold, Dalcher will either disregard the low-confidence call frame or it will enable additional monitoring to validate the call frame. (EX-1004 0042-0043, Fig. 5 (steps 512-514), 0033, Fig. 4 (steps 410-412).) Additional monitoring includes, for example, monitoring “within code pointed to by the suspected call return address” of the candidate stack frame. (EX-1004 0044.)

Thus, Dalcher continuously monitors past and pending execution flow of a function, both within the call stack and in freed memory, to assemble a picture of the genuine execution flow of the function. (EX-1004 0047.)

B. Overview of Pappas

Return-oriented programming (ROP) is a program-exploitation technique that “has sometimes been illustrated like a ransom note created from cut-out letters from newspapers and magazines.” (EX-1002 ¶119.) During ROP, a malicious program identifies the location of a known standard library in memory and executes small

sequences of instructions from within that library. (EX-1002 ¶116-118.) These sequences are known as “gadgets” and allow the attacker to build what is essentially a new program out of an array of gadgets. (EX-1002 ¶116-118.) Pappas is “a practical runtime ROP exploit prevention technique for the protection of third-party applications.” (EX-1005 2:1.⁵)

Pappas's disclosure is prototyped in Windows 7 in a program called “kBouncer.” (EX-1005 9:2.) The kBouncer module operates in kernel space and functions as an interface for inspecting the Last Branch Recording (“LBR”) stack. (EX-1005 Fig. 8.) The LBR stack is a series of model-specific registers used by certain types of Intel CPUs to “track[] the last N (16 for the CPU model [Pappas] used) most recent branches” of an executing program. (EX-1005 4:1.) Each branch record holds two values: the linear address of the branch instruction (where the code is located that initiated the jump) and the target instruction (where the process will jump to). (EX-1005 4:1.)

Pappas's ROP exploit prevention technique is based on two common characteristics of ROP techniques: “i) illegal *ret* instructions that target locations not

⁵ Citations to EX-1005 (Pappas) refer to page and column, *e.g.*, EX-1005 2:1 refers to page 2, column 1.

preceded by call sites, and ii) sequences of relatively short code fragments “chained” through any kind of indirect branches.” (EX-1005 6:2.) To identify the first characteristic, Pappas checks for whether the *ret* instruction points to a location in the code that is preceded by a function call (as a legitimate *ret* instruction would). (EX-1005 6:1-2.) To accomplish this, kBouncer “fetches a few bytes before each return target and attempts to decode any *call* instruction located right before the target instruction.” (EX-1005 10:2.) If no *call* instruction is found, the return is considered illegal. (EX-1005 7:1.) To identify the second characteristic, Pappas checks for a sequence of gadgets exceeding eight gadgets in length. (EX-1005 9:2.) To identify gadgets, Pappas discloses an “offline gadget extraction and analysis toolkit,” which scans the executable segments of a protected application “in advance” and creates hash tables of the offsets of the gadgets. (EX-1005 9-10:1.) Using the hash tables, “the number of consecutive targets that point to gadgets are counted” and if the threshold is exceeded, then Pappas considers the chain suspicious. (EX-1005 10:2.)

Pappas teaches that kBouncer takes a two-tiered approach. (EX-1005 6:1, Fig. 3.) First, kBouncer “selectively hooks the set of the protected Windows API functions.” (EX-1005 10:1.) When a protected function is called, kBouncer is triggered to perform an “LBR check” by “inspect[ing] the contents of the LBR stack”

for the characteristics described above. (EX-1005 6:1, 10:1.) If “an attack attempt is detected after the analysis of the recorded branches, the process is terminated, and the user is informed with an alert message.” (EX-1005 10:2.) But if the LBR check is clear at the API function’s entry point, kBouncer writes a checkpoint that denotes a legitimate invocation of that particular function. (EX-1005 6:1.)

In the second tier, when the respective system call is later invoked, Pappas verifies the checkpoint was properly set, before allowing the monitored system call to proceed. (EX-1005 6:1, 10:1.) In this way, Pappas establishes a gating mechanism that blocks code from executing unless a checkpoint is “previously set by the expected API function,” even if neither of the ROP characteristics from tier one was detected. (EX-1005 6:1.)

**VII. PETITIONER FAILS TO SHOW A REASONABLE LIKELIHOOD
THAT IT WOULD PREVAIL WITH RESPECT TO ANY
CHALLENGED CLAIM**

Ground 1	Section 103(a)	Obvious over Dalcher and Pappas	Claims 1-18
-----------------	----------------	---------------------------------	-------------

An IPR may be instituted *only if* the petitioner demonstrates a “reasonable likelihood” that it would prevail as to at least one of the challenged claims. 35 U.S.C. §314(a). “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); *InfoBionic Inc. v. Braemer Mfg., LLC*, IPR2015-01705, Paper 11 at 11 (PTAB Feb. 16, 2016) (denying institution where petitioner’s failure to “articulate” its combination left the Board to “speculate as to how [p]etitioner’s combination of references’ teachings would render obvious” the challenged claims); *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).

Petitioner asserts a single Section 103 ground in which Petitioner asserts that a “[POSITA] would have looked to the teachings of Pappas to modify the teachings in Dalcher.” (Petition 4, 13.) But Petitioner’s solitary ground fails to establish a reasonable likelihood that the challenged claims are unpatentable.

First, as addressed in Section A, Petitioner does not adequately explain the combination it asserts as its sole ground. Second, as addressed in Section B,

Petitioner does not show that a POSITA would have been motivated to combine Dalcher with Pappas and, in fact, a POSITA would not have been so motivated. Finally, as addressed in Section C, Petitioner has failed to show that the combination of Dalcher and Pappas discloses or renders obvious all limitations of any challenged claim.

A. Petitioner Does Not Adequately Articulate a Combination of Dalcher's and Pappas's Teachings

“It is Petitioner's burden to explain specifically how the references would be combined to produce the claimed invention.” *Lowe's Home Centers, LLC v. Epistar Corp.*, IPR2018-00932, Paper 7 at 19 (PTAB Nov. 1, 2018) (citing *TriVascular, Inc. v. Samuels*, 812 F.3d 1056, 1066 (Fed. Cir. 2016); *In re Magnum Oil Tools Int'l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016)). The petition lacks this explanation.

Instead, the petition reads like a choose-your-own adventure book (e.g., Petition 14-17 (“Dalcher discloses....Pappas also discloses....Therefore, Dalcher and Pappas each disclose this limitation.”)); Petitioner repeatedly recites what Dalcher and Pappas purportedly teach *individually* but fails to assert “what the **combined teachings** of those references would have suggested to those of ordinary skill in the art.” *In re: Keller*, 642 F.2d 413, 425 (CCPA 1981) (restating the test for Section 103 obviousness). The Board has rejected “[t]his ‘pick-and-choose’ presentation of evidence,” which “is challenging to follow and makes it difficult to

piece together in order to understand Petitioner's proposed combination or modification." *Lyft, Inc. v. Quartz Auto Techs., LLC*, IPR2020-01450, Paper 7 at 17-20 (Mar. 4, 2021).

"It [i]s Petitioner's 'burden to explain to the Board how' the combination of references 'rendered the challenged claims unpatentable.'" *TikTok Inc. v. Advanced Coding Techs. LLC*, IPR2022-01546, Paper 9 at 7 (PTAB June 15, 2023) (citing *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016)). While the features of a secondary reference need not be bodily incorporated into the structure of the primary reference, "[a] clear, evidence-supported account ***of the contemplated workings of the combination is a prerequisite*** to adequately explaining and supporting a conclusion that a relevant skilled artisan would have been motivated to make the combination and reasonably expect success in doing so." *Pers. Web Techs., LLC v. Apple, Inc.*, 848 F.3d 987, 994 (Fed. Cir. 2017) (overturning a panel's decision where "the Board nowhere clearly explained, or cited evidence showing, *how* the combination of the two references was supposed to work").

Accordingly, Petitioner's non-specific approach to its solitary Section 103 ground cannot meet its burden to establish a reasonable likelihood of success. *See ActiveVideo Networks, Inc. v. Verizon Comms., Inc.*, 694 F.3d 1312, 1327 (Fed. Cir. 2012) (affirming summary judgment of no invalidity where the defendant's expert

“failed to explain how specific references could be combined, which combination(s) of elements in specific references would yield a predictable result, or how any specific combination would operate or read on the asserted claims”); *ADT LLC v. Vivint, Inc.*, IPR2022-00642, Paper 7 at 13-14 (PTAB Oct. 4, 2022) (denying institution and finding a lack of clarity where Petitioner had mapped each limitation to both references but relied on the combination “without any discussion of the interplay between these two references”); *Lyft*, IPR2020-01450, Paper 7 at 18 (denying institution where petitioner “[did] not articulate adequately how it propose[d] to modify one of the references or to combine the two references’ teachings”).

Nowhere does Petitioner provide a “clear, evidence-supported account of the contemplated workings of the combination.” *See Pers. Web Techs.*, 848 F.3d at 994. The petition is organized such that Petitioner attempts to address the combination of Dalcher and Pappas and alleged motivation to combine up front, at pages 10-14, and then the petition makes extremely brief mentions of the purported combination within individual limitations. (*E.g.*, Petition 17 (limitation 1.b), 18 (limitation 1.c).) But neither the general motivation-to-combine section nor the cursory mentions of a combination in certain limitations meets Petitioner’s burden to explain what the combined teachings of Dalcher and Pappas would have suggested to a POSITA.

First, in its general motivation-to-combine discussion, Petitioner makes only conclusory statements that a POSITA would have combined the references without ever identifying or explaining a particular combination of prior art teachings. (See Petition 12-13.) For example, Petitioner proposes that “[a] [POSITA] would have implemented Pappas and Dalcher together to address the common exploitation technique of return oriented programming by analyzing the relationships between stack frames (calls).” (Petition 13.) This vague suggestion merely parrots the abstract purpose of each reference. Petitioner also alleges that a POSITA would have modified the teachings in Dalcher based on Pappas’s teachings “in order to enhance the tracking of calls, jumps, and returns and to reduce false positives in compromise detection rates.” (Petition 13.) But none of Petitioner’s statements about what might have motivated a POSITA to combine the teachings actually identifies which of Dalcher’s teachings a POSITA would have modified, how they would have been modified, and by which teachings of Pappas. (See Petition 12-13.)

While Petitioner asserts that “both references relate to detecting malware” using analogous methods (Petition 12), a “cursory assertion that the references are drawn from the same broadly characterized field” is not sufficient for Petitioner to meet its burden. *Securus Techs., Inc. v. Global Tel*Link Corp.*, IPR2015-00155, Paper 30 at 17 (PTAB Apr. 7, 2016). Even if both references use similar techniques,

that fact does not explain how they would be combined, much less explain their combined teachings. For example, Petitioner identifies purportedly similar features of each reference to conclude that they are “compatible” (e.g., “Dalcher teaches analysis of calls preserved in portions of memory called freed memory” and “Pappas teaches a compatible cache (preserved LBR stack) for analysis”). (Petition 13-14.) But Petitioner never explains what the suggested combination of these features would be or which teachings of each reference a POSITA would be motivated to use in a combination. (*Id.*)

Second, Petitioner’s limitation-specific analysis of the independent claims merely discusses Dalcher’s and Pappas’s teachings individually and still does not provide an “account of the contemplated workings *of the combination*.” See *Pers. Web Techs.*, 848 F.3d at 994. The below table includes, for each limitation in claim 1, the entirety of Petitioner’s discussion of the purported combination⁶:

Claim 1 Limitation	Petitioner’s Asserted Reference
1.a	“Therefore, Dalcher discloses the preamble.” (Petition 14.)

⁶ Though not identical verbatim, Petitioner’s discussion of the purported combination in the other independent claims is not substantively different for each limitation. (See Petition 38-50, 56-59.)

Patent Owner's Preliminary Response to
Petition for Inter Partes Review
Patent No. 10,284,591

1.b	“Therefore, Dalcher and Pappas each disclose this limitation.” (Petition 17.)
1.c	“Therefore, Dalcher in view of Pappas discloses this limitation.” (Petition 18.)
1.d	“Therefore, Dalcher discloses this limitation.” (Petition 21 (while also explaining how Pappas individually discloses this limitation).)
1.e	“Therefore, Dalcher discloses elements listed in this limitation and Pappas discloses compatible and consistent memory stack monitoring, any of which are sufficient to render the limitation obvious.” (Petition 25 (quoting a standard for anticipation by the prior art).)
1.f	“A PHOSITA would have looked to Pappas for further compatible details on preventing execution of quarantined software or functions when a compromise (suspicious behavior) is detected. Therefore, Dalcher and Pappas disclose this limitation.” (Petition 26.)
Conclusion	“[A] PHOSITA ... would have found claim 1 of the '591 patent to be obvious in view of the combination of Dalcher and Pappas

	because these prior art publications disclose each and every limitation recited in the claim.” (Petition 26.)
--	---

With the exception of the above statements, in each limitation Petitioner only recites what each reference discloses individually. (*E.g.*, Petition 14-17 (“Dalcher discloses....Pappas also discloses....Therefore, Dalcher and Pappas each disclose this limitation.”).) Such a showing is insufficient to establish obviousness under Section 103. *ActiveVideo*, 694 F.3d at 1327; *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007) (“A patent composed of several elements is not proved obvious by merely demonstrating that each of its elements was, independently, known in the prior art.”).

Once again, the result is a mix-and-match petition in which Petitioner leaves it up to the Board (and Patent Owner) to interpret which elements from each reference a POSITA would choose to combine or what the combined teachings might suggest to a POSITA. “It is not the Board’s job to cobble together assertions from different sections of a petition or citations of various exhibits in order to infer every possible permutation of a petitioner’s arguments,” and such an approach is insufficient to meet Petitioner’s burden. *Microsoft Corp. v. FG SRC, LLC*, 860 Fed. Appx. 708, 713 (Fed. Cir. 2021) (finding that the petitioner failed to challenge a

particular claim based on obviousness with any particularity whatsoever); *Liberty Mutual Ins. Co. v. Progressive Cas. Ins. Co.*, CBM2012-00003 Paper 8 at 14 (PTAB Oct. 25, 2012) (“[W]e address only the basis, rationale, and reasoning put forth by the [p]etitioner and resolve all vagueness and ambiguity in [p]etitioner’s arguments against the [p]etitioner”); *Lyft*, IPR2020-01450, Paper 7 at 17-20 (rejecting “‘pick-and-choose’ presentation of evidence”); *Palo Alto Networks, Inc. v. Centripetal Networks, Inc.*, IPR2021-01153, Paper 10 at 16-17 (PTAB Jan. 24, 2022).

Moreover, instituting an IPR on such a petition would be prejudicial to Patent Owner for several reasons. Because Patent Owner is unable to discern the manner in which Petitioner suggests Dalcher and Pappas should be combined, Patent Owner is limited in its ability to address some unspecified combination of disclosures from the petition. *T-Mobile US, Inc. v. TracBeam LLC*, IPR2015-01687, Paper 10 at 20 (Feb. 12, 2016) (“[w]e will not...improperly shift the burden of deciphering the arguments onto the Patent Owner”). Additionally, institution of trial on such a petition would enable Petitioner to develop arguments not initially presented in the petition. *ADT*, IPR2022-00642, Paper 7 at 13-14.

Because Petitioner does not discuss “***the contemplated workings of the combination,***” Petitioner’s single Section 103 ground cannot succeed. *Pers. Web*

Techs., 848 F.3d at 994; *ActiveVideo*, 694 F.3d at 1327; *ADT*, IPR2022-00642, Paper 7 at 13-14.

B. There Is No Motivation to Combine the Teachings of Dalcher and Pappas

1. Petitioner fails to show why a POSITA would have been motivated to combine Dalcher with Pappas to practice the claimed invention.

As referenced above, Petitioner provides only vague assertions that a POSITA would have combined Dalcher and Pappas based on the references being related in purpose and sharing some features. (See Petition 12-13.) But even if these statements were true (which they are not, as discussed below), “vague assertions do not substantiate a reason that would have prompted one of ordinary skill in the art to combine the specific relevant *element* or *teaching* of [one prior art reference with another] to achieve the invention recited in the challenged claims.” *Duo Security Inc. v. StrikeForce Techs., Inc.*, No. IPR2017-01064, 2017 WL 4677238, at *11 (PTAB Oct. 16, 2017). A proper motivation to combine “requires consideration of whether ... the hypothetical skilled artisan would recognize the potential benefits and pursue the variation.” *Belden Inc. v. Berk-Tek LLC*, 805 F.3d 1064, 1075 (Fed. Cir. 2015) (“[O]bviousness concerns whether a skilled artisan not only could have made but would have been motivated to make the combinations or modifications of prior art to arrive at the claimed invention.”).

ActiveVideo is instructive, in which the Federal Circuit, upon determining that the defendant's evidence regarding motivation to combine was "not sufficient and [] fraught with hindsight bias," held:

This testimony is generic and bears no relation to any specific combination of prior art elements. It also fails to explain why a person of ordinary skill in the art would have combined elements from specific references *in the way the claimed invention does*. See *KSR*, 550 U.S. at 418, 127 S. Ct. 1727 ("[I]t can be important to identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does...because inventions in most, if not all, instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known."); *Innogenetics*, 512 F.3d at 1373 (Fed. Cir. 2008) ("Knowledge of a problem and motivation to solve it are entirely different from motivation to combine particular references....").

694 F.3d at 1328 (emphasis in original).

Here, similarly, Petitioner's alleged motivation to combine relies on generalized statements about Dalcher's and Pappas's fields of endeavor, allegedly similar features, or the purported results that could be achieved through some unspecified combination of their teachings. For example, Petitioner alleges that both Dalcher and Pappas are directed to malware detection and use some form of stack

analysis to provide solutions to the allegedly same problem. (Petition 12-13.) But Petitioner never provides a reason why a POSITA would have modified or combined *particular* teachings of either reference to reach the claimed invention. (*See id.*)

Thus, despite Petitioner's conclusory argument that "a more than adequate written rationale is provided to support the combination of Dalcher and Pappas under *KSR*" (Petition 11), Petitioner's vague and generalized assertions do not meet its burden to establish why a POSITA would have been motivated to combine particular teachings of Dalcher and Pappas to practice the claimed invention.

2. Petitioner fails to establish a reasonable likelihood of success.

Petitioner incorrectly alleges that a combination of the references could achieve two benefits (while still failing to specify *how* such benefits would be achieved or *what combination* would achieve such benefits). (*See* Petition 13.) Specifically, Petitioner alleges that a POSITA would have modified Dalcher in view of Pappas "in order to [1] enhance the tracking of calls, jumps, and returns and [2] to reduce false positives." (*Id.*)

First, Petitioner does not explain why a POSITA would expect that modifying Dalcher in view of Pappas would enhance the tracking of calls, jumps, and returns. (*See* Petition 13.) And while Petitioner cites its expert, he does not elaborate. (*See* EX-1002, ¶¶185-189.) Particularly where Dalcher already discloses a method for

monitoring execution flow on a call stack, it would be unclear to a POSITA how Pappas's more limited method for monitoring branch records would enhance Dalcher's evaluation. (EX-2001, ¶¶76-77.)

Second, Petitioner also fails to explain why a POSITA would expect that modifying Dalcher in view of Pappas would result in a lower false positive rate. (See Petition 13.) Indeed, it would not. (EX-2001 ¶¶78-81.) The only means Pappas discloses for achieving an “increased resilience to false positives” is by “set[ting] a robust detection threshold”—specifically a “threshold of eight [chained] gadgets” for the disclosed set of protected functions. (EX-1005 9:2.) As explained above in Section VI.B, one of the attributes Pappas checks for in the branch records is a sequence of “chained” gadgets that exceeds a threshold length. (EX-1005 6:2.) For the disclosed set of protected functions, Pappas selected a threshold of eight chained gadgets to balance protection and resilience to false positives. (EX-1005 9:2.)

Petitioner does not explain how this teaching from Pappas might be applied to Dalcher or—even if it were—how it would result in fewer false positives than Dalcher's own method. As discussed above in Section VI.A, Dalcher does not use LBRs or monitor for sequences of gadgets. (EX-2001, ¶81.) As a result, Pappas's “robust threshold” number for a sequence of gadgets could not be used in Dalcher. (EX-2001, ¶¶80-81.) While Dalcher discloses a threshold confidence determination

(EX-1004 [0042], Fig. 5), this threshold has nothing to do with gadget chains, nor does Dalcher use this threshold determination to terminate execution, as Pappas does. (EX-1005 10:2; EX-2001, ¶¶80-81.) Rather, it uses a confidence threshold merely to determine whether a candidate call frame should be disregarded from Dalcher's subsequent monitoring steps or whether additional monitoring should be enabled. (EX-1004 [0042], Fig. 5.) Thus, it is unclear even what "false positives" Dalcher might generate that would need to be reduced. (EX-2001, ¶79.)

Furthermore, Pappas explicitly teaches that its threshold of eight gadgets is specifically set "[f]or the current set of protected functions." (EX-1005 9:2.) Pappas acknowledges that in other applications there may be a need to protect more API functions or perform LBR checks in other parts of a program, for which a different threshold would have to be determined. (EX-1005 9:2.) Dalcher does not disclose monitoring an identical set of protected functions, or monitoring branch records at all. Accordingly, a POSITA would not be motivated by Pappas's teaching to implement any kind of similar threshold in Dalcher to reduce false positives, particularly when doing so would serve no purpose in Dalcher's method and would not, in fact, reduce false positives. (EX-2001, ¶¶78-81)

Petitioner's analysis is plagued with hindsight bias. The pertinent analysis is not "whether a person of skill, with two (and only two) references sitting on the table

in front of him, would have been motivated to combine ... the references in a way that renders the claimed invention obvious.” *WBIP, LLC v. Kohler Co.*, 829 F.3d 1317, 1337 (Fed. Cir. 2016). “The real question is whether that skilled artisan would have plucked one reference out of the sea of prior art ([Pappas]) and combined it with [Dalcher] to address some need present in the field.” *See id.* Petitioner has not identified why a POSITA would have picked Pappas to combine with Dalcher, why a POSITA would have found Pappas’s specific teachings useful, or why a POSITA would have had a reasonable expectation of success in the unspecified combination.

3. A POSITA would not have been motivated to combine the teachings of Dalcher and Pappas.

Even setting aside Petitioner’s failure of proof on this issue, a POSITA would not, in fact, have been motivated to combine the teachings of Dalcher and Pappas. Petitioner cherry-picks a few discrete features from Dalcher and Pappas and describes them in an exceedingly abstract manner in order to cast them as similar. (Petition 12-13.) But the only fair reading of Dalcher and Pappas is that they disclose fundamentally different systems and methods with distinct goals, such that a POSITA would not have been motivated to combine their teachings.

First, there is a significant difference between Dalcher’s *software*-based analysis and Pappas’s “[reliance] mainly on *hardware* for instruction-level monitoring.” (Compare EX-1004 0014 with EX-1005 3:1.) Pappas’s method relies

on its ability to evaluate LBR registers, a “hardware feature[] of commodity processors.” (EX-1005 2:1; EX-2001, ¶85.) Pappas advocates reliance on hardware because it “allows for minimal runtime overhead and completely transparent operation, without requiring any modifications to the protected applications.” (EX-1005 3:1.)

Combining the teachings of these systems would be impractical, and a POSITA would not be motivated to do so. Dalcher's analysis of a call stack, which resides in virtual memory and which is loaded upon functions being invoked, could not be achieved using hardware. (EX-1004 0021; EX-2001, ¶85.) And implementing Pappas's teachings in Dalcher's software-based model—besides violating Pappas's guidance to rely on hardware— would impact runtime, decrease transparency, and require drastic modification of Pappas's method. (EX-1005 4:2.)

Second, Dalcher is directed to monitoring a call stack in process memory, whereas Pappas is directed to monitoring LBR registers in the CPU. (*Compare* EX-1004 Abstract, 0018 *with* EX-1005 4:1.) That is, Dalcher and Pappas are directed to monitoring different things. (EX-2001, ¶86.) There would have been no reason—and Petitioner identifies none—to apply Pappas's teachings, which are specific to the LBR registers contained in the CPU, to Dalcher's distinct evaluation of a call stack in process memory. (EX-2001, ¶86.)

Finally, while Dalcher and Pappas both arguably analyze a stack, this is as far as their similarities go. (EX-2001, ¶86.) For example, Dalcher discloses “stack analysis” as a means for determining “an amount of confidence associated with the candidate stack frame” and, as a result, whether that stack frame should be disregarded or whether Dalcher’s monitoring method should enable additional monitoring to validate the candidate stack frame. (EX-1004 0039, 00042-0043, Fig. 5.) Pappas, by contrast, analyzes the LBRs as a preliminary means to determine whether a checkpoint should be written, which will then be used to authorize a subsequent invocation of a system call. (EX-1005 6:1-2, Fig. 3.) These are entirely distinct methods and goals, and the mere fact that both mention stack analysis does not mean that a POSITA would have been motivated to apply any teaching from one to the other. (EX-2001, ¶86.)

The fact that Dalcher’s and Pappas’s methods are fundamentally different is best illustrated when one considers that each method—applied in the same scenario—would produce a different result. (EX-2001, ¶¶87-90.) For example, Dalcher discloses that one method of “determin[ing] [] whether the candidate call stack frame is genuine” is determining whether “a suspected call return address points to code [that] includes a call to the candidate stack frame.” (EX-1004 0041.) In this way, Dalcher analyzes whether the return address is legitimate by identifying

that it returns to a call site specific to that frame. (EX-2001, ¶88.) Pappas, on the other hand, explicitly states that “[i]nstead of enforcing a strict control flow, kBouncer simply makes sure that *ret* instructions always target *any* among all valid call sites (even those that correspond to non-intended *call* instructions).” (EX-1005 7:2.) That is, Pappas discloses a simpler implementation intended to reduce processing overhead. (EX-1005 7:2.)

Thus, under a scenario in which an illegal return happens to point to a call site of a different function, Pappas would not flag such an instruction as indicative of a return-oriented programming exploit, because it is a legitimate target “among all valid call sites.” (EX-1005 7:2; EX-2001, ¶89.) Dalcher, however, would determine this instruction to have low confidence—perhaps suggesting an ingenuine frame—because the call site invocation does not match the candidate stack frame function. (EX-1004 0041; EX-2001, ¶89.) Where Pappas and Dalcher employ fundamentally distinct methods for distinct purposes, such that they would provide different results under the same scenario, a POSITA would not have been motivated to apply the teachings of one to the other. (EX-2001, ¶90.)

C. Petitioner Fails to Show That the Combination of Dalcher and Pappas Renders Obvious Any Independent Claim

Petitioner asserts that Dalcher in view of Pappas renders obvious the challenged claims. (Petition 4, 10.)⁷ As discussed for each of the three exemplary claim limitations below: (1) Dalcher's method is fundamentally distinct from the claimed inventions of the '591 patent and does not disclose every limitation of any independent claim; and (2) because Petitioner does not adequately explain how or why a POSITA would modify Dalcher in view of Pappas or combine their teachings, Petitioner fails to show that Dalcher in view of Pappas renders obvious all limitations of the challenged claims. *See ADT*, IPR2022-00642, Paper 7 at 13 (finding petition insufficient because it failed to acknowledge "any shortcomings or deficiencies of the references," making it unclear which teachings of which reference should be combined in each claim).

⁷ Petitioner does not argue that Dalcher, alone, renders obvious any limitations of the independent claims.

1. Petitioner fails to show that the combination of Dalcher and Pappas renders obvious performing a memory check before an address of an originating caller function is reached (1.e, 9.f, 17.e).

Petitioner asserts that Dalcher discloses this limitation and that Pappas discloses a portion of this limitation, and then concludes that Dalcher and Pappas are compatible. (Petition 21-25.) However, Dalcher does not disclose “performing, during the executing of the stack walk processing before an address of an originating caller function is reached, a memory check” as required by claim elements 1.e, 9.f, and 17.e,⁸ and Petitioner fails to show how or why a POSITA would combine the teachings of Dalcher and Pappas to render this limitation obvious. (EX-2001, ¶91.)

Petitioner argues that “Dalcher traces a chain of function calls ... from most recently called function ... back through [the] stack to an originating function” and thus discloses performing the memory check “before an address of an originating caller function is reached.” (Petition 21-22.) This is incorrect.

⁸ Regarding independent claim 9, Petitioner presents substantively the same arguments as those it presents for claim 1; similarly, Petitioner presents no new arguments regarding independent claim 17. (See Petition 38-50, 56-59.) Accordingly, Petitioner's arguments regarding independent claims 9 and 17 fail for the same reasons as those discussed regarding claim 1.

Dalcher begins its analysis at the call stack frame associated with a monitored function. (EX-1004 Fig. 3 (step 302), 0017; EX-2001 ¶91.) Dalcher subsequently expands its monitoring as it identifies more frames in the call stack associated with the execution flow of the monitored function. (EX-1004 Fig. 4 (steps 402-404, 412), 0028-0029; EX-2001 ¶91.) And while Dalcher discloses both “identifying a current top of the call stack” and “moving through freed stack memory” to “identify one or more preserved call stack frames in the freed stack memory,” neither of these disclosures would suggest to a POSITA that Dalcher’s memory check takes place “before an address of an originating function is reached” because neither indicates where the stack walk begins or when the alleged memory check takes place. (EX-1004 0037; EX-2001 ¶91.)

To the contrary, a POSITA would have understood Dalcher’s disclosure of moving through freed stack memory to simply mean that Dalcher’s method observes previously executed call stack frames, which would require monitoring call frames that are not even in the active call stack. (EX-2001 ¶¶91-92.) As stated above, Dalcher explains that “freed memory” is “memory in the stack that previously held an active stack entry” that has been popped from the call stack. (Section VI.A, *supra*; EX-1004 0021; EX-2001 ¶¶63, 91-92.) Thus, freed memory is not even part of the active stack that contains the originating caller function and likely located at the

opposite end of the call stack from the originating caller function. (EX-2001 ¶¶91-92.) Nothing in Dalcher indicates how Dalcher traverses a stack or whether the alleged memory check is performed before an address of an originating caller function is reached. (*Id.*)

And identifying a current top of the call stack simply means that the monitored function could be at the top of the call stack. (EX-2001 ¶93.) That is, this disclosure still does not indicate where Dalcher's stack walk begins, how it traverses the stack, where the originating caller function is located, or whether the alleged memory check is performed before an address of an originating caller function is reached. Accordingly, Dalcher does not disclose or render obvious performing a memory check before an address of an originating caller function is reached.

Petitioner asserts that Pappas discloses an alleged stack walk and alleged memory check, but does *not* assert that Pappas discloses performing its alleged memory check “*during the executing of the stack walk processing before an address of an originating caller function is reached.*” (See Petition 24.) Again, just like Dalcher, there is no disclosure in Pappas that indicates an address of an originating caller function and thus Pappas cannot disclose whether an alleged memory check is performed before an address of an originating caller function is reached. (EX-2001 ¶94.) Thus, neither Dalcher nor Pappas discloses this portion of these limitations.

Petitioner concludes its argument on this limitation by stating that “Dalcher discloses elements listed in this limitation and Pappas discloses compatible and consistent memory stack monitoring, any of which are sufficient to render the limitation obvious.” (Petition 25.) But, as previously discussed, even if Pappas did disclose this limitation (which it does not), Petitioner does not specify *how* a POSITA would have modified Dalcher in view of Pappas or combined their teachings to practice this limitation or how their combined teachings would have suggested the missing portion of these limitations. *See* Section VII.A, *supra*.

Relatedly, and as discussed above in connection with Petitioner's flawed general discussion of motivation to combine (Section VII.B, *supra*), Petitioner fails to explain *why* a POSITA would have combined specific teachings from Dalcher and Pappas. (Petition 25.) Merely stating that each reference discloses compatible monitoring methods, “any of which are sufficient to render the limitation obvious,” does not satisfy Petitioner's burden to explain “why a person of ordinary skill in the art would have combined elements from specific references *in the way the claimed invention does*.” *ActiveVideo*, 694 F.3d at 1328; *Duo Security*, 2017 WL 4677238, at *11.

Such cursory assertions fail to meet Petitioner's burden to show what the combined teachings of the references would have suggested to a POSITA. *Securus*

Techs., IPR2015-00155, Paper 30 at 17. Accordingly, Petitioner fails to show that Dalcher, or a combination of Dalcher and Pappas, renders obvious performing a memory check before an address of an originating caller function is reached, as required by all challenged claims.

2. Petitioner fails to show that the combination of Dalcher and Pappas renders obvious triggering an alert when the memory check detects one of the enumerated elements (1.e, 9.f, 17.e).

Petitioner argues that Dalcher discloses this limitation and that Pappas discloses this limitation, and then, again, merely concludes that Dalcher and Pappas are compatible. (Petition 21-25.) Dalcher does not disclose “wherein an alert of suspicious behavior is triggered when the performing of the memory check detects at least one of the following [elements]” as required by claim elements 1.e, 9.f, and 17.e., and Petitioner fails to show how or why a POSITA would combine the teachings of Dalcher and Pappas to render this limitation obvious.

Petitioner mischaracterizes Dalcher by alleging that Dalcher teaches generating a flag when “the level of confidence that a compromise has occurred is above a certain threshold.” (Petition 25.) Dalcher’s disclosure of flagging something is limited to a single sentence in the specification, which states: “If the monitoring indicates that that other software was involved in this [event] handling...the monitoring could flag the involvement as outside allowed system operations....”

(EX-1004 0026.) That is, contrary to Petitioner's assertion, Dalcher does not disclose generating a flag during its evaluation of a candidate stack frame or its determination of whether that candidate stack frame has low confidence. (EX-1004 0039, 0042-0043; EX-2001 ¶96.)

The purpose of Dalcher's confidence determination is to assess whether a candidate stack frame is genuinely part of the monitored function's execution flow. (See EX-1004 0041-0043; EX-2001 ¶96.) When confidence in a particular frame does not meet a predetermined threshold, that frame is disregarded—and Dalcher does not even seek to validate that disregarded call frame in its additional monitoring. (EX-1004 0043, Fig. 5.) Thus, Petitioner's assertion that Dalcher discloses any type of alert, flag, or quarantine as a result of its confidence determination is incorrect. Accordingly, Dalcher does not disclose or render obvious these limitations. (EX-2001 ¶¶95-96.)

Petitioner also asserts that Pappas discloses this step of limitation 1.e (Petition 24) and, again, concludes by stating that “Dalcher discloses elements listed in this limitation and Pappas discloses compatible and consistent memory stack monitoring, any of which are sufficient to render the limitation obvious.” (Petition 25.) But, as previously discussed, even if Pappas did disclose a memory check, Petitioner does not specify *how* a POSITA would have modified Dalcher in view of

Pappas or combined their teachings to practice this limitation. *See* Section VII.A, *supra*.

Furthermore, in addition to Petitioner's general discussion of motivation to combine being both insufficient and incorrect (Section VII.B, *supra*), Petitioner fails to explain *why* a POSITA would have combined these specific teachings from Dalcher and Pappas. (See Petition 25.) Merely stating that each reference discloses compatible monitoring methods, "any of which are sufficient to render the limitation obvious," does not satisfy Petitioner's burden to explain "why a person of ordinary skill in the art would have combined elements from specific references *in the way the claimed invention does*." *ActiveVideo*, 694 F.3d at 1328; *Duo Security*, 2017 WL 4677238, at *11.

Such cursory assertions fail to meet Petitioner's burden to show what the combined teachings of the references would have suggested to a POSITA. *Securus Techs.*, IPR2015-00155, Paper 30 at 17. Accordingly, Petitioner fails to show that Dalcher, or a combination of Dalcher and Pappas, renders obvious triggering an alert when the memory check detected one of the claimed elements.

3. Petitioner fails to show that the combination of Dalcher and Pappas renders obvious preventing execution of a payload (1.f, 9.g, 17.f).

Petitioner asserts that Dalcher discloses this limitation, Pappas discloses a portion of this limitation, and that both Dalcher and Pappas disclose this limitation. (Petition 25-26.) Dalcher does not disclose “preventing execution of a payload for the invoked function from operating” when an alert is triggered as required by limitations 1.f, 9.g, and 17.f., and Petitioner fails to show how or why a POSITA would combine the teachings of Dalcher and Pappas to render this limitation obvious.

Petitioner argues that Dalcher’s “placing [of] software or functions in quarantine” prevents them from operating. (Petition 25.) But Dalcher discloses placing “the *system*” in quarantine—not a particular function or software—as an *alternative* to flagging unexpected software involvement, and not as a result of a flag. (EX-1004 0026.)

More importantly, Dalcher does not specify whether the system is quarantined before or after a potential payload might have been executed. (See EX-1004 0026; EX-2001 ¶¶98-99.) Dr. Medvidovic explains that, due to the nature of Dalcher’s method—which uses call frames from freed memory—a POSITA would not understand that the system is quarantined before the payload has been executed.

(EX-2001 ¶¶99.) Dalcher discloses determining a call frame from freed memory in the call stack, i.e., a call frame that has already been executed and which is no longer “pending” execution. (EX-1004 0021.) Dalcher explicitly teaches that such a call frame may include memory that “previously held an active stack entry.” (*Id.*) Dalcher then monitors execution flow of a function “utilizing the call stack frame from the freed memory.” (EX-1004 0022-0023.) That is, Dalcher’s monitoring is not limited to pending, not-yet-executed call frames, and its possible detection of unexpected software or an overwritten call frame may refer to a call frame that has already been executed. (EX-2001 ¶¶99.)

A POSITA would understand that such detection and quarantine is still useful—Dalcher explains that the system can be quarantined until “cleaned”—and it is possible that the malware may have infected other functions. (EX-1004 0026; EX-2001 ¶¶99.) However, a POSITA would not understand that Dalcher teaches preventing execution of a payload when behavior is flagged. (EX-2001 ¶¶98-99.)

Reading Dalcher’s disclosure of quarantining within the context of its specification confirms this understanding. For example, if Dalcher inherently prevented the execution of a payload (which it does not), it would be redundant to then place the system in quarantine; a POSITA would have understood that placing the system in quarantine is only necessary if execution of the payload cannot

confidently be prevented. (EX-2001 ¶98.) Indeed, because Dalcher disregards compromised stack frames when monitoring execution flow, Dalcher actually suggests that the remedial actions are not taken until analysis is complete—after the payload has been executed. (EX-2001 ¶98.)

In limitation 1.f, Petitioner also asserts that Pappas discloses a portion of this limitation—preventing execution—while not mentioning whether such prevention occurs when an alert is triggered. (Petition 25-26.) Petitioner then concludes by stating that “[a] [POSITA] would have looked to Pappas for further compatible details on preventing execution of quarantined software or functions when a compromise (suspicious behavior) is detected. Therefore, Dalcher and Pappas disclose this limitation.” (Petition 26.) But, as previously discussed, Petitioner does not provide sufficient specificity to determine *how* a POSITA would have modified Dalcher in view of Pappas or combined their teachings to practice this limitation. *See* Section VII.A, *supra*.

Furthermore, in addition to Petitioner's general discussion of motivation to combine being both insufficient and incorrect (Section VII.B, *supra*), Petitioner fails to explain *why* a POSITA would have combined these specific teachings from Dalcher and Pappas. Merely stating that a POSITA would have looked to Pappas for “compatible details” does not satisfy Petitioner's burden to explain “why a person

of ordinary skill in the art would have combined elements from specific references *in the way the claimed invention does.*" *ActiveVideo*, 694 F.3d at 1328; *Duo Security*, 2017 WL 4677238, at *11.

Such cursory assertions fail to meet Petitioner's burden to show what the combined teachings of the references would have suggested to a POSITA. *Securus Techs.*, IPR2015-00155, Paper 30 at 17. Accordingly, Petitioner fails to show that Dalcher, or a combination of Dalcher and Pappas, renders obvious preventing execution of a payload.

D. Petitioner Fails to Show a Reasonable Likelihood of Success as to Any of Challenged Dependent Claims

Regarding the dependent claims, because Petitioner fails to establish a reasonable likelihood of success as to any independent claim, Petitioner has failed to show a reasonable likelihood of prevailing with respect to any of the challenged claims. *Twinstrand Biosciences v. Guardant Health, Inc.*, IPR2022-01115, Paper 14 at 19 (PTAB Dec. 30, 2022).

VIII. OTHER LIMITATIONS 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.

**IX. THERE ARE SECONDARY CONSIDERATIONS OF
NON-OBVIOUSNESS**

In determining obviousness of the claimed invention, objective evidence of secondary considerations pertaining to nonobviousness must be considered, including long-felt but unsolved needs, failure of others, and commercial success. *In re Huang*, 100 F.3d 135, 139 (Fed. Cir. 1996) (citing *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966)); *Leo Pharm. Prods., Ltd. v. Rea*, 726 F.3d 1346, 1358 (Fed. Cir. 2013).

As discussed above, a long-felt and persistent need existed to improve detection and neutralization of malicious exploits—particularly where malware was successful in starting shellcode. (Section IV, *supra*.) Prior attempts to develop reliable anti-exploit technology came with heavy costs associated with employee time and resources, which did not sufficiently address advanced exploits. (*Id.*) Where others failed, the innovative approach claimed in the '591 patent succeeded. (Section V.A, *supra*.)

Patent Owner currently accuses at least four industry participants, including Petitioner, of infringing the innovative technology claimed in the '591 patent, demonstrating the critical need and widespread industry reliance on these novel

methods. (Paper 3 at 1.) Petitioner, for example, markets and provides Apex One, the components of which practice the systems and methods claimed in the '591 patent. (EX-2003 ¶¶209-236.)

Accordingly, secondary objective indicia of nonobviousness exist, such that Petitioner's grounds fail. (EX-2001 ¶¶100-102.)

X. CONCLUSION

For all the reasons stated herein, Petitioner has failed to establish that the challenged claims of the '591 Patent are unpatentable. Patent Owner respectfully requests the Board find the challenged claims patentable over the grounds in the Petition.

Date: August 11, 2023

Respectfully submitted,

KING & SPALDING LLP

BY: /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'S RESPONSE** specified in 37 C.F.R. § 42.24 have 13,986 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.2202.

Date: August 11, 2023

Respectfully submitted,

KING & SPALDING LLP

BY: /Mikaela Stone/
Mikaela Stone
Registration No. 70,404

*Attorney for Patent Owner
Open Text Inc.*

CERTIFICATE OF SERVICE

The undersigned hereby certifies that the foregoing **PATENT OWNER'S PRELIMINARY RESPONSE** and all supporting **EXHIBITS 2001 – 2025** were served by filing the documents through the Patent Trial and Appeal Case Tracking System (P-TACTS) on August 11, 2023 and courtesy copies were also served via electronic mail, in their entireties on the following:

Robert M. Hansen (Reg. No. 43,656) 1
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, Virginia 20191
Tel: (703)-391-2900
Fax: (703)-391-2900
Email: rhansen@marburylaw.com;
Email: samankwatia@marburylaw.com
Email: manderson@marburylaw.com; and
Email: pat-docketing@marburylaw.com.

*Counsel for Petitioner
Trend Micro, Inc.*

Date: August 11, 2023

KING & SPALDING LLP

BY: /Brian Eutermoser/
Brian Eutermoser
Registration No. 64,058

*Attorney for Patent Owner
Open Text Inc.*