IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
WACO DIVISION

WEBROOT, INC., OPENTEXT INC.,	**§**	
	§	
Plaintiffs,	**§**	
v.	**§**	
AO KASPERSKY LAB	**§**	**6:22-CV-00243-ADA-DTG**
TREND MICRO, INC.	**§**	**6:22-CV-00239-ADA-DTG**
SOPHOS LTD.	**§**	**6:22-CV-00240-ADA-DTG**
CROWDSTRIKE HOLDINGS, INC.	**§**	**6:22-CV-00241-ADA-DTG**
FORCEPOINT, LLC	**§**	**6:22-CV-00342-ADA-DTG**

PLAINTIFFS
RESPONSIVE CLAIM CONSTRUCTION BRIEF [86]

TABLE OF CONTENTS

I. LEGAL STANDARD 1
II. ’250, ’389, AND ’721 PATENTS 1
A. “similar objects” / “similar computer objects” (’250 patent, claims 1, 13; ’389 patent, claims 1, 15) 4
B. “behavior of the object” (’250 patent, claims 1-30) / “behavior of the computer object” (’250 patent, claims 1-12, 25-30); “behavior information about the first computer object” / “object behavior information (’721 patent, claims 1, 9, and 14) 8
C. “automatically” (’250 patent, claims 1, 13, and 25) 9
D. “a mask for the computer object that defines acceptable behavior for the computer object” / “a mask for an object ..., said mask defining acceptable behavior for the object” / “a mask for the computer object, the mask defining acceptable behaviour for the object” / “a mask for said computer object that defines acceptable behavior for the computer object” (’250 patent, claims 1, 13, and 25; ’389 patent, claim 14) 10
E. “running said object” / “executing the computer object” (’250 patent, claims 1-12 and 25-30) 14
F. “events initiated [by] or involving the [computer] object” (’250 patent, claims 2, 10, 14, 22, and 25-30; ’389 patent, claims 1-14, 26, and 30) 15
G. “first remote computer” and “second remote computer” (’389 patent, claim 1) 16
H. “correlating by the base computer, at least a portion of the data about the computer object received from the first remote computer to at least a portion of the data about the computer object received from the second remote computer; comparing, by the base computer, the correlated data about the computer object received from the first and second remote computers to other objects or entities to identify relationships between the correlated data and the other objects or entities” (’389 patent, claim 1) 16
I. “correlating, by the base computer, at least a portion of the data about the computer object received from the first remote computer to at least a portion of the data about the computer object received from the second remote computer” (’389 patent, claim 1) 18
J. “remote computer(s)” (’250 patent, claims 1, 2, 5, 7, 10, 12-14, 17, 19, 22, 24, 25, and 28; ’389 patent, claims 1, 6, 8, 9, 12, 14, and 30) 18
III. ’386 PATENT 22
A. “security appliance” (’386 patent, claims 1, 5, 7, and 11) 23
IV. ’045/’224 PATENTS 28
A. “global perspective” (’045 patent, claim 1; ’224 patent, claims 1, 10, and 17) 30
B. “wherein the global perspective for one or more related events to the at least one event across a network” (’045 patent, claim 1) 34
C. “event line” (’045 patent, claim 1; ’224 patent, claims 1, 10, and 17) 36
D. “gathering an event defining an action of a first object acting on a target…; generating contextual state information for the event by correlating the event to an originating object of the first object; obtaining a global perspective for the event based on the contextual state information, wherein the global perspective comprises information associated with one or more of the first object and the

originating object, and wherein the global perspective relates to one or more other events related to the event across a network;" ('045 patent, claims 1, 8, and 16; '224 patent, claims 1, 10, and 17) 37
E. "a sensor agent …" ('224 patent, claims 10, 14, 15, and 17) 40
V. '591 PATENT 43
A. "a component for evaluating the at least one monitored function in the memory space" ('591 patent, claims 1, 9, and 17) 44
B. "invoking one of the plurality of functions" ('591 patent, claims 1, 9, and 17) 47
C. "stack walk processing" ('591 patent, claims 1, 3, 5, 9, 11, 13, 17, and 18) 49
D. "before an address of an originating caller function is reached" ('591 patent, claims 1, 9, and 17) 52
E. "low level user mode function" ('591 patent, claims 3 and 11) 54
F. "at least one processor …" ('591 patent, claims 9-16) 57
G. "The system according to claim 9, further comprising identifying an address associated with the invoked function; and when the address of the invoked function is found in a cache, and wherein the processor further executing an operation of preventing execution of a payload for the invoked function" ('591 patent, claim 10) 59
VI. '844 PATENT 61
A. "an executable file" ('844 patent, claims 1 -20) 62
B. The terms relating to whether "character strings" are "within a predetermined range" ('844 patent, claims 1, 10, and 17) 65
C. "within a predetermined range" ('844 patent, claims 1, 10, and 17) 67
D. "support vector processing" ('844 patent, claims 1-20) 69
E. "at least one processor …" ('844 patent, claims 17-20) 70

TABLE OF AUTHORITIES

Cases:

3M Innovative Props. v. Avery Dennison Corp.,
350 F.3d 1373, 1374(Fed. Cir. 2003).....70

Akzo Nobel Coatings, Inc. v. Dow Chem. Co.,
811 F.3d 1334 (Fed. Cir. 2016).....6, 33, 35, 56, 59, 61

Almirall, LLC v. Amneal Pharm. LLC,
28 F.4th 265 (Fed. Cir. 2022) 51, 53

Anchor Wall Sys., Inc. v. Rockwood Retaining Walls, Inc.,
340 F.3d 1298 (Fed. Cir. 2003).....62, 69

Apex Inc. v. Raritan Computer, Inc.,
325 F.3d 1364 (Fed. Cir. 2003).....41

Apple Inc. v. Motorola, Inc.,
757 F.3d 1286 (Fed. Cir. 2014).....45, 71

Arlington Indus., Inc. v. Bridgeport Fittings, Inc.,
345 F.3d 1318 (Fed. Cir. 2003).....21

Baldwin Graphic Sys., Inc. v. Siebert, Inc.,
512 F.3d 1338 (Fed. Cir. 2008).....38

BASF Corp. v. Johnson Matthey Inc.,
875 F.3d 1360 (Fed. Cir. 2017).....55

Best Med. Int'l, Inc. v. Elekta Inc.,
46 F.4th 1346 (Fed. Cir. 2022) 25

Biogen Idec, Inc. v. GlaxoSmithKline LLC,
713 F.3d 1090 (Fed. Cir. 2013).....70

C.R. Bard, Inc. v. U.S. Surgical Corp.,
388 F.3d 858 (Fed. Cir. 2005).....14

Cellular Commc'ns Equip. LLC v. AT&T, Inc.,
No. 2:15-cv-576, 2016 WL 7364266 (E.D. Tex. Dec. 19, 2016) 57, 71

Collaborative Agreements, LLC v. Adobe Sys.,
No. 15-cv-03853-EMC, 2015 WL 7753293 (N.D. Cal. Dec. 2, 2015) 42

Cont'l. Circuits LLC v. Intel Corp.,
915 F.3d 788 (Fed. Cir. 2019).....50

CoreLogic Information Solutions, Inc. v. Fiserv, Inc.,
2012 WL 4355394 (E.D. Tex. 2012).....45

Curtiss-Wright Flow Control Corp. v. Velan, Inc.,
438 F.3d 1374 (Fed. Cir. 2006).....12

CXT Sys., Inc. v. Acad., Ltd.,
No. 2:18-CV-171, 2019 WL 4253841 (E.D. Tex. Sept. 6, 2019).....45

Deere & Co. v. Bush Hog, LLC,
703 F.3d 1349 (Fed. Cir. 2012).....12, 15, 21, 50, 53

Digital Generation, Inc. v. Boring,
869 F. Supp. 2d 761 (N.D. Tex. 2012).....7

Dyfan, LLC v. Target Corp.,
28 F.4th 1360 (Fed. Cir. 2022).....45, 57, 58, 71

EI du Pont de Nemours & Co. v. Phillips Petro. Co.,
849 F. 2d 1430 (Fed. Cir. 1988).....21

Enzo Biochem, Inc. v. Applera Corp.,
599 F.3d 1325 (Fed. Cir. 2010).....31

Finjan, Inc. v. Proofpoint, Inc.,
No. 13-cv-05808-HSG, 2015 WL 7770208 (N.D. Cal. Dec. 3, 2015).....42

Fisher-Rosemount Sys., Inc. v. ABB Ltd.,
No. 4:18-cv-178, 2019 WL 6830806 (S.D. Tex. Dec. 12, 2019).....57, 58, 71

Flypsi, Inc. v. Dialpad, Inc.,
6:21-CV-00642-ADA, 2022 WL 3593131 (W.D. Tex. Aug. 22, 2022).....48, 51, 53

Freeny v. Murphy USA Inc.,
No. 2:13-CV-791-RSP, 2015 WL 294102 (E.D. Tex. Jan. 21, 2015).....6, 33, 35, 56, 59. 61

Gart v. Logitech, Inc.,
254 F.3d 1334 (Fed. Cir. 2001).....21

Genband USA LLC v. Metaswitch Networks Ltd.,
No. 2:14-cv-33, 2015 WL 4722185 (E.D. Tex. Aug. 6, 2015).....41

Huawei Techs. Co. v. Verizon Commc'ns, Inc.,
No. 2:20-CV-30, 2021 WL 150442 (E.D. Tex. Jan. 15, 2021).....45

Intelligent Agency, LLC v. 7-Eleven, Inc.,
2022 Markman 760203, 2022 WL 760203 (E.D. Tex. 2022).....42

Interactive Gift Express, Inc. v. Compuserve Inc.,
256 F.3d 1323 (Fed. Cir. 2001).....16, 38

Interval Licensing LLC v. AOL, Inc.,
766 F.3d 1364 (Fed. Cir. 2014).....31

IQASR LLC v. Wendt Corp.,
825 F. App'x 900 (Fed. Cir. 2020) 33

Leader Technologies, Inc. v. Facebook, Inc.,
2010 WL 809848 (D. Del. 2010) 45

Liebel-Flarsheim Co. v. Medrad, Inc.,
358 F.3d 898 (Fed. Cir. 2004).....25, 48, 68

Linear Tech. Corp. v. Int'l Trade Comm'n,
566 F.3d 1049 (Fed. Cir. 2009).....26

Luminati Networks Ltd. v. BIScience Inc.,
No. 2:18-CV-00483-JRG, 2019 WL 6683268 (E.D. Tex. Dec. 6, 2019).....6, 33, 35, 56, 59, 61

Mastermine Software, Inc. v. Microsoft Corp.,
874 F.3d 1307 (Fed. Cir. 2017).....39, 59, 61

Media Rights Techs., Inc. v. Capital One Fin. Corp.,
800 F.3d 1366 (Fed. Cir. 2015).....70

Microsoft Corporation v. Uniloc 2017 LLC,
2022 WL 11531235 (Fed. Cir., Oct. 20, 2022)..... 16-17, 38

Niazi Licensing Corp. v. St. Jude Med. S.C., Inc.,
30 F.4th 1339 (Fed. Cir. 2022) 30, 31, 32

One-E-Way, Inc. v. ITC,
859 F.3d 1059 (Fed. Cir. 2017).....55

Phillips v. AWH Corp.,
415 F.3d 1303 (Fed. Cir. 2005)..... *passim*

R2 Sols. LLC v. Deezer S.A.,
No. 4:21-CV-122, 2022 WL 36240 (E.D. Tex. Jan. 4, 2022).....30, 31, 33

Realtime Data, LLC, v. Actian Corp. et al.,
No. 6:15-cv-463-RWS-JDL (E.D. Tex. Nov. 1, 2016).....5

Rembrandt Data Techs., LP v. AOL, LLC,
641 F.3d 1331 (Fed. Cir. 2011).....59, 61

Respironics, Inc. v. Invacare Corp.,
303 Fed.Appx. 865 (Fed. Cir. 2008).....17, 39

Salazar v. Procter & Gamble Co.,
414 F.3d 1342 (Fed. Cir. 2005).....70

Seachange Int'l, Inc. v. C-COR, Inc.,
413 F.3d 1361 (Fed. Cir. 2005).....64

Skedco, Inc. v. Strategic Operations, Inc.,
685 F. App'x 956 (Fed. Cir. 2017).....26

Sonix Tech. Co. v. Publ'ns Int'l, Ltd.,
844 F.3d 1370 (Fed. Cir. 2017).....55

SourceProse Corp. v. AT&T Mobility,
Case No. A-11-CV-117-LY, slip op. (W.D. Tex. June 24, 2014).....9

Syncpoint Imaging, LLC v. Nintendo of Am. Inc.,
No. 2:15-cv-00247-JRG-RSP, 2016 WL 55118 (E.D. Tex. Jan. 5, 2016).....43

Thorner v. Sony Comput. Ent. Am. LLC,
669 F.3d 1362 (Fed. Cir. 2012).....13, 51

TracBeam, LLC v. Apple, Inc.,
No. 6:14-cv-680, 2015 WL 5786449 (E.D. Tex. Sept. 29, 2015).....6, 33, 56, 59, 61

Typemock, Ltd. v. Telerik, Inc.,
No. 17-cv-10274, 2018 WL 4189692 (D. Mass Aug. 31, 2018).....57, 71

Uniloc 2017 LLC v. Google LLC,
No. 2:18-CV-00492-JRG-RSP, 2020 WL 569858 (E.D. Tex. Feb. 5, 2020).....5

United States v. Jackson,
426 F.3d 301 (5th Cir. 2005).....6, 33, 56, 59, 61

VDPP LLC v. Vizio, Inc.,
2021-2040, 2022 WL 885771 (Fed. Cir. Mar. 25, 2022) 45, 58

VERVE LLC v. Crane Cams,
311 F.3d 1116 (Fed. Cir 2002).....5

Widevine Technologies, Inc. v. Verimatrix, Inc.,
2009 WL 3734106, *16 (E.D. Tex. 2009) 45

Wi-LAN, Inc. v. Apple, Inc.,
811 F.3d 455 (Fed. Cir. 2016).....38

Zeroclick, LLC v. Apple Inc.,
891 F.3d 1003 (Fed. Cir. 2018).....41, 45, 57, 58

TABLE OF EXHIBITS

Exhibit 1	U.S. Patent 8,418,250
Exhibit 2	U.S. Patent 8,725,389
Exhibit 3	U.S. Patent 9,413,721
Exhibit 4	Declaration of Dr. Nenad Medvidovic
Exhibit 5	GB Priority Appl.
Exhibit 6	U.S. Patent 8,438,386
Exhibit 7	Declaration of Sam Malek Ph.D.
Exhibit 8	U.S. Patent 9,578,045
Exhibit 9	U.S. Patent 10,257,224
Exhibit 10	U.S. Patent 9,578,045 File History Excerpts
Exhibit 11	U.S. Patent 10,257,224 File History Excerpts
Exhibit 12	*Intentionally Left Blank*
Exhibit 13	U.S. Patent 10,284,591
Exhibit 14	Trend Micro 10/7/2022 Proposed Claim Constructions
Exhibit 15	U.S. Patent 10,284,591 File History Excerpts
Exhibit 16	U.S. Patent 10,599,844
Exhibit 17	U.S. Patent 10,599,844 File History Excerpts

Plaintiffs Webroot, Inc. and Open Text, Inc. ("Plaintiffs" or "Open Text") respond to AO Kaspersky Lab, Trend Micro, Inc., Sophos Ltd., CrowdStrike, Inc., CrowdStrike Holdings, Inc., and ForcePoint, LLC's (collectively "Defendants") Opening Claim Construction Brief. Defendants variously attempt to propose constructions that depart from the plain and ordinary meaning and improperly import limitations from the specification or prosecution history. None of the terms that defendants identify are limited by any clear and unequivocal disclaimer or definition found in the intrinsic record. The claim terms that Defendants assert to be indefinite are all well understood with reasonable certainty by a person having ordinary skill in the art from the patent disclosures.

I. LEGAL STANDARD

Per the Court's OGP, "[t]he Court has familiarity with the law of claim construction and encourages the parties to forego lengthy recitations of legal authorities and to instead focus on the substantive issues unique to each case." OGP 4.2, 7. Accordingly, Plaintiffs provide any substantive legal authority to the extent relevant to specific claim construction issues presented.

II. '250, '389, AND '721 PATENTS

The '250 and '389 Patents (referred to here as the "Advanced Malware Detection Patents") are part of the same patent family and generally disclose and claim systems and processes related to real-time and advanced classification techniques for as-yet unknown malware. The '721 Patent is not related to the '250 or '389 Patents and was filed nearly six years later. However, Defendants have improperly grouped terms from this unrelated patent with the '250 and '389 Patents. Plaintiffs will therefore also address these terms at the same time.

<u>The '250 and '389 Patent:</u> Malware detection systems in use at the time the Advanced Malware Detection Patents were filed identified malware by maintaining a database of signatures

identifying known bad objects (*i.e.*, malware). The signature for an object was conventionally made by creating a hash or checksum corresponding to the object file, which uniquely identifies that object. The signature of each object was then compared to the database to look up whether it matches known malware. If the signature of the object is not found in the database, it is assumed safe or alternatively, the whole file is sent for further investigation by a human analyst. The process of further investigation was typically carried out by subjecting the file to detailed analysis, for example by emulation or interpretation, which can take days given the human analysis that is typically required. (*See, e.g.*, Exhibit 2, '389 Patent, 2:9-17.)

This approach had significant drawbacks, including that it required considerable effort by the providers of such systems to identify and analyze new malware and generate signatures of objects that are found to be bad after human analysis. Large vendors of anti-malware packages typically employed thousands of human analysts to identify and analyze objects and keep the database of signatures of bad objects reasonably up to date. However, as the volume of network traffic increases, the task of keeping up with identifying suspect objects and investigating whether or not they are bad becomes practically impossible. (*Id.*) It can take days to subject a suspicious file to detailed analysis given the human analysis, and a considerable period of time elapses before a new file is classified as safe or as malware. Thus, the human analysis introduces a time delay where users are exposed and unprotected from the risks posed by previously unidentified malware. (*See* Exhibit 2, '389 Patent, 2:9-23, 2:63-67.)

By contrast, the methods and systems disclosed and claimed in the '250 and '389 Patents perform sophisticated review of the actual attributes of a software object or process and the behavior engaged in by, or associated with, that object or process on computers connected to a network. This review enables a determination of "the nature of the object," (*e,g.*, whether it is

malicious or not based on review of the object, its behaviors, or the activities associated with the object), without requiring a detailed manual analysis of the code of the object itself, or relying exclusively on whether it has a signature that matches an extensive database of known malicious "signatures." (*See* Exhibit 2, '389 Patent, 3:14-24; Exhibit 1, '250 Patent, 3:7-18.)

<u>The '721 Patent:</u> The systems and methods described and claimed in the '721 Patent are directed to improved techniques for detecting and classifying malware using network architectures. The '721 Patent explains that prior anti-malware products used signature matching to detect malware, either locally or at a central server. (Exhibit 3, '721 Patent, 1:37-2:14.) The local anti-malware product suffered from delays in identifying new malware threats and obtaining signatures for them so they could be blocked. (*Id.* at 1:37-55.) Central servers stored signatures in the cloud. (*Id.* at 56-57.) But only signature or very basic information was sent to the central server for matching. (*Id.* at 1:67-2:2.) If the object was unknown, a copy had to be sent to the central server for investigation by a human, a time consuming and laborious task. (*Id.* at 2:5-7.) In a network environment, it was unrealistic for a human to investigate each new object due to the high volume of incursions that take place over a network. (*Id.* at 2:7-10.) Thus, under these approaches, "malevolent objects may escape investigation and detection for considerable periods of time." (*Id.* at 2:10-13.) To address these shortcomings, the '721 Patent describes and claims unconventional, novel distributed system architectures, such as

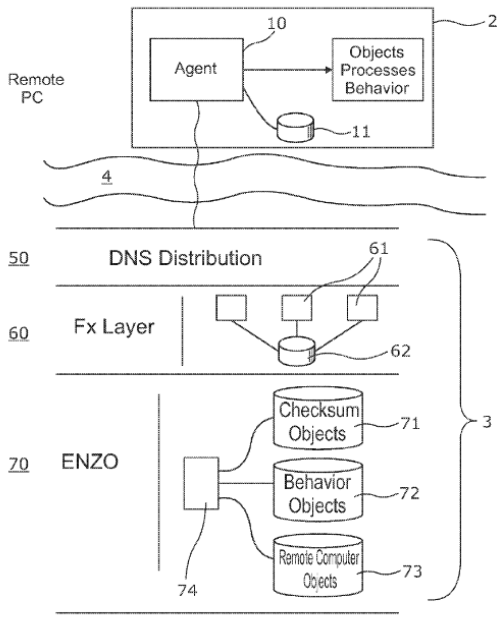


Fig. 3

remote computers that may be allocated to "threat" servers, with "central" servers sitting behind them. (Exhibit 3, '721 Patent, 9:16-57.) In particular, embodiments may include three-tiered architectures of remote computers, threat servers, and a central server that provides a technical enhancement to the computer network itself (improving upon the two-tiered architectures of traditional systems having only remote computers and a central server) by enabling the central server to keep a master list "of all data objects, their metadata and behaviour seen on all of the remote computers" and propagate it back to the threat servers. (Exhibit 3, '721 Patent 12:28-54.) The '721 Patent explained that this "scheme has been found to reduce workload and traffic in the network by a factor of about 50 compared with a conventional scheme." (*Id*. at 12:55-57.)

A. "similar objects" / "similar computer objects" ('250 patent, claims 1, 13; '389 patent, claims 1, 15)

Plaintiffs	Sophos/CrowdStrike/Trend Micro/ForcePoint/Kaspersky
Plain and ordinary meaning	Indefinite

Defendants wrongly contend that the claim terms "similar objects" and "similar computer objects" are indefinite because "the patent specification[s] [do] not provide any way to discern what qualifies as 'similar.'" ECF 86 at 2-3. As discussed below and in the accompanying Declaration of Dr. Nenad Medvidovic, Defendants' arguments are facially deficient, and Defendants have wholly failed to prove by clear and convincing evidence that these terms would not be understood with reasonable certainty by a person having ordinary skill in the art. *Microsoft Corp. v. i4i Ltd. Partnership*, 131 S.Ct. 2238, 2241 (2011) (holding that "[t]he burden of establishing invalidity of a patent or any claim thereof shall rest on the party asserting such invalidity … [and] requires an invalidity defense to be proved by clear and convincing evidence.")

At the outset, the '250 and '389 Patents are directed to a specific technological field: cyber security and malware detection. The claims of the '250 and '389 Patents are generally directed to methods of classifying a computer object as malware. Broadly speaking, the claimed methods

involve receiving, at a base computer, data about the computer object from remote computers on which the object and "similar objects" (in the case of the '250 Patent) or "similar computer objects" (in the case of the '389 Patent) are stored. The data about the computer object received by the base computer is used to determine whether the computer object is malware. In other words, these systems collect and leverage data about computer objects from multiple sources and use that information to identify malicious objects. (Ex. 4, Medvidovic Decl., ¶¶37, 38.)

Defendants ignore the context of the patents and make general allegations that the terms "similar objects" and "similar computer objects" are indefinite based on the use of the word "similar" alone because it is a term of degree. However, the use of a term of degree, alone, is common in patent claims and does not render a claim indefinite. Courts often find the term "similar" or like terms sufficiently definite in their recited context. *See*, *e.g.*, *VERVE LLC v. Crane Cams*, 311 F.3d 1116, 1120 (Fed. Cir 2002) (holding that the term of degree "substantially" in the phrase "substantially constant wall thickness" does not of itself render the claims indefinite); *Uniloc 2017 LLC v. Google LLC*, No. 2:18-CV-00492-JRG-RSP, 2020 WL 569858, at *12 (E.D. Tex. Feb. 5, 2020) (holding that the term "similar vector" in the context of computerized video processing was not indefinite); *Realtime Data, LLC, v. Actian Corp*. et al., No. 6:15-cv-463-RWS-JDL, at *4 (E.D. Tex. Nov. 1, 2016) (stating that "the term 'similar' must be construed specifically…as it would be understood by a person of ordinary skill in the art"). The terms are thus not analyzed in a vacuum, but instead are considered in their appropriate context and from the perspective of a person having ordinary skill in the art ("POSITA").

Significantly, Defendants offer ***no*** expert testimony concerning what a POSITA would understand from reading the specifications of the '250 and '389 Patents concerning the scope of

these terms.[1] In fact, Defendants have failed to provide any definition of a POSITA in terms of the relevant levels of knowledge, education, and experience. (Exhibit 4, Medvidovic Decl., ¶¶31-33, 43.) Thus, Defendants' indefiniteness arguments are facially deficient because there is no foundation for any of their arguments regarding what a POSITA would have understood (or not) based on reading the patents and their prosecution histories in view of the knowledge of a POSITA.

These deficiencies are fatal to Defendants' claims. Courts have found that a party has not met its burden of proving indefiniteness where it "does not provide or rely upon any expert for its proposition that the term is indefinite or that one of ordinary skill in the art would not be reasonably certain as to the scope of [the] claim element." *Freeny v. Murphy USA Inc.*, No. 2:13-CV-791-RSP, 2015 WL 294102, at *14 (E.D. Tex. Jan. 21, 2015); *see also Luminati Networks Ltd. v. BIScience Inc.*, No. 2:18-CV-00483-JRG, 2019 WL 6683268, at *9 (E.D. Tex. Dec. 6, 2019) (finding no indefiniteness where party "did not provide expert testimony on the issue and thus cannot meet the clear-and-convincing standard" of proving the claim indefinite); *Akzo Nobel Coatings, Inc. v. Dow Chem. Co.*, 811 F.3d 1334, 1345 (Fed. Cir. 2016) (concluding that claims were not indefinite after party provided "no evidence to show that a person of ordinary skill in the art would not know with reasonable certainty the steps to which the limitation in [the claim] applies."). Moreover, Defendants cannot attempt to prove indefiniteness by clear and convincing evidence by belatedly offering evidence in a reply brief. *See, e.g.*, *United States v. Jackson*, 426 F.3d 301, 304 n.2 (5th Cir. 2005) ("arguments raised for the first time in a reply brief are waived"); *TracBeam, LLC v. Apple, Inc.*, No. 6:14-cv-680, 2015 WL 5786449, at *2 (E.D. Tex. Sept. 29, 2015) (courts disregard new evidence or argument offered for the first time in the reply brief*);*

[1] Indefiniteness is evaluated through the eyes of a skilled artisan having read the specification and prosecution history. *Nautilus*, 134 S. Ct. at 2124.

Digital Generation, Inc. v. Boring, 869 F. Supp. 2d 761, 771 (N.D. Tex. 2012) (stating that "the court exercises its discretion to not consider the new arguments and evidence to avoid prejudice").

Defendants' failure to present expert evidence here is understandable because object similarity was a well-known and widely used concept in the malware detection and cyber security field. Indeed, it is essentially a necessity—these systems operate in environments where malicious objects can easily be modified (either by malicious actors or on their own in the case of polymorphic code) that would have rendered even the most sophisticated systems ineffective if they could only identify identical objects, while any malicious objects with minor changes would pass undetected. For at least this reason, object similarity was extensively studied in the art even before the June 30, 2005 filing of the priority document of '250 and '389 Patents, and is a concept that is widely used today. (Exhibit 4, Medvidovic Decl., ¶¶44, 45) Thus, the skilled artisan would have known and understood ways that objects and computer objects could be considered "similar" in the June 2005 time frame, and would have had no difficulty determining if objects are similar in the appropriate context. (Exhibit 4, Medvidovic Decl, ¶¶ 44, 45)

Defendants are also wrong that the specification provides no guidance on object similarity. The patents disclose, for example, that data can be sent to the base computer in the form of a "key" that contains information about the executable instructions contained within a given computer object. (Exhibit 1, '250 Patent, 3:42-44; Exhibit 2, '389 Patent, 3:57-59.) By comparing the keys of two different computer objects, the base computer can determine, in these embodiments, whether two differently named computer objects have executable instructions in common. (Exhibit 1, '250 Patent, 3:55-59; Exhibit 2, '389 Patent, 3:61-65.) Further, the patents disclose that "if at least one other object to which said object is related is classed as malware, then the method may comprise classifying said object as malware." (Exhibit 1, '250 Patent, 3:35-37, Exhibit 2, '389

Patent, 41-43.) This passage clearly requires no specific threshold degree or level of similarity to classify a given object as malware by referring to a related object. A POSITA would understand that two objects or computer objects can be "similar objects" or "similar computer objects" without necessarily specifying a threshold degree or level of similarity.

In view of the foregoing, Defendants have failed to prove that the terms "similar objects" and "similar computer objects" are indefinite.

B. "behavior of the object" ('250 patent, claims 1-30) / "behavior of the computer object" ('250 patent, claims 1-12, 25-30); "behavior information about the first computer object" / "object behavior information ('721 patent, claims 1, 9, and 14)

Plaintiffs	CrowdStrike/Sophos
Plain and ordinary meaning	"actions performed by the [object / computer object / first computer object]"

Only CrowdStrike and Sophos have offered a construction for these terms. Thus, Plaintiffs and three Defendants agree that these terms do not need a construction. Defendants' construction should be rejected as it is inconsistent with the plain and ordinary meaning of the proposed claim language. For example, Defendants propose that "behavior," "behavior information" and "object behavior information" all mean "actions performed by." The plain meaning of "behavior information" is not an "action" at all, it is information. Moreover, the plain meaning of "behavior" is broader than actions performed by an object. For example, it would include reactions—such as how an object changes to its environment or actions performed on it. And, of course, in the field of computer science, a POSITA would understand that the behavior of an object may be ascertained from the characteristics or features of the object itself, such as its source code, without the action actually being performed.

Indeed, the specification makes clear that "behavior" is not limited to "actions performed by" an object. For example, the specification explains that the "behaviors" detected include

"certain behaviour and characteristics." (Ex. 1 at 11:42-44.) The specification further discloses that behaviors can include actions performed ON the object—the opposite of Defendants' proposed construction. (*Id.* at 12:14-30 ("The fourth said process considers the behaviour of an object. This allows a score to be developed based on the types of events performed by an object ***or events performed on it by itself or other object***.") Defendants even affirmatively cite a portion of the specification that makes clear "actions" and "behaviors" are not interchangeable: "events" are the "actions or behaviors of an object acting upon another object…" (ECF 86 at 14 (citing '250 Patent at 12:64-13:15).) Under Defendants' proposed construction, this passage would nonsensically read: "events" as the "actions or [actions] of an object acting upon another object."

C. "automatically" ('250 patent, claims 1, 13, and 25)

Plaintiffs	CrowdStrike/Sophos/Trend Micro/Kaspersky
Plain and ordinary meaning	"without human involvement"

Automatically means automatically. It is a well-known term with a commonly understood definition and needs no construction. *See, e.g.*, Claim Construction Order, *SourceProse Corp. v. AT&T Mobility*, Case No. A-11-CV-117-LY, slip op. at 22-23 (W.D. Tex. June 24, 2014) ("The court also disagrees with Defendants that substitution 'without user interaction' for 'automatically' advances the goals of claim construction or makes the claim language more easily understood by a jury. The court does not find that the word 'automatically' is in need of additional clarification or that it is used in the context of the patent any differently than its plain and ordinary meaning."). Defendants do not appear to argue that Plaintiffs redefined the term "automatically." Instead, they argue that the "use of this term in the specification is entirely ***consistent*** with this definition." (ECF 86 at 7.) But that is not the standard for departing from the plain and ordinary meaning. *Golden Bridge*, 758 F.3d at 1365. And the specification also uses the term automatically to describe an action that necessarily and inevitably occurs as a result of a particular circumstance. (Exhibit 1,

’250 patent, 17:36-39 (“[b]y monitoring activity of the parent process, if it is later found that the parent process is malware, then all of the issue processes can all *automatically* be recorded in the local database as bad.”)

The prosecution history does not require “automatically” to be defined as “without human involvement.” Defendants identify a statement made during prosecution that referred to the entire phrase “automatically generating a mask for the computer object” in contrast to a manual malware identification process performed entirely by the administrator, including classifying the software. *See* ECF 86 at 11 (citing Response to Feb. 15, 2011 OA at 15-18 (internal citations omitted). But it is a bridge too far to say that automatically must be without ***any*** human involvement at all. Accordingly, “automatically” should be accorded its plain and ordinary meaning.

D. “a mask for the computer object that defines acceptable behavior for the computer object” / “a mask for an object ..., said mask defining acceptable behavior for the object” / “a mask for the computer object, the mask defining acceptable behaviour for the object” / “a mask for said computer object that defines acceptable behavior for the computer object” (’250 patent, claims 1, 13, and 25; ’389 patent, claim 14)

Plaintiffs	CrowdStrike/Sophos/ Trend Micro/ForcePoint
Plain and ordinary meaning	“a list of acceptable actions for the [object/computer object] to perform”

Plaintiffs and Kaspersky appear to agree that these terms do not need a construction. The remaining Defendants’ proposed construction of the “mask” language in the ’250 and ’389 patent claims should be rejected because, as with other terms, it departs from the plain and ordinary meaning of the claim language. For example, claim 1 of the ’250 patent, recites in relevant part:

> automatically generating a mask for the computer object that defines acceptable behaviour for the computer object, wherein the mask is generated in accordance with normal behaviour of the object determined from said received data;
>
> …
>
> allowing the computer object to continue to run when behavior of the object is permitted by the mask; [and]

> disallowing the computer object to run when the actual monitored behaviour of the computer object extends beyond that permitted by the mask ….

Reviewing claim 1, a POSITA would understand that the mask "define[s] acceptable behaviour for [a] computer object" to determine whether to "continue" or "disallow" the object to continue to run when its behavior falls within or extends beyond that permitted by the mask. In other words, the mask establishes the permissible and impermissible behavior of an object.

There is nothing in Defendants' proposed construction that is consistent with the plain and ordinary meaning. First, the plain and ordinary meaning of the term "mask" as used in the claims is not synonymous with a "list." Defendants cite nothing to support their statement that "[c]entral to the disclosure of the '250 patent is the concept an automatically generated 'mask' that ***lists***." (ECF 86 at 9.) Instead, the claims make clear that the mask "***defines*** acceptable behavior." In their argument for the term "automatically," Defendants cited a portion of the prosecution history that provides an example of a mask as "rules" (not a "list"): "[a]dditionally, Lambert teaches that an administrator classifies the software using the criteria and ***sets the rules*** for the token. *See* Lambert page 6 par. [0052]-[0054]. This is clearly manual, and not automatic generation ***of a mask."*** (ECF 86 at 8 (citing Ex. A-1 ('250 FH) at 2-3.) Thus, Defendants implicitly recognize that a mask is not limited to a "list of acceptable actions."

Indeed, a POSITA would have known that there is more than one way a mask can "define" acceptable behavior.[2] For example, while acceptable behavior could certainly be defined in part by "a list" of acceptable behaviors, it can also be defined by a list of unacceptable behaviors. A simple example is illustrative: acceptable behaviors for a child may be defined by listing things

[2] As another example, rather than a "list" of behaviors, a mask could also define acceptable behaviors by, for example the characteristics of the behaviors, rather than listing the behaviors themselves.

they are permitted to do, such as play tag on the lawn, shoot baskets in the driveway, or ride bikes on the sidewalk. Or the acceptable behaviors could be defined by what the child is not permitted to do, such as playing in traffic or with fire. The latter "defines acceptable behavior" as activities that do not include the unpermitted behavior. As another example, airport security does not list all of the items that can be brought onto a plane; they identify the items that are prohibited. Anything not listed is therefore permitted. Indeed, in some circumstances, it may be more efficient to define acceptable behaviors by what is unacceptable.

Second, Defendants' proposed construction again attempts to construe the "behavior for the object" portion of this language as limited to "actions" the object performs ("actions for the object/computer object to perform"). As discussed above, the plain and ordinary meaning of "behavior" of an object encompasses, but is not limited to, actions performed by an object.

Third, Defendants' proposed construction improperly attempts to read limitations from the specification into the claims. *Deere,* 703 F.3d at 1354 (Fed. Cir. 2012) But even then, the specification does not support Defendants' construction. For example, Defendants cite to examples in the specification where the "mask" that is generated identifies "expected behavior of the object" or is generated "in accordance with normal behavior of the object" (ECF 86 at 10-11) but none of these embodiments require the "mask" to refer solely to a list of acceptable actions or disclaim masks that define acceptable behaviors in other ways.

Defendants also point to dependent claim 12, but that claim is entirely consistent with the plain and ordinary which encompasses any suitable way to define acceptable behavior. As an initial matter, dependent claims presumptively have a narrower scope than the independent claims from which they depend. *Curtiss-Wright Flow Control Corp. v. Velan, Inc*., 438 F.3d 1374, 1380 (Fed. Cir. 2006) (stating that "the statute stresses that a dependent claim must add a limitation to those

recited in the independent claim."). Under Defendants' theory, dependent claim 12 supports their position because it claims a specific embodiment in which a mask cannot include a list of non-permitted behaviors. But this language stands in contrast to the claim 1 language which does not include this limiting language and indeed, if anything, broader than the mask of the dependent claim.

Defendants further misplace reliance on the prosecution history, pointing to evidence that does not disclaim or modify the plain and ordinary meaning of the "mask" claim language and even confirms the mask is not limited to a list of acceptable behaviors. Defendants' argue that "[s]ince the data are collected by monitoring the normal behavior of the object, the mask ***must*** be generated as a list of those normal actions." ECF 86 at 11 (citing Response to Examiner's Feb. 15, 2011 Non-Final Rejection at 14). That is not true, as explained above. This passage, at most, indicates that the mask is "generated *based on* normal behavior of the object," but that does not dictate how the mask is ***defined***. *See Thorner*, 669 F.3d at 1365.

Finally, Defendants argue, without basis in the intrinsic record, that interpreting the mask to be a list of unacceptable behaviours would "defy the very point of the alleged invention, which is to monitor the behavior of the object to determine if the object deviates from normal behavior," and "would be superfluous, because there is no need to monitor the behavior of an object that is already known to be malware." ECF 86 at 9-10. Defendants, all well-known cyber security companies, appear to be suggesting that these malware detection processes only monitor objects when it is already known whether the behavior of an object is or is not malware. These processes monitor objects where it is ***not known*** whether they contain malware or not. Defendants' claim that the only way to monitor whether an object to "deviates from normal behavior" is to have a list of normal behaviors defies both logic and common sense. As an example, an object can also be

monitored to see whether it "deviates from normal behavior" by comparing the behavior of the object to identified behaviors that are already known to be abnormal.

Accordingly, Defendants' proposed construction of the "mask" claim language should be rejected. A jury is perfectly capable of understanding how a mask may define acceptable behavior.

E. "running said object" / "executing the computer object" ('250 patent, claims 1-12 and 25-30)

Plaintiffs	CrowdStrike
Plain and ordinary meaning	"carrying out the object's computer code"

CrowdStrike has replaced two-different, easily understood terms with a single, less-clear multi-word phrase under the guise that it would benefit the jury and has resorted to "less significant" extrinsic, dictionary evidence for support. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1317 (Fed. Cir. 2005) (quoting *C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 862 (Fed. Cir. 2005)). "Running said object" and "executing the computer object" are clear from the claims and the specification and need not be construed. CrowdStrike even agrees that "[t]here should be no disagreement over the technical meanings of these terms." ECF 86 at 12.

Without support or explanation, CrowdStrike offers a construction that departs from the plain and ordinary meaning of both terms. The claim language refers to the "object" itself, not the object's code. The various dictionary definitions that CrowdStrike cites in a table as support actually define these terms in different ways and differently from its proposed construction. (*See* ECF 86 at 13.)

The terms "running said object" and "executing the computer object" are simple, easily understood terms, and CrowdStrike cites to no disavowal or definition in the specification that redefines these terms or construes them to be exactly the same. CrowdStrike also believes that a jury will not appreciate the difference between the common and technical meanings of "run" and "execute," but Plaintiffs are confident that jurors in Waco will readily understand that "running

said object," does not mean training it for a footrace and that "executing the computer object" does not mean carrying out its death sentence.

F. "events initiated [by] or involving the [computer] object" ('250 patent, claims 2, 10, 14, 22, and 25-30; '389 patent, claims 1-14, 26, and 30)

Plaintiffs	CrowdStrike/ForcePoint
Plain and ordinary meaning	"actions performed by or upon the object"

Two Defendants propose a construction that departs from the plain and ordinary meaning of the claim language in several respects. First, the claim language refers to "events" not "actions." As explained above, the '250 and '389 specifications are clear that "events" include both "actions" and "behaviors": "'Events': these define the actions or behaviours of an object acting upon another object or some other entity." (Exhibit 1, '250 Patent, 12:64-65.) Second, the claim language refers to events "initiated by" a computer object. An object does not need to actually perform an entire action to initiate an event. Third, the claim language refers to events "involving" the computer object. An event can ***involve*** a computer object indirectly, without any action being performed directly by or upon an object, such as actions performed by a child object may involve the parent object simply because the parent created the child, even though the specific action was not taken by or on the parent object. (Exhibit 1, '250 patent, 11:3-41).

Further, Defendants' proposed construction is inconsistent with the '250 specification, which does not define "events" as limited to actions performed by or upon an object. For example, the specification describes events that establish other relationships between objects—such as a parent–child relationship. (*See, e.g.*, Exhibit 1, '250 patent, 11:3-15.) The specification further explains that "events initiated by or involving the object when the object is created, configured or runs" serves to help "identif[y] relationships between the object and other objects." (*Id.* at 3:28-29, 3:31-32.) At bottom, Defendants' narrow construction is a classic improper attempt to import limitations from the specification into the claims. *Deere & Co. v. Bush Hog*, LLC, 703 F.3d 1349,

1354 (Fed. Cir. 2012) (*citing Phillips*, 415 F.3d at 1323) ("While claim terms are understood in light of the specification, a claim construction must not import limitations from the specification into the claims."

G. "first remote computer" and "second remote computer" ('389 patent, claim 1)

Plaintiffs	CrowdStrike
Plain and ordinary meaning	The "first" and "second" remote computers are different computers

Plaintiffs contend that the term "remote computer" should be given its plain and ordinary meaning as set forth below and do not agree with Defendants' narrow construction of that term, which departs from the plain and ordinary meaning. With that understanding, Plaintiffs agree that the "first remote computer" in '389 patent, claim 1 is not the identically same computer as the "second remote computer."

H. "correlating by the base computer, at least a portion of the data about the computer object received from the first remote computer to at least a portion of the data about the computer object received from the second remote computer; comparing, by the base computer, the correlated data about the computer object received from the first and second remote computers to other objects or entities to identify relationships between the correlated data and the other objects or entities" ('389 patent, claim 1)

Plaintiffs	Sophos/ForcePoint
Plain and ordinary meaning	Order of Steps: comparing must follow correlating

Two of the Defendants propose a construction that reads the claim language too narrowly. The law is clear that "[u]nless the steps of a method actually recite an order, the steps are not ordinarily construed to require one." *Interactive Gift Express, Inc. v. Compuserve Inc.*, 256 F.3d 1323, 1342-43 (Fed. Cir. 2001) (holding that step one of a method claim can be performed after step four, even though step four referred to "the catalog code" introduced in step one). Courts have consistently held that claim language including reference back to previous steps for input is not dispositive. *See, e.g., Microsoft Corporation v. Uniloc 2017 LLC*, 2022 WL 11531235, at *3 (Fed.

Cir., Oct. 20, 2022) (holding that step of "comparing *the determined component*" does not have to after the step of "*determining at least one device component*") (emphasis added); *Respironics, Inc. v. Invacare Corp.*, 303 Fed.Appx. 865, 870 (Fed. Cir. 2008) ("[I]n a method claim, a step that recites 'said' or 'the,' referring to an earlier object, does not always have to be performed after the step that first introduces the object.").

Defendants' proposal – "comparing must follow correlating" – improperly narrows the claims and invites confusion and ambiguity. As discussed above, the plain meaning of this language refers to correlating some data about a computer object received from a first remote computer and some data about a computer object received from a second remote computer, and comparing the correlated data to other objects or entities to identify relationships. These steps do not require the same data, or any data other than data about the computer object to be correlated and/or compared. But under Defendants' "comparing follows correlating" construction, a jury may be misled to believe that *all correlating* of data must occur before *any comparing* of data can occur. Likewise, under Defendants' proposed construction, a jury could be misled into believing that comparing must follow correlating within a temporal time limit, when all that is required is comparing of the correlated data at any time. Even further, under Defendants' proposed construction, a jury could be misled into believing that comparing does not follow correlating if an initial correlating step occurs, and then at some future time a comparing step occurs followed by another correlating step. Each of these potential misuses are created by Defendants' proposed "comparing follows correlating" construction and each one conflicts with the unambiguous meaning of this claim term.

Accordingly, there is no need for Sophos's and Forcepoint's proposed construction, which only invites inapplicable interpretations. Plain and ordinary meaning is sufficient.

I. "correlating, by the base computer, at least a portion of the data about the computer object received from the first remote computer to at least a portion of the data about the computer object received from the second remote computer" ('389 patent, claim 1)

Plaintiffs	Kaspersky
Plain and ordinary meaning	"correlating, by the base computer, at least a portion of the data about the computer object received from the first remote computer to at least a portion of the data about the ***same*** computer object received from the second remote computer"

Kaspersky's proposed construction should be rejected because it adds unnecessary additional claim language—the word "same" highlighted in the box above—that may lead to ambiguity and juror confusion. The plain language of the claim refers to correlating data about ***the computer object*** received from the first remote computer, to data about ***the computer object*** received from the second remote computer. This plain language indicates that the computer object is present on the first remote computer and is also present on the second remote computer. It is not technically (or required to be) the "same" computer object as it could, for example, be two instances of the computer object, like two different instances of Microsoft Word. Kaspersky's proposed construction creates uncertainty and confusion by potentially creating the misimpression that the "same" object means that, for example, the same instance of Microsoft Word or Notepad has to be running on the first remote computer and the second remote computer. The jury will be more than capable of understanding the plain meaning of this claim language and no added, potentially confusing claim language is needed.

J. "remote computer(s)" ('250 patent, claims 1, 2, 5, 7, 10, 12-14, 17, 19, 22, 24, 25, and 28; '389 patent, claims 1, 6, 8, 9, 12, 14, and 30)

Plaintiffs	Trend Micro/ForcePoint
Plain and ordinary meaning	"local terminal(s) connected to the base computer via the Internet"

Only two Defendants propose a construction of "remote computer(s)" and it markedly departs from the plain and ordinary meaning of the claim language. The claims of the '250 and

’389 patents require a “base computer” to receive data about a computer object from a “remote computer.” Thus, the plain and ordinary meaning of a “remote computer” is simply a computer that is “remote” in comparison to the base computer.

Defendants’ proposed construction departs from this plain and ordinary meaning in many respects. First, it requires the remote computer to be a “local terminal,” which is a specific type of computer. Second, it conflicts with the plain and ordinary meaning of “remote” in the context of the claim language. As discussed above, the claims use “remote” to refer to the location of the remote computer in comparison to a base computer. The term “local terminal” is divorced from the location of the base computer. Third, Defendants’ construction narrows the term to local terminals that are ***connected to the Internet***. This is not only just one type of computer; it is further narrowed to one type of computer connected to a particular communication network. Nothing in the term “remote computer” limits the type of computer or how it is connected other than the term “remote” which as discussed above simply refers to “remote” in comparison to the base computer.

Defendants attempt to redefine “remote computer” from its ordinary meaning focuses solely on the GB application (“GB Priority Application”) to which the ’250 patent claims priority. ECF 86 at 19-20. Defendants admit, however, that the term “remote computer” appears in US application 11/477,807 which issued as the ‘250 patent. ECF 86 at 19-20 (citing ’250 patent at 7:54-59, Figs. 1 and 2). The ’250 Patent includes detailed description of remote computers that are not limited to a local terminal connected to the internet. The specifications of the ’250 and ’389 Patents do not, for example, require that the network transmitting data to the base computer be limited to the “Internet.” Both patents reprise the description of the “computer network” as “being based around **a distributed network *such as* the Internet 1**.” (Exhibit 1, ’250 Patent, 7:54-56;

Exhibit 2, '389 Patent 7:58-61.) Furthermore, both explain that the "Internet" or some "other network" may be used to transmit data to the base computer:

> If the object is unknown locally, then details of the object 45 are **passed over *the Internet 1 or other network* to the base computer** 3 for storing in the community database 7 and preferably for further analysis at the base computer 3.

(Exhibit 1, '250 Patent, 8:44-47; Exhibit 2, '389 Patent, 8:47-50.)

Further, Defendants' claim that the GB Priority Application is limited to a Local Terminal connected to the internet is mistaken. The GB Priority Application describes the "computer network" of the invention as follows: "[r]eferring to Figure 1, a computer network is shown as being based around **a distributed network such as the Internet 1**." (Exhibit 5, GB Priority Application at 5 (emphasis added).) The plain language of the GB Priority Application shows that (1) the invention includes "a computer network"; and (2) the "computer network" is based around a "distributed network such as the Internet." The use of the phrase "such as" indicates that the "Internet" is just one example of a "distributed network" contemplated by the inventors. Nothing in this phrase suggests that a "distributed network*" must be* the "Internet." And other passages in the GB Priority Application support this view. For instance, in describing the operation of the invention, the GB Priority Application explains that the "Internet" or some "other network" may be used to transmit data to the base computer:

> If the process is unknown, then this means that the process has not been run before at that local terminal. In this case, details of the process are passed **over *the Internet* or *other network*** to the community database and the community database is searched for a signature for that process at step 25.

(Exhibit 5, GB Priority Application at 7:23-8:2.) Indeed, several passages in the GB Application, refer to the use of "other networks" besides the Internet. (*See id.* at 7:23-8:2; Exhibit 1, '250 Patent, 8:44-47; Exhibit 2, '389 Patent, 8:47-50.) Further, Defendants' assertion that the GB Priority

Application only discloses a "local terminal" as a computer is false. The GB Priority Application is replete with general references to the term "computer," thus confirming that the disclosure is not limited to one type of computer. (Exhibit 5, GB Priority Application 1:1-5 ("This invention relates to computer security. In particular, it relates to methods and systems for detecting whether a process operated on a computer is likely to be one which is safe to operate or is likely malware"); 3:1-3 ("An object of the invention is to be able to detect whether a processor component operated on a computer is likely to be one which is safe to operate or is likely to be malware"); 3:9-10 ("according to the present invention n a first aspect, there is provided a computer network…"); *see also* 1:8-11; 1:13-18; 4:4-11; 4:13-20. Indeed, even the GB Priority Application's description of terminal includes ""a computer … or any other device capable of loading and operating processes" *Id.* at 5:20-22. Thus, Defendants' attempt to narrowly define "remote computer" based on the GB Application is flawed. In any event, Defendants' attempt to insert the GB Priority Application disclosure into the '250 claims to limit the term "remote computer" improperly imports limitations from the specification into the claims. *Deere,* 703 F.3d at 1354. *See also EI du Pont de Nemours & Co. v. Phillips Petro. Co*., 849 F. 2d 1430, 1433 (Fed. Cir. 1988) (stating that "[i]t is entirely proper to use the specification to interpret what the patentee meant by a word or phrase in the claim. . .[b]ut this is not to be confused with adding an extraneous limitation appearing in the specification, which is improper"); *Arlington Indus., Inc. v. Bridgeport Fittings, Inc*., 345 F.3d 1318, 1327 (Fed. Cir. 2003); *Gart v. Logitech, Inc*., 254 F.3d 1334, 1343 (Fed. Cir. 2001).

Defendants' reliance on the '250 Patent prosecution history is similarly flawed. The entire excerpt of the '250 Patent file history reads as follows: "Claim 1 requires running objects on a remote computer while behaviour is monitored. Dozortsev teaches receiving an object at a central computer and investigating the object centrally." *See* ECF 86 at 21 citing to Ex. A-2 ('250 File

History) at 4. This passage only contrasts the different activities of the claimed "remote computer" and the "central computer" of Dozortsev. It does not clearly and unambiguously disavow certain types of remote computers, nor does it even require the use of Internet to connect a remote computer to a base computer. Accordingly, Defendants' construction of "remote computers" should be rejected.

III. '386 PATENT

The '386 patent is entitled "System and Method for Developing a Risk Profile for an Internet Service," and is generally directed to a method for controlling access to an Internet resource by determining that resource's reputation and risk. Before the filing of the '386 Patent, conventional methods for preventing users from accessing malicious code on webpages involved "'Content Filtering' or "Security solutions, such as antivirus products." (Exhibit 6, '386 Patent, 1:30-46.) Content filtering is where "Web sites are organized into categories and requests for Web content are matched against per-category policies and either allowed or blocked." (*Id.* at 1:30-40.) Security solutions like anti-virus products, by contrast, "examine file or Web page content to discover known patterns or signatures that represent security threats to users, computers, or corporate networks. These focus not on the subject matter of a site but look for viruses and other 'malware that are currently infecting the site.'" (*Id.* at 1:41-46.) These approaches, however, are reactive rather than predictive; for example, they focus on "accessing known infected sites" and "identif[ying] and distribut[ing]" signatures of known malware. (*Id.* at 1:46-50.)

Unlike these prior art systems (*e.g.*, signature matching systems) which cannot predictively prevent access, the '386 Patent improved on prior-art security solutions by assigning risk profiles to internet resources that have not been previously characterized before granting or allowing access. (*Id.* at 1:23-24.) The patent explains that a "predictive security assessment for an Internet

resource is provided based on known facts about the Internet resource, which is more secure than relying only on knowledge of previously experienced security attacks." (*Id.* at 9:42-45.) Techniques described in the '386 Patent system include the use of samples of internet resources that have varying degrees of risk and, using a reputational model, predicts an internet resource's relative degree of risk. (*Id.* at 4:34-36.)

A. "security appliance" ('386 patent, claims 1, 5, 7, and 11)

Plaintiffs	ForcePoint
Plain and ordinary meaning	"a security device separate from, but on the same local area network as, the client-side device(s)"

Like many of Defendants' proposed constructions, Forcepoint's proposed construction of "security appliance" markedly departs from the plain and ordinary meaning of the term, and improperly attempts to import an embodiment from the '386 specification into the claims. The plain and ordinary meaning of "security appliance" is simply an appliance used in providing security and would be easily understood by a lay jury.

Forcepoint's proposed construction should be rejected because it departs from this plain and ordinary meaning in several respects--it changes "appliance" to "device" and then adds "separate from, but on the same local area network as, the client-side device(s)." First, Forcepoint reads into the term "security appliance" a requirement for a local area network and a client-side device in addition to the security appliance. Second, Forcepoint incorrectly equates the term "appliance" with the term "device." Skilled artisans would understand that an "appliance" may encompass hardware or software. Third, it requires the "security appliance" to be "separate from" a client-side device when the term "security appliance" itself makes no mention of a client-side device. Fourth, it reads into the term "security appliance" a requirement for the security appliance to be situated in a particular location—a local area network that is the same local area network as

the client-side devices. Nothing in the term "security appliance" contains any of these additional device or location requirements.

Forcepoint points to nothing in the intrinsic record that clearly and explicitly defines "security appliance" in a manner different from its plain and ordinary meaning. Instead, throughout its argument, Forcepoint variously claims that the intrinsic record "supports" its construction (ECF 86 at 25), that "every embodiment…describes the security appliance as a separate device" (*id.*), and that the specification "uniformly" describes "the 'security appliance' as a device or piece of equipment 'deployed' on a local area network and describe the internal components of the 'security appliance.'" ECF 86 at 23.

In reality, Forcepoint's claims about the '386 patent specification are mistaken. The specification does not "uniformly" describe the security appliance as a device or piece of equipment (hardware). For example, several claims and specification passages cited by Forcepoint can be understood to refer to hardware and/or software. For example, claim 1 refers to "receiving" a request "at a security appliance of a local area network." (Exhibit 6, '386 Patent, Cl. 1.) A "security appliance" that is software can be a "security appliance of a local area network" in accordance with this claim language. Forcepoint also quotes the abstract statement referring to a "security appliance for the LAN" which could equally refer to a software that provides security for the LAN. Similarly, Forcepoint refers to the specification statement that "a security appliance is deployed at a LAN" which again could equally refer to software deployed at a LAN.

Similarly, the specification and claims do not "uniformly" describe the security appliance as having "internal" components. The '386 patent never states that the security appliance must have specific physical internal components. The '386 patent describes the security appliance as having modules that perform specific tasks—much like a typical software system. (*See, e.g.*,

Exhibit 6, ’386 patent, FIG. 11 (depicting a security appliance using a schematic diagram).) As disclosed in nonlimiting embodiments, such a software system may have various modules, such as a network access module and a service engine. (*Id.* at, FIG. 11, 10:5–16, 51–56, 11:15–19, 26–28, 14:66–15:2.) These descriptions of the security appliance as having various software modules is consistent with the security appliance being software not just a physical device.

And even if Forcepoint was right (it is not) and the specification embodiments described only hardware implementations of a security appliance, the claims are not limited to embodiments described in the specification, or even preferred embodiments. *Best Med. Int'l, Inc. v. Elekta Inc.*, 46 F.4th 1346, 1355 (Fed. Cir. 2022); *Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 906 (Fed. Cir. 2004) (“Even when the specification describes only a single embodiment, the claims of the patent will not be read restrictively unless the patentee has demonstrated a clear intention to limit the claim scope using ‘words or expressions of manifest exclusion or restriction.’” (citation omitted)).

Further, the claims do not “require” the “security appliance” to be separate and distinct from the client-side devices as Forcepoint claims. ECF 86 at 24. If anything, the claims require that the term “security appliance” **not** be construed as Forcepoint proposes. Inserting Forcepoint’s proposed construction creates inconsistencies, ambiguities and confusion:

’386 Patent Claim 1	’386 Patent with Forcepoint’s construction inserted
receiving the request for the Internet resource at a security appliance of the local area network prior to transmission of the request over the Internet;	receiving the request for the Internet resource at a ***security device separate from, but on the same local area network as, the client-side device(s) of the local area network*** prior to transmission of the request over the Internet;

When plugged into the claim language, Forcepoint’s proposed claim language contradicts the claim language as the security device is “***of*** the local area network,” but Forcepoint’s construction inconsistently refers to a security device as being “***on*** the same local area network.” In addition,

while the actual "security appliance" claim language is clear that "of the local area network" refers to the security appliance, Forcepoint's proposed construction could misleadingly be understood to suggest that "of the local area network" refers to the client-side devices. Perhaps worse, while the actual claim language is straightforward, Forcepoint's proposed construction presents the jury with a grammatically confusing mess: "receiving the request for the Internet resource at a security device separate from, but on the same local area network as, the client-side device(s) of the local area network." There is no reason to invite such juror confusion and turn the simple term "security appliance" into something complicated and confusing.

Forcepoint's contention is also based on an improper parsing of the claim language that crumbles when the actual claim language is analyzed. The "transmitting" element recites "transmitting a request for an Internet resource from an Internet-enabled client application from a client-side device of a local area network" and the "receiving" element recites "receiving the request for the Internet resource at a security appliance of the local area network prior to transmission of the request over the Internet." While claim 1 includes a "client-side device" element in connection with a "transmitting" step and a "security appliance" element in connection with a "receiving" step, nothing in the claim language suggests that the security appliance cannot, for example, be located within one or more of the client-side devices. *Skedco, Inc. v. Strategic Operations, Inc.*, 685 F. App'x 956, 960 (Fed. Cir. 2017) (finding that nothing in the claim language forbids one claim limitation from being housed in another limitation); *Linear Tech. Corp. v. Int'l Trade Comm'n*, 566 F.3d 1049, 1055 (Fed. Cir. 2009) ("[T]here is nothing in the claim language or specification that supports narrowly construing the terms to require a specific structural requirement or entirely distinct "second" and "third" circuits."). Further, the transmission does not have to be performed by or be a transmission external to the client-side

device; it merely has to be a transmission of a "request … from an Internet-enabled client application from a client device…." It is the request that must be from an internet-enabled client application of a client device, not the transmission. Similarly, the "receiving" step merely requires receiving the request at a security appliance of the local area network. This language does not at all limit the location of the security appliance vis-à-vis the client device. Indeed, the only limitation concerning the location of the security appliance--"of a local area network"—supports that the security appliance could be part of or within the client device. The "transmitting" claim language refers to a request from a "client-device of a local area network" and the "security appliance" is also of a local area network." Thus, the security appliance could be within the client device and both be "of a local area network."

The '386 patent confirms that the security appliance can reside anywhere on the local area network, including on the client-side devices. For instance, the specification describes the client-side devices using physical examples, such as a desktop computer. (Exhibit 6, '386 patent, FIG. 11, 10:7–12.) In contrast, the specification describes the security appliance by how it functions, noting the network access and service engine. (*Id.* at FIG. 11, 10:12–16.) Nowhere does the specification indicate that the security appliance, including a network access and service engine, must reside on a physical device that is separate from all the client-side devices.

Because Forcepoint's proposed construction departs from the ordinary meaning of the claim language, is not supported by the claims or the specification, and improperly attempts to read embodiments from the specification into the claims, its proposed construction of "security appliance" should be rejected and the term should be given its plain and ordinary meaning.

IV. ’045/’224 PATENTS

The ’045 and ’224 Patents, referred to herein as the “Forensic Visibility Patents,” are entitled “Method and Apparatus for Providing Forensic Visibility into Systems and Networks.” Unlike traditional network forensic tools, these patents create forensic visibility into the computing devices on the communication network to identify malware or other security issues in operation of those devices. (*See* Exhibit 8, ’045 Patent, 2:36-38; Exhibit 9, ’224 Patent, 2:38-40.) In particular, the Forensic Visibility Patents improve network security by gathering an “event,” generating “contextual state information,” obtaining a “global perspective” for the event in comparison to other events, and generating/transmitting an “event line” that includes information for the event. (*See* Exhibit 8, ’045 Patent, cl. 1; Exhibit 9, ’224 Patent, cl. 1.) The described and claimed systems and methods intercept network events, create audit trails, or contextual states, for each individual event by correlating the event to objects such as their originating processes, devices, and/or users, and establishing a global perspective of the objects.

In addition to analyzing the behavior of an object to identify those that are potentially malicious, malware detection is further improved by understanding the context of the vent and computer objects of interest. (*See* Exhibit 8, ’045 Patent, 2:39-45 (“The system filters may be built upon the same or similar technology related to behavior monitoring and collection, as discussed in U.S. application Ser. No. 13/372,375 filed Feb. 13, 2012, (Methods and Apparatus for Dealing with Malware”).) In particular, in many cases a potentially malicious object is identified by the system as a result of other events that provide information as to whether the code is malicious. For example, if an object or event under investigation originated from an object or event that is known to be malicious or have malicious behaviors or characteristics, the presence of the known,

malicious object provides a further indication that the potentially malicious object or event is malicious as well.

The '045 and '224 Patents further explain that in addition to context information, the systems and techniques can also use information from the network to obtain a global perspective of the network operation. The combination of contextual information and global perspective enables detection of new zero-day threats, including objects created from objects (or similar objects) that have been identified previously as malicious. Indeed, in the context of modern computers and network systems that generate tens of millions of events every minute, the use of a global perspective and contextual information to correlate an event or object under investigation with prior, related events and objects—including the originating object—significantly improves the ability of the system to identify potential threats.

The '045 and '224 Patents further disclose technical improvements to forensic systems by "assembling" or "generating" an "event line" based on the contextual information—including the correlation to the originating object—and global perspective. (*See, e.g.*, Exhibit 8, '045 Patent, 9:50-58.) The generation of the event line makes it easier for end users to "identify events, and/or instances of malware, that require more immediate attention"—thereby improving the accuracy and efficiency of identifying additional malicious code, as well as enabling administrators to more readily analyze malware, assess vulnerabilities, and correct damage done by the originating objects (and other objects in the event chain). (*See id*. at 9:45-49.) The generation and use of an event line itself was, at the time, an unconventional way in which event information, contextual state information and global perspectives are generated, communicated, and/or potentially displayed to, and interacted with by, an administrator or end user.

A. "global perspective" ('045 patent, claim 1; '224 patent, claims 1, 10, and 17)

Plaintiffs	Sophos
Plain and ordinary meaning	Indefinite

Only one Defendant contends that this term is indefinite—the others agree with Plaintiffs that a person of ordinary skill in the art understands the meaning of this term. Sophos has not shown by clear and convincing evidence that the '224 and '045 patents are indefinite because the patents fail to inform a person having ordinary skill in the art about the scope of the term "global perspective" with reasonable certainty. *Nautilus*, 572 U.S. at 910. In the context of the '045 and '224 patents, a person having ordinary skill in the art would understand the scope of the term "global perspective" with reasonable certainty. (Exhibit 4 at ¶ 82.) For example, the '224 patent claims recite that the "global perspective relates to one or more other events related to the event across a network." Likewise, Claim 1 of '045 patent recites that "the global perspective [is] for one or more related events to the at least one event across a network." (*See* Section B *infra*). This claim language informs a person of ordinary skill in the art that the term "global perspective" includes events related to a particular event being monitored across a network. (Exhibit 4 at ¶¶ 83-85.) Contrary to Sophos' insistence that the context of the claim language should be ignored (ECF 86 at 27), courts regularly rely on limitations in the claim to find the boundaries of terms. *See, e.g., Niazi Licensing Corp. v. St. Jude Med. S.C., Inc.*, 30 F.4th 1339, 1349 (Fed. Cir. 2022) ("The claim language itself provides guidance on what this term means—the outer catheter must have 'shape memory.'"); *R2 Sols. LLC v. Deezer S.A.*, No. 4:21-CV-122, 2022 WL 36240, at *17 (E.D. Tex. Jan. 4, 2022) ("The claim itself discusses the meaning of 'intents' by reciting that 'each of the plurality of intents indicates a type of information regarding the query keyword that is likely to be desired by a user submitting the query.'").

The specification further discloses several examples of information that a "global perspective" can include, such as "an age 1260, popularity 1264, malware determination 1268, and/or IP/URL information 1272, where IP/URL information may include, but is not limited to a categoryID of the IP/URL, a confidence of such a categoryID, a reputation, geolocation information such as which country the IP/URL is associated with, and/or a threat history associated with the IP/URL." (Exhibit 8, '045 patent, at 23:4-12; *see also* 3:59-4:3; 13:3-17; 9:19-27; 9:39-45; 17:51-59; 17:63-18:2; 18:61-19:5.) These are examples of information that may "relate[] to one or more other events related to the [monitored] event across a network." (Exhibit 4 at ¶¶ 86-88.)

In light of the guidelines provided by the claim language and specification, a POSITA would understand the scope of the term "global perspective" with reasonable certainty. (Exhibit 4 at ¶ 89); *Enzo Biochem, Inc. v. Applera Corp.*, 599 F.3d 1325, 1335 (Fed. Cir. 2010) (holding that the intrinsic evidence provides a general guideline and examples that are sufficient to enable a person of ordinary skill in the art to determine the scope of the claims); *see also Niazi Licensing*, 30 F.4th at 1346 (Fed. Cir. 2022) (holding that the term "resilient" is not indefinite because the claim language and written description both provide guidance on what the term means); *Interval Licensing LLC v. AOL, Inc.*, 766 F.3d 1364, 1373 (Fed. Cir. 2014) ("a patent which defines a claim phrase through examples may satisfy the definiteness requirement"); *R2 Sols.*, 2022 WL 36240, at *17 (E.D. Tex. Jan. 4, 2022) (holding that defendants failed to prove indefiniteness in light of the "context provided by the claim itself as well as the specification").

The prosecution history of both '045 and '224 patents further inform a POSITA of the scope of the term "global perspective" with reasonable certainty. During prosecution of the '045 patent, Applicant added the clause "wherein the global perspective for one or more related events

to the at least one event across a network" and pointed out that the Kumar reference cited by the PTO does not disclose this clause because "there is ***no information*** displayed ***about events*** related to the application integrity ***across other devices on a network***." (Exhibit 10, '045 Patent File History, Response to OA dated 08/31/2016 at 11 (emphasis added).) A similar statement was made during prosecution of the '224 patent. (Exhibit 11, '224 Patent File History, Response to OA dated 09/05/2017 at 13.) From these statements, a POSITA would understand that a "global perspective" includes information about events related to the monitored event across a network. (Exhibit 4 at ¶90.) The PTO also had no problem understanding the term, strongly suggesting its definiteness. *See Niazi Licensing*, 30 F.4th at 1348 ("We also found it highly relevant that the examiner understood this phrase throughout prosecution."). In fact, Defendant Kaspersky also appears to understand the scope of "global perspective" because they use this term to construe "event line." ECF 86 at 33-34.

Sophos argues that "the prosecution history shows that the 'global perspective' by itself is not required to be related to multiple events across a network (ECF 86 at 27-28) but that statement does not render the claims indefinite. Rather, it confirms that a POSITA would understand that the claims require that the recited "global perspective" includes information about "one or more related events" across a network. (Exhibit 4 at ¶ 90.)

Sophos also contends that "the patent fails to articulate any objective boundaries between what constitutes a 'contextual state' and a 'global perspective.'" ECF 86 at 28-29. For example, Sophos wrongly contends that "the global perspective may simply be the same as the event information or the contextual state," citing a specification statement that the "global perspective may comprise information for each object in the event and/or the contextual state." *Id*. But Sophos ignores the rest of the disclosure *in the same paragraph*, including the description of Figure 5C,

where contextual state is labelled as 524 and global perspectives are labelled as 528, 532, and 536. (Exhibit 9, '224 patent, 17:56-66 and FIG. 5C.) In addition, examples of "global perspective" disclosed in the same paragraph, such as age, popularity, and malware determination, are all specific examples of information about other events across the network that may relate to the monitored event under the general guideline discussed above. (Exhibit 4 at ¶¶ 86-88.) Accordingly, a POSITA would readily understand the distinction between "contextual state" and "global perspective" after reviewing the patent. (*Id.* at 91.)

The *IQASR* case cited by Sophos is also readily distinguishable because neither the claim nor the specification in *IQASR* set forth any general guideline to explain the term "magnetic fuzz." *IQASR LLC v. Wendt Corp.*, 825 F. App'x 900, 906 (Fed. Cir. 2020). In fact, courts have relied on the same distinction to conclude that a claim is not indefinite. *See, e.g., R2 Sols.*, 2022 WL 36240, at *18 ("Non-limiting examples do not on their own expressly define the bounds—the limits—of the claim.") (internal quotation omitted).

Furthermore, Sophos provided no expert testimony in support of its indefiniteness contentions, and thus its arguments are deficient. Courts have found that a party has not met its burden of proving indefiniteness where it "does not provide or rely upon any expert for its proposition that the term is indefinite or that one of ordinary skill in the art would not be reasonably certain as to the scope of [the] claim element." *Freeny*, No. 2:13-CV-791-RSP, 2015 WL 294102, at *14; *see also Luminati*, No. 2:18-CV-00483-JRG, 2019 WL 6683268, at *9; *Akzo Nobel Coatings*, 811 F.3d at 1345. Sophos cannot now attempt to prove indefiniteness by clear and convincing evidence by belatedly offering evidence in a reply brief. *See, e.g.*, *Jackson*, 426 F.3d at 304 n.2; *TracBeam, LLC*, 2015 WL 5786449, at *2.

Accordingly, Sophos has failed to prove by clear and convincing evidence that the term "global perspective" is indefinite.

B. "wherein the global perspective for one or more related events to the at least one event across a network" ('045 patent, claim 1)

Plaintiffs	CrowdStrike/Sophos/Trend Micro/Kaspersky
"wherein the global perspective **is** for one or more related events to the at least one event across **other devices on** a network"	Indefinite

Plaintiffs propose construing the claim language to clarify how a POSITA would readily understand the claim term, which includes adding the verb "is" where a person of ordinary skill in the art would understand the term is missing, and adding the terms "other devices on" to clarify what is implied by the claim language "across a network" and conform the claim language with what applicant stated during prosecution.

As discussed above, a person of ordinary skill in the art would understand the global perspective in the context of the '045 patent claims to relate to information about events across the network. The "wherein" clause identifies a particular aspect of global perspective—it is "for one or more related events to the at least one event across other devices on a network" Plaintiffs' proposed construction, accordingly, simply adds the term "is" where it has been omitted. Even Defendants concede that the verb is missing. ECF 86 at 30. Plaintiffs' proposed construction is confirmed by the '045 prosecution. The "wherein clause" was added during prosecution in response to a rejection based on US Patent Application Publication No. 20130298244 to Kumar et al., ("Kumar"), which discloses a graphical user interface (GUI) for tracking application integrity. (Exhibit 10, '045 Patent File History, Response to OA dated 08/31/2016 at 2, 10-11.) Applicant pointed out that Kumar's GUI only "presents information related to the integrity of *a single application running on a single device*," and "there is no information displayed about *events*

related to the application integrity across other devices on a network." *Id* (emphasis added). Plaintiff's proposed construction tracks applicant's statement during prosecution that the global information is for events across other devices on a network and should be adopted.

Defendants fail to prove by clear and convincing evidence that the "wherein clause" is indefinite. ECF 86 at 32-33. For starters, Defendants proffer no expert testimony that a person of ordinary skill would not understand the claim language with reasonable certainty. That alone is fatal. *Freeny*, No. 2:13-CV-791-RSP, 2015 WL 294102, at *14. *See also Luminati*, No. 2:18-CV-00483-JRG, 2019 WL 6683268, at *9; *Akzo Nobel Coatings*, 811 F.3d at 1345. Defendants cannot cure this defect through a table of "alternate" constructions for several reasons. (ECF 86 at 32 (Table).) First, Defendants admit that the "wherein" clause was added during prosecution to overcome prior art, but all but one of Defendants' purported alternate constructions ignore this context and cite instead to unrelated portions of the '045 patent specification or the '224 patent claims. ECF 86 at 32 (Table row 1 citing '045 patent at 17:51-52, row 3 citing '045 patent at 13:25-27, row 4 citing '224 patent at cl. 1).

Second, Defendants' hypothetical constructions lack a proper antecedent basis. Each hypothetical construction recites "***the*** global perspective for one or more related events," thus requiring an antecedent basis for this entire claim phrase. But there is none. By contrast, Plaintiffs' proposed construction refers to "the global perspective" and an antecedent basis ("a global perspective") exists in claim 1. Contrary to Defendants' allegation of conflicts with the antecedent claim language (DI 86 at 32, FN 6), Plaintiff's proposed construction is in fact the only construction that has proper antecedent basis.

Third, three of Defendants' four hypothetical constructions (the "stored," "applied," and "displayed" constructions) are inconsistent with the prosecution history. As discussed above,

Applicant distinguished the "wherein clause" from Kumar on the ground that Kumar only "presents information related to the integrity of a single application running on a single device," while claim 1 recites "information about events related to the application integrity across other devices on a network," as required by the "wherein clause." (Exhibit 10, '045 Patent File History, Response to OA dated 08/31/2016 at 2 and 10-11.) Therefore, a POSITA would readily understand that the "wherein clause" requires the global perspective to include information about events across other devices on a network, instead of requiring the global perspective to be stored, applied, or displayed. (Exhibit 4 at ¶ 95.) Accordingly, Plaintiff's proposed construction should be adopted.

C. "event line" ('045 patent, claim 1; '224 patent, claims 1, 10, and 17)

Plaintiffs	Kaspersky
Plain and ordinary meaning	"a line of data containing at least event, contextual state, and global perspective information"

Plaintiff dispute both parts of Kaspersky's construction of "event line." First, Kaspersky identifies no intrinsic support (and there is none) that requires narrowly construing the "event line" to be "a line of data." This construction is inconsistent with at least FIG. 12 and its corresponding description, which provides details of data structures 1200 that could comprise an event line. (Exhibit 8, '045 Patent, 22:55-23:23 ("[i]n some embodiments, the data structure 1200 may be used as an event line")):

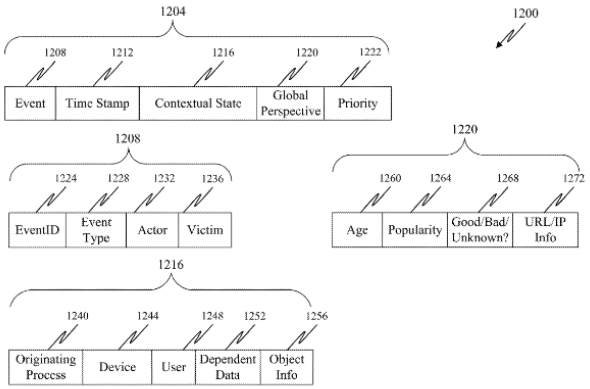


Figure 12

Figure 12 clearly depicts an "event line" as comprising one or more data structures, not as a single "line of data." While the specification discloses "a line of log data," this is not identified as an event line, but instead "used to send each event as a line of log data to be processed by an end consumer—usually a display/query engine)." (*Id*. at 9:66-10:2; 13:52-54.) Kaspersky's attempt to restrict the event line information to implementations where the event line is solely formed as a "line of data" is unsupported by the specification.

In addition, none of the intrinsic evidence cited by Kaspersky *requires* that that event line includes "at least event, contextual state, and global perspective information." In fact, the specification discloses that the "event line" *may* include these types of information. *See, e.g.*, '045, at 4:1-7 9:50-58; 13:16-17; 17:61-63. In other cases, the types of information included in an "event line" are only limited to "some embodiments." *See, e.g.,* 5:40-43; 5:53-56; 22:55-60.

Because the specification does not required Kaspersky's narrow construction, which expressly reads out embodiments disclosed in the specification, Kaspersky's proposed construction should be rejected.

D. "gathering an event defining an action of a first object acting on a target…; generating contextual state information for the event by correlating the event to an originating object of the first object; obtaining a global perspective for the

event based on the contextual state information, wherein the global perspective comprises information associated with one or more of the first object and the originating object, and wherein the global perspective relates to one or more other events related to the event across a network;" ('045 patent, claims 1, 8, and 16; '224 patent, claims 1, 10, and 17)

Plaintiffs	CrowdStrike/Sophos
Plain and ordinary meaning	Order of steps: gathering, generating, and obtaining steps must be performed in order

Defendants' proposed construction confining the gathering, generating and obtaining steps to performance in a strict sequential order should be rejected. The general rule is that a method claim is not limited to performance of the steps in the order recited, unless the claim explicitly or implicitly requires a specific order. *Baldwin Graphic Sys., Inc. v. Siebert, Inc.*, 512 F.3d 1338, 1345 (Fed. Cir. 2008). Defendants misstate the law in their attempt to impose a strict order on the gathering, generating and obtaining steps. Defendants contend that "[w]hile claims are ordinarily presumed to not require an order, the opposition [sic] presumption applies when the language of the claims mandates that these steps are performed in the order in which they are recited," citing to *Wi-LAN, Inc. v. Apple, Inc.*, 811 F.3d 455, 460 (Fed. Cir. 2016). (DI 86, at 35). But the *Wi-LAN* case does not impose such an "opposition [sic] presumption" and no such "opposite" presumption exists. *Wi-LAN*, 811 F.3d at 463.

Defendants also seek to impose a strict order on the claim steps by asserting that "each step relies upon input from the previous steps," going so far as to say that "the unequivocal language of the claims … is dispositive" in determining the ordering requirement. ECF 86 at 35-36. Defendants' assertion is both factually and legally erroneous. Courts have consistently held that claim language including reference back to previous steps for input is not dispositive. *See, e.g., Microsoft*, 2022 WL 11531235, at *3 (holding that step of "comparing *the determined component*" does not have to after the step of "*determining at least one device component*") (emphasis added); *see also Interactive Gift Express,* 256 F.3d at 1328 (holding that step one of a method claim can

be performed after step four, even though step four referred to "the catalog code" introduced in step one); *Respironics*, 303 Fed.Appx. at 870 ("[I]n a method claim, a step that recites 'said' or 'the,' referring to an earlier object, does not always have to be performed after the step that first introduces the object."). Contrary to Defendants' assertions, the claim steps do not all include language that references other claim steps. For example, claims 8 and 16 of the '045 patent and claims 10 and 17 of the '224 Patent do not have claimed steps at all, and the "obtaining" step in at least '045 patent claim 1 does not refer back to the "generating" step. In all but one claim, no order of steps could possibly be implicated.

In particular, Defendants proposed order of steps argument does not legally apply to '224 patent claims 10 and 17 and '045 patent claims 8 and 16. As Defendants' admit, each of these claims are apparatus/system claims. As such, they do not recite steps but use "active verbs" to "represent permissible functional language used to describe capabilities" of the claimed processor as a structure. *Mastermine Software, Inc. v. Microsoft Corp.*, 874 F.3d 1307, 1315 (Fed. Cir. 2017). While Defendants contend otherwise (ECF 86 at 36), they parse the claim language to create the false impression that system claims have steps when they do not, and ignore the portion of the claim language that indicates the language being cited recites a particular system component that is capable of or configured to perform the claimed functions.

Moreover, (and contrary to Defendants' assertions), the '224/'045 patent specifications disclose embodiments that are not performed in Defendants' strict order. The specification expressly discloses that "the methods may be performed in a different order than that described." (Exhibit 8, '045 patent, 24:41-44.) Under this general guideline, the specification further discloses specific examples where the recited steps can be performed in different orders. For example, the specification discloses that "the ***later occurring*** duplicate system event may receive all contextual

state information and global perspective information without requiring all of the determinations and/or lookups of such information." (*Id.* at 12:2-10 (emphasis added).) In another example, both the contextual state and the global perspective can be prepared in advance and stored at various locations, such as the local cache, a server, or other computing devices, for fast retrieval. (*Id.* at 8:58-9:18.) Further, the input of the "generating" step does not have to be the event gathered in the "gathering" step; nor does the "obtaining" step require the contextual state information generated at the "generating" step as input. (Exhibit 4 at ¶¶ 98-99.) Based on this disclosure, a POSITA would understand that the "generating" and "obtaining" steps can be performed after the "gathering" step, and the "generating" and "obtaining" steps can be performed in any order.

The specification also discloses a method including obtaining an event line and then parsing the event line to retrieve event information such as "an event type, a contextual state for the event, a global perspective for the event, and a priority for the event." (Exhibit 8, '045 patent, 5:37-45.) A POSITA would understand that in this method, the "gathering," "generating," and "obtaining" step can be performed concurrently. (Exhibit 4 at ¶ 100.)

Because the claim language does not disclose steps, does not explicitly or implicitly require a specific order, and the specification discloses embodiments to perform the recited steps in different orders, the "gathering," "generating," and "obtaining" steps do not have to be performed in sequential order as recited.

E. "a sensor agent …" ('224 patent, claims 10, 14, 15, and 17)

Plaintiffs	Sophos/Trend Micro/Kaspersky
Not subject to § 112 (f); Not Indefinite.	Subject to § 112 (f); Indefinite.

The "sensor agent…" claim language in the '224 patent claims should be given its plain and ordinary meaning and is not subject to means-plus-function construction under Section 112 ¶ 6. The "sensor agent" claim language does not use the term "means." Thus, there is a rebuttable

presumption that this claim language is not subject to construction under 35 U.S.C. § 112 ¶ 6. *Williamson*, 792 F.3d 1339 at 1348-49. Defendants fail to overcome this presumption.

First, the term "agent" by itself has structural meaning. (Exhibit 4 at ¶ 102.) For example, when the application that led to the '224 patent was filed, an "agent" was generally known as "a computational entity that acts on behalf of other entities in an autonomous fashion" or "in the client-server model, the part of the system that performs information preparation and exchange on behalf of a client or server." (*Id.* at ¶ 102.) Courts have recognized the structural meaning of the term "agent." *See, e.g.*, *Genband USA LLC v. Metaswitch Networks Ltd.*, No. 2:14-cv-33, 2015 WL 4722185, at *17-19 (E.D. Tex. Aug. 6, 2015) (holding that the "agent" as used in the term "interworking agent" has structural meaning).

Second, the addition of the qualifier "sensor" further narrows the term to make its structural meaning more definite. *See Personalized Media Communications, LLC v. International Trade Comm'n,* 161 F.3d 696, 705 (Fed. Cir. 1998) ("An adjectival qualification ('digital') placed on an otherwise sufficiently definite structure ('detector') … further narrows the scope of those structures covered by the claim and makes the term more definite."); *Apex Inc. v. Raritan Computer, Inc.*, 325 F.3d 1364, 1373 (Fed. Cir. 2003) ("[T]he term 'circuit' with an appropriate identifier such as 'interface,' 'programming' and 'logic,' certainly identifies some structural meaning to one of ordinary skill in the art."). Here, the qualifier "sensor" clarifies that the "agent" is configured for collecting and processing computer events in the context of computer security. (Exhibit 4 at ¶ 103.)

Third, the "sensor agent" claim language must be considered in context as it appears in the claims and specification. *Apex Inc.* 325 F.3d 1372 (Fed. Cir. 2003); *see also Zeroclick, LLC v. Apple Inc.*, 891 F.3d 1003, 1008 (Fed. Cir. 2018) (finding district court erred where "the court's

analysis removed the terms [‘program’ and ‘user interface code’] from their context, which otherwise strongly suggests the plain and ordinary meaning of the terms.”). The claims provide further structure by reciting how the claim term interacts with other components. *See Finjan, Inc. v. Proofpoint, Inc.*, No. 13-cv-05808-HSG, 2015 WL 7770208, at *10–11 (N.D. Cal. Dec. 3, 2015). Claim 10 of the ’224 patent is exemplary and provides, in pertinent part: “a sensor agent stored on *the data storage* that is executable by ***the processor***, wherein the sensor agent is operable to … transmit the generated event line utilizing the communication interface.” (Exhibit 9, ’224 patent, cl. 10 (emphasis added).) Further, the specification makes clear that the “sensor agent” uses “an infrastructure” (*e.g.*, Exhibit 9, ’224 Patent, 3:62-4:10) and is part of or included in a “computing device” (*id.* at 7:14-33, 8:42-49, Fig. 2, 3) and calls on a “global information server” (*id.* at 9:12-25). Here, the sensor agent’s interactions with other components including being stored on the data storage, being executable by a processor, and utilizing the communication interface to transmit the event line, confirm that the “sensor agent” is a sufficiently definite structure. (Exhibit 4 at ¶ 104.)

Fourth, the claim language and specification also describe the “objective and operation” of the sensor agent. For example, claim 10 of the ’224 patent describes both the objective (e.g., gathering an event, generating contextual state information, obtaining a global perspective, generating an event line, and transmitting the event line) and operation (e.g., correlating the event to an originating object, generating the contextual state based on the contextual state information, utilizing the communication interface) of the sensor agent. These descriptions connote additional and definite structure to a POSITA. *See Collaborative Agreements, LLC v. Adobe Sys.*, No. 15-cv-03853-EMC, 2015 WL 7753293 at * 5 (N.D. Cal. Dec. 2, 2015) (holding that the term “code segment” does not invoke § 112 ¶ 6 because the claims describe the objective and operation of the code segment); *Intelligent Agency, LLC v. 7-Eleven, Inc.*, 2022 Markman 760203, 2022 WL

760203, at * 13 (E.D. Tex. 2022) ("By reciting the objectives of the 'agent-user matching algorithm,' and how the algorithm operates within the context of the claimed invention, the claim language connotes sufficiently definite structure to one of skill in the art.); *Syncpoint Imaging, LLC v. Nintendo of Am. Inc.*, No. 2:15-cv-00247-JRG-RSP, 2016 WL 55118, at *24 (E.D. Tex. Jan. 5, 2016) ("Claims 25 and 26 themselves recite the objectives and operations of the instructions in the 'Instructions For' limitations.").

In the context of the '224 patent claims and specification, the claimed "sensor agent" is "a term of art that was well-understood to a POSITA at the time of [the '224 patent's] invention as having sufficiently definite structure." (*See* Exhibit 4 at ¶ 105.) Accordingly, Defendants have not overcome the presumption that Section 112, ¶ 6 does not apply to the sensor agent terms.

Even if (incorrectly) construed under Section 112, ¶ 6, Defendants have not proved the "sensor agent" claim language indefinite by clear and convincing evidence. The '224 patent specification provides sufficient structure in the form of disclosure of structure and algorithms for how to implement the sensor agent and perform the claimed functions assigned to the "sensor agent." For example, Figures 2-4 of the '224 patent and the associated descriptions disclose in detail the structure of the sensor agent, the functions performed, and its interaction with other components. FIGS. 6-11 and their accompanying description provide flow charts to show the operation of the sensor agent in various embodiments. These disclosures are sufficient for a POSITA to understand the corresponding structure of the sensor agent that performs the claimed functions. (Exhibit 4 at ¶¶ 107-123.)

V. '591 PATENT

The '591 Patent is entitled "Detecting and Preventing Execution of Software Exploits" and describes and claims an "anti-exploit" technique to prevent undesirable software and/or other

computer exploits from executing. (*See* Exhibit 13, '591 Patent, 1:13-28, 1:32-33.) Computer "exploits" include code, software, data, or commands that take advantage of a bug, glitch, or vulnerability in a computer system. To accomplish this goal, the novel anti-exploit techniques described and claimed in the '591 Patent monitor memory space of a process for execution of functions and performs "stack walk processing" upon invocation of a function in the monitored memory space. (*Id.* at 1:33-39.) During that stack walk processing, a memory check may be performed to detect suspicious behavior. (*Id.*) If the memory check detects certain types of suspicious behavior, an alert may be triggered that prevents the execution of a payload for the invoked function. (*Id.* at 1:39-48.) The anti-exploit techniques can include performing "memory checks performed during the stack walk processing once an address is reached for an originating caller function." (*See id.* at 8:6-7.) In one embodiment, "memory checks from the lowest level user function of the hooked function down through the address of the originating caller function" may be performed to detect and identify suspicious behavior. (*Id.* at 6:7-11.)

A. "a component for evaluating the at least one monitored function in the memory space" ('591 patent, claims 1, 9, and 17)

Plaintiffs	Kaspersky
Not subject to § 112(f), not indefinite	Indefinite

One defendant, Kaspersky, contends this term is indefinite. It is not. Kaspersky concedes that the "component" claim language does not include "means" and therefore is presumed to not invoke a means-plus-function construction under § 112(f). ECF 86 at 42. Kaspersky cannot rebut the presumption against invoking § 112(f).

Kaspersky admits that the "component" claim language refers to software. *See* ECF 86 at 43 ("The specification identifies the 'component' as a 'custom module'" and "The specification makes clear that module refers to software."). "Claim terms 'need not connote a single, specific

structure,' and may instead 'describe a class of structures' and still recite 'sufficiently definite structure' to not invoke § 112, ¶6." *Dyfan*, 28 F.4th at 1356 citing *Apple Inc. v. Motorola, Inc.*, 757 F.3d 1286, 1300 (Fed. Cir. 2014). Here the "component" claim language does just that by refer to a class of structures, namely software and/or software running on hardware. Courts have routinely found claim terms like "component" refer to software and thereby connote structure. *See, e.g.*, *Zeroclick, LLC v. Apple Inc.*, 891 F.3d 1003, 1008 (Fed. Cir. 2018) ("program" and "user interface code" not means-plus-function limitations); *Dyfan, LLC v. Target Corp.*, 28 F.4th 1360, 1366–69 (Fed. Cir. 2022) ("code," "application," and "system"); *VDPP LLC v. Vizio, Inc.*, 2021-2040, 2022 WL 885771, at *3–4 (Fed. Cir. Mar. 25, 2022) ("storage" and "processor"); *CoreLogic Information Solutions, Inc. v. Fiserv, Inc.*, 2012 WL 4355394, *7–8 (E.D. Tex. 2012) ("model development component" not a means-plus-function limitation and one of skill would understand that "component" was part of the software); *Leader Technologies, Inc. v. Facebook, Inc.*, 2010 WL 809848, *5–6 (D. Del. 2010) ("storage component," "tracking component," and "context component" not means-plus-function limitations even though specification had a broad definition of component); *Widevine Technologies, Inc. v. Verimatrix, Inc.*, 2009 WL 3734106, *16 (E.D. Tex. 2009) ("component" in claim limitation reciting "a component that is configured to enable the other network device to decrypt the selectively encrypted payload portion of the other packet" not a means-plus-function limitation); *Huawei Techs. Co. v. Verizon Commc'ns, Inc.*, No. 2:20-CV-30, 2021 WL 150442, at *32–33, *54–55 (E.D. Tex. Jan. 15, 2021) ("processing unit" and "processing module" not means-plus-function and not indefinite); *CXT Sys., Inc. v. Acad., Ltd.*, No. 2:18-CV-171, 2019 WL 4253841, at *11 (E.D. Tex. Sept. 6, 2019) ("processing module for processing" not means-plus-function and not indefinite).

The claim language and specification are clear that a POSITA would understand "a component for evaluating the at least one monitored function in the memory space" to refer to sufficiently definite structure, namely software and/or software running on hardware. For example, the claim language states that the "component for evaluating the at least one monitored function" is loaded in "memory space." (*See, e.g.,* Exhibit 13, '591 Patent, cl. 1 ("wherein monitoring the memory space comprises ***loading a component for evaluating*** the at least one monitored function in the memory space" (emphasis added)).) In fact, even co-defendant Trend Micro recognized that this term does not invoke § 112(f); proposing a non-means-plus-function construction of this very claim language. (Exhibit 14, Trend Micro's 10/7/2022 Proposed Claim Constructions at 5.)

The specification further states "[t]he anti-exploit component 102 loads a component such as a custom module into the memory space of each monitored process. This component intercepts monitored operating system applications or APIs and evaluates a record (e.g., stack frame) of a call stack." (Exhibit 13, '591 Patent, 4:45–49; *see also id.* at 4:54–58 ("When unidentified code tries to allocate/mark a new executable page of memory which is not related to any valid process module, the custom component monitors these newly allocated memory regions and prevents any attempted execution of monitored functions.").) Furthermore, the specification discloses software components running on hardware:

> In examples, system 100 may include … ***software components*** (e.g., applications, application programming interfaces, modules, virtual machines, runtime libraries, etc.) ***running on hardware***. In one example, an exemplary system 100 may provide an environment for ***software components to run, obey constraints set for operating***, and makes use of resources or facilities of the system 100, where ***components may be software (e.g., application, program, module, etc.) running on one or more processing devices***.

(*Id.* at 2:60–3:3 (emphasis added); *see also, e.g., id.* at 5:23–6:30; *id.* at FIGS. 1–3.) Accordingly, Kaspersky cannot overcome the presumption that Section 112(f) does not apply.

Even if (incorrectly) construed under Section 112(f), Kaspersky has not proved the "component" claim language indefinite by clear and convincing evidence. The '591 specification provides sufficient structure. The '591 patent specification fully describes structure for performing the claimed functions, including descriptions that would have been recognized by a POSITA to disclose one or more corresponding algorithms. (*See, e.g.,* Exhibit 13, '591 Patent, 2:60–3:3; *id.* at 4:45–58; *id.* at 5:23–6:30; *id.* at FIGS. 1–3 (demonstrating related algorithms); Ex. 4, Medvidovic Decl., ¶¶49-60.)

B. "invoking one of the plurality of functions" ('591 patent, claims 1, 9, and 17)

Plaintiffs	Sophos
Plain and ordinary meaning	"running one of the plurality of functions in kernel mode, such that it can manipulate system resources"

"Invoking one of the plurality of functions" is a simple phrase using easily understood words like "invoking" and "function," and does not require further construction. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005). Sophos's proposed construction should be rejected for several reasons. First, it substantially departs from the plain and ordinary meaning of the claim language. Nothing in the plain language "invoking one of the plurality of functions" requires that the invoking occur only in a specific mode (kernel mode). Nor does this plain language require the invoking of a plurality of functions to be for a narrow purpose – "such that it can manipulate system resources."

Second, Sophos' proposed construction is based on incorrect interpretations of the intrinsic record. Sophos cites to general disclosures of kernel mode, user mode, and system APIs in the specification that do not limit the term "invoking." ECF 86 at 44. (*See* Exhibit 13, '591 Patent, 3:43-4:4.) Sophos's contentions that "[t]he specification's explanation that system APIs are used to access kernel mode confirms Defendants' proposed construction" and the limitation "must refer

to running a function in kernel mode" (ECF 86 at 44) are unsupported attorney arguments and improperly conflate entering the kernel mode with running in the kernel mode. Indeed, the specification references that Sophos cites do not use definitional or disclaimer language to limit the term, and thus amount to nothing more than an improper attempt to import limitations from the specification into the claims. *Flypsi, Inc. v. Dialpad, Inc.*, 6:21-CV-00642-ADA, 2022 WL 3593131, at *2 (W.D. Tex. Aug. 22, 2022) ("This term does not fit the exceptions of lexicography or disclaimer to warrant a departure from the plain and ordinary meaning."); *see Phillips*, 415 F.3d at 1323 (Fed. Cir. 2005); *Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 906 (Fed. Cir. 2004).

Sophos also misrepresents the prosecution history, which does not clearly or unambiguously disavow or disclaim "invoking." *Flypsi*, 6:21-CV-00642-ADA, 2022 WL 3593131, at *2 (W.D. Tex. Aug. 22, 2022). The amendments and arguments Sophos points to were made to show that "stack walk processing," by virtue of being a system process, could allow a system call to enter the kernel mode and process the called function ***before*** the code actually runs in the kernel. This does not limit the scope of the term invoking:

> In short, whereas Yeap interrupts a ***system call from an API prior to entering the kernel mode***, ***aspects of the technology execute a stack walk upon invocation of a system call from an API***. To clarify this distinction, Applicants have amended claim 1 to include "invoking one of the plurality of functions as a result of receiving a call from an application programming instance; ***executing stack walk processing upon the invocation of one of the plurality of functions in the monitored memory space***."

(Exhibit 15, '591 Patent File History, Response to OA dated 02/28/2018 at 9–10 (emphasis added).) The same response further clarifies these arguments were directed to a stack walk being a system process—and not a limitation on the invoking term:

> It is thus, the intention of Yeap to prevent calls that require interaction with the kernel to be intercepted and analyzed for malicious software.… As such, Yeap would never allow the ***execution of a stack walk, which is a system process***, in response to a function call by an application, as the claims require.

(*Id.*at 8 (emphasis added).) Sophos's incorrect interpretation of the prosecution history does not limit the "invoking" term and improperly conflates "entering the kernel mode" with "running in kernel mode." A process can enter kernel mode without yet running in kernel mode. Furthermore, as Sophos admits, the terms "receiving a call," "invoking," and "executing" are different and presumed to have different meanings. ECF 86 at 43. Hence, Sophos's construction is incorrect because it conflates the terms "invoking" and "executing" by equating the former with "running." *See, e.g.,* '591 Patent, 2:61-62 ("execute / run operating system (OS)"). Accordingly, Sophos' attempt to construe the "invoking" claim language in a manner that departs from the plain and ordinary meaning and improperly imports limitations from the specification into the claims should be rejected.

C. "stack walk processing" ('591 patent, claims 1, 3, 5, 9, 11, 13, 17, and 18)

Plaintiffs	Trend Micro
Plain and ordinary meaning	"a process that walks a stack for a hooked function from a lowest level user mode function to an address of the invoked/originating caller function"

Only Trend Micro has offered a construction for "stack walk processing." Therefore, the other Defendants apparently agree with Plaintiff that this term need not be construed.

"Stack walk processing" is simply processing that involves using a stack walk. This plain and ordinary meaning is supported by the specification, which states, for example:

> A hooked function (including but not limited to NtCreateProcess, NtCreateProcessEX, NtCreateUserProcess, LdrLoadDll) is monitored/evaluated through execution of stack walk processing. ***The anti-exploit component 102 executes stack walk processing that walks up a call stack for the process (function) until an address of the first function of a called function is reached***. As an example, for stack walk processing, when one of the functions listed before is invoked, the first step is to get the value of the call frame pointer for being able to walk the call stack frame. The call frame pointer contains the address of the stack frame for the called function.

(Exhibit 13, '591 Patent, 5:23–34 (emphasis added); *see also id.* at 3:15–16 ("As one example, the system 100 comprises an anti-exploit component 102.…").) While this example refers to processing that walks up the stack, a person of ordinary skill in the art would understand that the stack can be walked in a variety of ways, including going down as well as up.

Trend Micro's proposed construction should be rejected because it departs from the ordinary meaning. For example, instead of just a process that walks a stack, Trend Micro limits stack walk processing to a specific type of stalk walk processing that 1) walks a stack for a specific type of function (hooked function); and 2) walks a stack from a particular origin to a particular destination (from a lowest level user mode function to an address of the invoked/originating caller function). Nothing in the plain language "stack walk processing" limits what the stalk walk processing must be performed for, or the manner of the stack walk processing.

In reality, Trend Micro's construction is a classic attempt to improperly import limitations from one embodiment into the term "stack walk processing" (*see* ECF 86 at 46), all while ignoring the rest of the specification. *Cont'l. Circuits LLC v. Intel Corp.*, 915 F.3d 788, 797 (Fed. Cir. 2019) (An "embodiment, without more, does not result in a clear disavowal of claim scope."); *Deere*, 703 F.3d at 1354. Indeed, Trend Micro's proposed construction is based on cites to specification language that is clearly exemplary only. For example:

> Flow proceeds to operation 304 where stack walk processing is performed on a hooked and/or monitored function. When one of the plurality of functions is invoked, the stack walk processing is executed for a stack frame associated with an invoked function (***e.g., originating caller function***). ***Execution of the stack walk processing is described in further detail in the description of FIG. 1***.

(*Compare* Exhibit 13, '591 Patent, 10:30–35 (emphasis added) *with* ECF 86 at 46 (citing '591 Patent at 10:35–39); *see also id.* at 2:52–54 ("FIG. 1 illustrates an exemplary system 100.…"); *id.* at 10:5–7 ("FIG. 3 illustrates an exemplary method 300.…").) Further:

> As an example, for stack walk processing, when one of the functions listed before is invoked, the first step is to get the value of the call frame pointer for being able to walk the call stack frame. The call frame pointer contains the address of the stack frame for the called function. In the examples disclosed herein, the anti-exploit technology may go back through the stack_entry chain until the initial caller's stack frame is reached (***e.g., back-walk the stack frame starting from the lowest level user mode function to an address where the call sequence started for the hooked function***)."

(Exhibit 13, '591 Patent at 5:29–39 (emphasis added). The use of the term "e.g." denotes that the descriptions are exemplary only. *See Almirall, LLC v. Amneal Pharm. LLC*, 28 F.4th 265, 270 (Fed. Cir. 2022) (using "e.g." to denote something as "exemplary").

As demonstrated, the specification generally refers to walking a stack, an action which a POSITA would have understood at the time of the invention to be consistent with the plain and ordinary meaning. (*See, e.g.,* Exhibit 13, 5:29–39; *id.* at 10:30–39; *id.* at FIGS. 1 and 3.) The '591 Patent specification includes no language that clearly and unambiguously defines or disclaims "stalk walk processing" to have a meaning other than its plain and ordinary meaning, including where the stack walk processing begins, ends, or is directed. Although Trend Micro argues for a definition, it fails to point to quotation marks or requirements clearly setting forth a definition other than plain and ordinary meaning. ECF 86 at 46; *Thorner v. Sony Comput. Ent. Am. LLC*, 669 F.3d 1362, 1365 (Fed. Cir. 2012); *Flypsi*, 6:21-CV-00642-ADA, 2022 WL 3593131, at *2. The specification clearly indicates that this language is exemplary, using terms such as "example" and "e.g." *Almirall*, 28 F.4th at 270.

Finally, Trend Micro's construction improperly imports limitations from the dependent claims into "stack walk processing" and violates the doctrine of claim differentiation. For example, Trend Micro's construction of "stack walk processing" is essentially identical to the specific stalk walk processing recited in claim 3: "walking a stack frame associated with the invoked function

from a low level user mode function to the originating address of the invoked function" (claim 3). This violates the doctrine of claim differentiation. *See Phillips*, 415 F.3d at 1315.

Accordingly, Trend Micro's attempt to depart from the plain and ordinary meaning of "stack walk processing" and to import limitations from the specification should be rejected.

D. "before an address of an originating caller function is reached" ('591 patent, claims 1, 9, and 17)

Plaintiffs	Sophos
Plain and ordinary meaning	"before the originating address of the invoked function is reached"

Only Sophos has offered a construction for "before an address of an originating caller function is reached." Therefore, the other Defendants apparently agree with Plaintiff that this term need not be construed.

The Court should reject Sophos's proposed construction which changes the claim language from its plain and ordinary meaning. The claim refers to an address of "an originating caller function," while Sophos's proposed construction refers to an address of an "invoked function." The claim language also refers simply to an "address." Sophos's proposed construction rewrites "address" to be an "originating address." Sophos provides no basis to rewrite the claims to equate an originating caller function and an invoked function. *See* ECF 86 at 47–48. The '591 Patent specification makes clear that an originating caller function is an example of a stack frame associated with an invoked function:

> Flow proceeds to operation 304 where stack walk processing is performed on a hooked and/or monitored function. When one of the plurality of functions is invoked, the stack walk processing is executed for a ***stack frame associated with an invoked function*** (***e.g., originating caller function***).

(Exhibit 13, '591 Patent, 10:30–35 (emphasis added); *see id.* at 5:26–29 ("The anti-exploit component 102 executes stack walk processing that walks up a call stack for the process (function)

until an address of the first function of a called function is reached.”).) The use of the term “e.g.” before originating caller function is a clear indication that the originating caller function is an example of a function associated with an invoked function; it does not define an invoked function. *See Almirall*, 28 F.4th at 270. The prosecution history further confirms that stack walk processing reaches “an address of an originating caller function.” (*See* Exhibit 15, ’591 Patent File History, Response to OA dated 10/01/2018 at 9 (citing Application at [0022]) (“More specifically, a stack walk processing is executed which walks up a call stack for the process (function) until an ***address of the first function of a called function*** is reached.” (emphasis added)).)

Sophos’s attempt to rewrite the claim language to equate “invoked function” and “originating caller function” misinterprets the specification and improperly attempts to import limitations into the claims via the specification. *Deere*, 703 F.3d at 1354; *Cont’l. Circuits LLC*, 915 F.3d at 797; *see also Flypsi*, 6:21-CV-00642-ADA, 2022 WL 3593131, at *4 (“patentee must clearly express an intent to redefine the term”) (quotation removed). (*See, e.g.,* Exhibit 13, ’591 Patent, 2:52 (“FIG. 1 illustrates an exemplary system 100…”); *id.* at 9:59–61; *id.* at 10:30–39; *id.* at 10:5 (“FIG. 3 illustrates an exemplary method 300…”).)

Sophos’s construction also improperly imports limitations from the dependent claims into the limitation and violates the doctrine of claim differentiation. For example, Sophos’s construction of the term essentially shifts the language from claim 3 into claim 1: “walking a stack frame associated with the invoked function from a low level user mode function to the originating address of the invoked function” (claim 3) (*compare* “during the executing of the stack walk processing before an address of an originating caller function is reached” (claim 1)) which violates the doctrine of claim differentiation. *See Phillips*, 415 F.3d at 1315.

Accordingly, Sophos' attempt to depart from the plain and ordinary meaning of the claim language should be rejected.

E. "low level user mode function" ('591 patent, claims 3 and 11)

Plaintiffs	CrowdStrike
Not indefinite, plain and ordinary meaning	Indefinite

CrowdStrike has failed to prove by clear and convincing evidence that the term "low level user mode function" is indefinite. A person having ordinary skill in the art would understand the scope of the term "low level user mode function" with reasonable certainty. The term "low level user mode function" is a common and well-understood term in the computer arts, and the Court need not construe the term beyond its plain and ordinary meaning. (Ex. 4, Medvidovic Decl., ¶¶63-66.) A person of ordinary skill would understand that the term "low level user mode function" generally refers to a user mode function that is low level with respect to its layer of abstraction. (*Id.* at ¶¶64-66.) "Low level user mode function" is described by the claims and disclosed throughout the specification in a manner consistent with its plain and ordinary meaning. (*See, e.g.,* Exhibit 13, '591 Patent, cl. 3.). The specification discloses examples of "low level" user mode functions:

> ***Low-level*** languages can be converted to machine code without using a compiler or interpreter, and the resulting code runs directly on the processor. Once the CALL instruction is reached, the code flow of the program starts to follow a chain of functions which will lead to the ***lowest level user mode function***, before transitioning to Kernel Mode. In Kernel mode, the executing code has complete and unrestricted access to the underlying hardware.

(*Id.* at 3:52–60 (emphasis added); *see also, e.g., id.* at 4:24–27 ("In order to reach a goal of starting a new process, an exploit has to execute a complete execution stack starting from the shellcode and running down through to a ***lowest level user function (e.g., instruction that includes machine or assembly instruction***." (emphasis added)); *id.* at 10:21-26 (In at least one example, operation

302 further comprises applying, upon detecting a function call of one of the plurality of functions in the monitored memory space, a function hooking technique to hook a ***lowest level user mode function of a called process (e.g., API).***) (emphasis added).) The specification even explains that "low" refers to the small or nonexistent amount of abstraction between the programming language and machine language:

> An action of calling a system process (e.g., function of an application or application programming interface (API)) from a low level perspective, involves the presence of a particular set of assembly instruction, such as a CALL, inside the process memory. 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.

(Exhibit 13, '591 Patent, 3:43–52 (emphasis added).) As shown, the intrinsic record provides clear guidance for the word "low" such that a POSITA would have understood the scope of the claim language "low level user mode function" with reasonable certainty. (Ex. 4, Medvidovic Decl., ¶¶64-66.) *See, e.g., One-E-Way, Inc. v. ITC*, 859 F.3d 1059, 1067 (Fed. Cir. 2017); *Sonix Tech. Co. v. Publ'ns Int'l, Ltd.*, 844 F.3d 1370, 1379 (Fed. Cir. 2017); *BASF Corp. v. Johnson Matthey Inc.*, 875 F.3d 1360, 1365 (Fed. Cir. 2017).

CrowdStrike's argument that the term "low level user mode function" "conflates two functions on opposite sides of the privilege spectrum" misreads the specification. ECF 86 at 49; *see* Ex. 4, Medvidovic Decl., ¶¶65-66. The passage CrowdStrike points to states in full: "[l]ow-level languages ***can be converted to machine code without using a compiler or interpreter, and the resulting code*** runs directly on the processor." (Exhibit 13, '591 Patent, 3:52–55 (emphasis added showing CrowdStrike's omissions).) CrowdStrike's arguments about user mode functions being unable to access hardware is misleading and omits the specification's disclosure about user mode functions using system call instructions to execute kernel instructions:

> Kernel mode is generally reserved for the lowest-level, most trusted functions of the operating system. In user mode, the executing code has no ability to directly access hardware or reference memory. Code running in user mode must delegate to system APIs to access hardware or memory. Most of the code running on a processing device such as a computer will execute in user mode. ***Thus, an instruction from user mode involves system call instructions (e.g., triggering operating system dependent software interrupt or control transfer instructions such as a sysenter assembly instruction) to execute a kernel function***.

(*Id.* at 3:61–4:4 (emphasis added); *see also id.* at 10:21–26.)

Moreover, that "low level user mode function" is not indefinite is confirmed by other Defendants' including Trend Micro. Notably none of the other defendants except CrowdStrike contend that they cannot understand the scope of "low level user mode function." And Trend Micro in particular has proposed a claim construction of "stack walk processing" which explicitly includes the term "lowest level user mode function": "a process that walks a stack for a hooked function from a ***lowest level user mode function*** to an address of the invoked/originating caller function." ECF 86 at 46, 49 (emphasis added). Surely a POSITA would understand the scope of "lower level user mode function" if other defendant competitors like Trend Micro do.

Furthermore, CrowdStrike provided no expert testimony in support of its indefiniteness contentions, and thus its arguments are facially deficient. Courts have found that a party has not met its burden of proving indefiniteness where it "does not provide or rely upon any expert for its proposition that the term is indefinite or that one of ordinary skill in the art would not be reasonably certain as to the scope of [the] claim element." *Freeny*, No. 2:13-CV-791-RSP, 2015 WL 294102, at *14; *see also Luminati*, No. 2:18-CV-00483-JRG, 2019 WL 6683268, at *9; *Akzo Nobel Coatings*, 811 F.3d at 1345. CrowdStrike cannot now attempt to prove indefiniteness by clear and convincing evidence by belatedly offering evidence in a reply brief. *See, e.g.*, *Jackson*, 426 F.3d at 304 n.2; *TracBeam, LLC*, 2015 WL 5786449, at *2.

Accordingly, CrowdStrike has failed to prove by clear and convincing evidence that "low level user mode function" is indefinite.

F. "at least one processor …" ('591 patent, claims 9-16)

Plaintiffs	Trend Micro
Not subject to § 112(f), not indefinite	Subject to § 112(f), indefinite

The "at least one processor connected with the memory configured to perform operation of" particular functions including "monitoring," "invoking," "executing" and "performing" in '591 claim 9 is not subject to means-plus-function construction under § 112(f) and is not indefinite. It is undisputed that this claim language does not contain the word "means" and thus is presumed not to invoke § 112(f). *Zeroclick, LLC v. Apple Inc.*, 891 F.3d 1003, 1007 (Fed. Cir. 2018).

Trend Micro has failed to rebut the presumption against invoking § 112(f). "Claim terms 'need not connote a single, specific structure,' and may instead 'describe a class of structures' and still recite 'sufficiently definite structure' to not invoke § 112, ¶6." *Dyfan*, 28 F.4th at 1356 citing *Apple*, 757 F.3d at 1300. Here, a POSITA would have understood the claimed "processor" to refer to a class of structures well known in the art. (Ex. 4, Medvidovic Decl., ¶¶68-69.) "'[P]rocessor' is not a 'nonce' term but rather connotes a class of structures." *Cellular Commc'ns Equip. LLC v. AT&T, Inc.*, No. 2:15-cv-576, 2016 WL 7364266, at *15-16 (E.D. Tex. Dec. 19, 2016) (internal citation omitted). "A 'processor' in the computer arts is commonly understood to refer to the component of a computer that executes software instructions and performs computations." *Typemock, Ltd. v. Telerik, Inc.*, No. 17-cv-10274, 2018 WL 4189692, at *7 (D. Mass Aug. 31, 2018). "[A]s …courts have noted, a processor generally refers to a tangible object that can be purchased and that can perform certain functions even without specific instructions. Thus, unlike terms such as 'means,' 'element,' and 'device' that typically do not connote structure, 'processor' can on its own recite at least some structure to persons of ordinary skill in the art." *Fisher-*

Rosemount Sys., Inc. v. ABB Ltd., No. 4:18-cv-178, 2019 WL 6830806, at *16 (S.D. Tex. Dec. 12, 2019) (internal citation omitted). Here is no different.

Furthermore, claims 9 through 16 connote sufficient structure about the claimed "processor." A POSITA would have been able to reasonably discern from the claim language that "processor" merely referred to conventional processors existing in the prior art at the time of the invention. (Ex. 4, Medvidovic Decl., ¶¶68-69.) "The term processor is 'not used as generic terms or black box recitations of structure or abstractions, but rather as [a] specific reference' to processors that are known in the art." *Fisher-Rosemount*, No. 4:18-cv-178, 2019 WL 6830806, at *16 (quoting *Zeroclick*, 891 F.3d at 1008); *see also VDPP LLC v. Vizio, Inc.*, 2021-2040, 2022 WL 885771, at *3–4 (Fed. Cir. Mar. 25, 2022) ("storage" and "processor" not means-plus-function limitations); *see also Zeroclick, LLC v. Apple Inc.*, 891 F.3d 1003, 1008 (Fed. Cir. 2018) (same for "program" and "user interface code"); *Dyfan*, 28 at 1369 (same for "code," "application," and "system").

Even if (incorrectly) construed under Section 112(f), Trend Micro has not proved the "processor" claim language indefinite by clear and convincing evidence. The '591 specification provides sufficient structure for the processor configured to perform the claimed operations, including, for example, "monitoring a memory space of a process…" and "executing stack walk processing.…" The '591 Patent states that "[l]ow-level languages can be converted to machine code without using a compiler or interpreter, ***and the resulting code runs directly on the processor***." (Exhibit 13, '591 Patent, 3:52–55 (emphasis added).) In another example, the '591 Patent states: "method 300 may be executed on a device comprising ***at least one processor configured to store and execute operations, programs or instructions***." (*Id.* at 10:9–11 (emphasis added); *see also, e.g., id.* at 11:27–40; Ex. 4, Medvidovic Decl., ¶¶68-71.) Furthermore, a POSITA

would have understood the ’591 specification to disclose an algorithm with which a processor can be configured to determine perform the operations claimed in claims 9 through 16. (*See, e.g.,* Exhibit 13, ’591 Patent, 4:36–49; *id.* at 5:23–53; *id.* at 5:54–6:3; *id.*at FIGS. 2 and 3; Ex. 4, Medvidovic Decl., ¶¶68-69.)

G. “The system according to claim 9, further comprising identifying an address associated with the invoked function; and when the address of the invoked function is found in a cache, and wherein the processor further executing an operation of preventing execution of a payload for the invoked function” (’591 patent, claim 10)

Plaintiffs	Trend Micro
Not indefinite, plain and ordinary meaning	Indefinite

Trend Micro has failed to show by clear and convincing evidence that dependent claim 10 is indefinite. Trend Micro’s sole theory is that claim 10 is a “system” claim that recites a method step and is thus invalid for combining apparatus and method steps under *Rembrandt Data Techs., LP v. AOL, LLC*, 641 F.3d 1331 (Fed. Cir. 2011). ECF 86 at 55. Initially, Trend Micro’s indefiniteness arguments are deficient because it provides no supporting testimony. *Freeny*, No. 2:13-CV-791-RSP, 2015 WL 294102, at *14; *Luminati*, No. 2:18-CV-00483-JRG, 2019 WL 6683268, at *9; *Akzo Nobel Coatings*, 811 F.3d at 1345. Trend Micro cannot now attempt to prove indefiniteness by clear and convincing evidence by belatedly offering evidence in a reply brief. *See, e.g.*, *Jackson*, 426 F.3d at 304 n.2; *TracBeam, LLC*, 2015 WL 5786449, at *2.

Trend Micro’s contention is also mistaken because Claim 10 does not recite method steps. Like claim 9, claim 10 uses “active verbs” to “represent permissible functional language used to describe capabilities” of the claimed processor as a structure. *Mastermine Software, Inc. v. Microsoft Corp.*, 874 F.3d 1307, 1315 (Fed. Cir. 2017). Claim 10 in its entirety states:

> 10. The system according to claim 9, further comprising identifying an address associated with the invoked function; and when the address of the invoked

> function is found in a cache, and wherein the processor further executing an operation of preventing execution of a payload for the invoked function.

Claim 10 depends from Claim 9, which is also a "system" claim. In relevant part, claim 9 recites "at least one processor connected with the memory configured to perform operation of" a series of functions including "monitoring a memory space…," "invoking one of the plurality of functions…," and "executing stack walk processing…."

A person having ordinary skill in the art would understand in the context of claims 9 and 10 that the active verbs used, including the "identifying" and "executing" verbs in claim 10 are all simply examples of functional language used to describe capabilities of the claimed processor as structure. (Ex. 4, Medvidovic Decl., ¶¶75-77.) The claim 9 "processor" is configured to perform operation of "invoking one of the plurality of functions" and as added by claim 10 also configured to perform operation of "identifying an address associated with the invoked function."

The subsequent language in claim 10 confirms this. Claim 10 expressly refers to two operations, "identifying" an address of an invoked function" and "executing an operation of preventing execution of a payload" when "the address of the invoked function is found in a cache." It expressly recites the processor further executing an operation when the address of the invoked function is found (*i.e.*, the address of the invoked function is identified) in a cache. In context, the reference to the processor further executing implies that the preceding verb, "identifying" the address of the invoked function, is also an operation the processor is configured to perform. (Ex. 4, Medvidovic Decl., ¶¶75-77.) Trend Micro concedes that the "executing" claim language in claim 10 is tied to the processor, but ignores that the phrase "identifying an address associated with the invoked function" is also tied to the processor. *See* ECF 86 at 55. No skilled artisan would view the functional language of claim 10 as not tied to the functional capabilities of the processor

as a structure. All the other defendants realize this, as none of them have alleged that claim 10 is indefinite.

Trend Micro's indefiniteness arguments are also insufficient for failing to provide supporting expert testimony. *Freeny*, No. 2:13-CV-791-RSP, 2015 WL 294102, at *14; *Luminati*, No. 2:18-CV-00483-JRG, 2019 WL 6683268, at *9; *Akzo Nobel Coatings*, 811 F.3d at 1345. Trend Micro cannot now attempt to prove indefiniteness by clear and convincing evidence by belatedly offering evidence in a reply brief. *See, e.g.*, *Jackson*, 426 F.3d at 304 n.2; *TracBeam, LLC*, 2015 WL 5786449, at *2. And "unlike the claims in *Rembrandt*, the functional language here does not appear in isolation, but rather, is specifically tied to structure," that is, the processor used for purposes of identifying. *Mastermine Software*, 874 F.3d at 1316. Accordingly, Trend Micro fails to show by clear and convincing evidence that claim 10 is indefinite for reciting method steps in a system claim.

VI. '844 PATENT

The '844 Patent is entitled "Automatic Threat Detection of Executable Files Based on Static Data Analysis." The '844 Patent explains that conventional approaches for diagnosing potential malware threats were costly and time consuming, making it difficult to realistically address zero-day threats for all of the files entering a system. These "[a]pproaches to detecting threats typically focus[ed] on finding malicious code blocks within a file and analyzing the behavior of the file." (*See* Exhibit 16, '844 Patent, 2:15-17.) Encrypted files would be decrypted then disassembled to extract the code for analysis, typically by traditional anti-virus software based on signature matching. (*Id.* at 2:15-20.) If the code was malware, investigating its behavior involved running the code on the system, which put the system at risk. (*Id.* at 2:20-23.)

The ’844 Patent significantly improves upon and address shortcomings associated with prior approaches. The Patents describe and claim methods and systems that detect threats in executable files by extracting “static data points inside of the executable file without decrypting or executing the file,” generating “feature vectors” from those static data points, selectively turning on or off features of the feature vector, and then evaluating the feature vector to determine if the file is malicious. (*See, e.g.*, Exhibit 16, ’844 Patent, 1:20-21; cl. 1.) The described system and methods enable accurate and efficient identification of malware without the need to distinguish between encrypted files and non-encrypted files (*id.* at 6:58-59), thereby significantly increasing efficiency and reducing processing resources required to analyze each potentially malicious computer object. (*See* Exhibit 16, ’844 Patent, 2:46-56.)

A. “an executable file” (’844 patent, claims 1 -20)

Plaintiffs	CrowdStrike
Plain and ordinary meaning	“an encrypted or packed computer file or program”

Only CrowdStrike proposes a construction for this term. Plaintiffs and four of the Defendants agree that this term should be given its plain and ordinary meaning.

CrowdStrike’s proposed construction should not be adopted because it departs from the plain and ordinary meaning, conflicts with the intrinsic evidence, and improperly reads out disclosed embodiments. *Anchor Wall Sys., Inc. v. Rockwood Retaining Walls, Inc.*, 340 F.3d 1298, 1308 (Fed. Cir. 2003) (“[I]t is axiomatic that a claim construction that excludes a preferred embodiment … ‘is rarely, if ever correct and would require highly persuasive evidentiary support.’”) (quoting *Vitronics*, 90 F.3d at 1583). CrowdStrike’s construction conflicts with the plain and ordinary meaning of the term “executable file.” An “executable file” is simply a file that can be executed. (See Exhibit 16, ’844 Patent, 2:4–6 (“The present disclosure is able to detect whether an executable file is harmful or benign before the executable file is actually executed on

a processing device.”).) An executable file that can be executed does not have to be encrypted or packed. Defendants’ proposed construction thus conflicts with the plain meaning of the claim language by confining the term “executable file” to the subset of executable files that are encrypted or packed.

CrowdStrike’s proposed construction further conflicts with the ‘844 specification which makes clear that an executable file can be encrypted or non-encrypted and also packed or not packed. For example, the specification expressly states, “Examples of the present disclosure need not distinguish between encrypted files and non-encrypted files.” (Exhibit 16, ’844 Patent, 6:58–59.) The specification further includes multiple passages referring to encrypted and non-encrypted examples of executable files. (*Id.* ‘844 patent 2:33-38 “the creation of a training set that adaptively learns what static data points may indicate the existence of malicious code,” which requires “a statistically significant number of both encrypted and non-encrypted examples of malicious or unwanted executable files….”); 6:61-66 (the “learning classifier component” trains on a “statistically significant number of encrypted and non-encrypted examples, as well as compressed and decompressed files” so that it “may be trained to identify features in the extracted data to identify malicious files.”); 2:38-45 (“the present disclosure ensures that its adaptive learning processing is robust enough to comprise sufficient representation of features (and distributions) from files that may be encrypted, not-encrypted, compressed, and uncompressed, among other examples.”).)

This variety of training data—including non-encrypted executable files—decreases the false positive rates—one potential benefit of embodiments of the invention over prior art:

> By collecting a large set of training examples as they appear "in the wild," a sufficient representation of features and distributions associated with the represented features exist **in a training set that are *both encrypted and not-encrypted***. The present disclosure provides at least one benefit over other threat

> detection applications/programs by utilizing a very large and diverse training sample, thereby enabling a more intelligent and adaptive learning classifier that is able to achieve high detection rates for threats and low false positive rates, among other benefits.

(*Id.* at 7:5–14.)

Contrary to Defendants' contention, the prosecution record contains no clear and unmistakable disclaimer of non-encrypted and/or unpacked executable files. *Seachange Int'l, Inc. v. C-COR, Inc.*, 413 F.3d 1361, 1373 (Fed. Cir. 2005) ("A disclaimer must be clear and unambiguous."). Defendants contend that the Applicant did not dispute the Examiner's statement that "the executable file may not be decrypted or unpacked thus hiding any encrypted or packed features of the executable file from evaluation" but this does not impact claim scope for several reasons. *See* ECF 86 at 58 (citing April 20, 2018 Non-Final Rejection at 4-5). First, the Examiner's statement does not clearly or unambiguously mean that only encrypted or packed files are executable files. Indeed, the Examiner never suggested that all executable files in the '844 patent were "encrypted or packed" files in the first instance. Second, Defendants' reliance on *Sandbox* is misplaced because applicant challenged the examiner's understanding by arguing against the rejection. Importantly, the Applicants distinguished the technique of extraction—not the file type. Third, there is no clear and unmistakable disclaimer regarding the entropy value for different "chunks." The Examiner rejected claim 1 by pointing to the prior art's use of entropy analysis on random, non-overlapping file chunks of a fixed length—specifically, a chunk's decomposition rate—to compare "known bad" files because the prior art also apparently did not require unpacking or decrypting. (Exhibit 17, '844 Patent File History, at WBRT007331; *see also* Exhibit 7, Malek Decl. at ¶ 27.) Applicant distinguished the "chunk" reference by noting that the boundaries of the prior art's "chunks" used to calculate entropy were random and "do not correspond to particular sections, points of reference, or portions of the file," while the '844 Patent utilized static data points

of encrypted files and non-encrypted files that "correspond to points of reference in a received executable file. The points of reference represent sections, subsections, or categories of an executable file." (Exhibit 17, '844 Patent File History, Amendment dated September 20, 2018 at 9; *see also* Malek Decl. at ¶ 28.) Accordingly, the prosecution history does not change the plain meaning of "an executable file."

All these statements indicate that the patentee clarified how the claims distinguished themselves from the prior art by significantly increasing efficiency and by reducing processing resources required to analyze each potentially malicious computer object. (*See* Exhibit 7, Malek Decl. at ¶ 28.) Thus, the patentee confirmed he was not disclaiming the '844 Patent's ability to detect malware without the need to distinguish between encrypted files and non-encrypted files, but instead was improving upon the prior art.

Accordingly, Defendants' attempt to construe "an executable file" differently from its plain and ordinary meaning should be rejected; Defendants provide no reason for limiting the claims to "an encrypted or packed computer file or program."

B. The terms relating to whether "character strings" are "within a predetermined range" ('844 patent, claims 1, 10, and 17)

Plaintiffs	Sophos/Trend Micro/ForcePoint
Not indefinite. Plain and ordinary meaning	Indefinite.

Two of the Defendants apparently agree with Plaintiffs that these terms should be given their plain and ordinary meaning and are not indefinite.

Claim 1 of the '844 Patent recites a "plurality of static data points" that "represent predefined character strings in the executable file" and also recites selectively turning on or off features of a feature vector "based on whether a value of one or more static data points from the plurality of extracted static data points is within a predetermined range." Defendants assert that the reference to "character strings" being "within a predetermined range" in the claims is indefinite based on a mistaken assumption: "[t]he extracted information cannot simultaneously be a character

string and a numerical value." ECF 86 at 59. That assumption contradicts the specification and the most basic precepts of computer science and programming because it is well-known that character strings can be numerical values. (Exhibit 7, Malek Decl. at ¶¶ 31–33.)

A person of ordinary skill would understand that "character strings" can be numerical values. For example, "strings" are used interchangeably with "byte sequences" in the very table that Defendants quote to summarize the '844 patent's categorization of the "different types of information (e.g., static data points) that can be extracted from an executable file." ECF 86 at 59–60. (Exhibit 16, '844 patent, 6:23–58; *see also id.* at 8:29-48 (describing generating "counts" of "n-grams" made up of "string and/or byte sequences" as "input to a generated feature vector").) To a POSITA, "byte sequences" are inherently numeric because they consist of a fixed-length sequence of 8-bit binary numbers, known as "bytes." (Exhibit 7, Malek Decl. at ¶¶ 31–33.) Moreover, because a POSITA would have known that nearly all programming languages readily permit conversion of "character strings" into "numeric values," and that computers in fact encode character text as numeric values to begin with, a POSITA would be able to readily understand with reasonable certainty what it means for "character strings" to fall within a "predetermined range," (*Id.*) and would not view that language as "contradictory and internally inconsistent" as Defendants' assert. ECF 86 at 62.

In addition, Defendants' assumption that the predetermined range has to be a numerical range is also inconsistent with the '844 specification. The patent discloses that the "legal information" extracted from a file, which according to Defendants' is non-numeric, may consist of at least four distinct categories of strings: "timestamp data, licensing information, copyright information, [and] indication of intellectual property protection." (Exhibit 16, '844 patent, 8:1-5.) This, in Defendants' parlance, would by definition constitute a non-numeric "predetermined range": if the strings collected from an executable file fall into one of these four categories, then the claimed methods and systems of the '844 patent could turn "on," and test for, features related to "an evaluation of the 'legal information.'" On the other hand, if the strings extracted from the executable file do not fall into the range of legal information just described, "this may trigger the

learning classifier to adaptively check for additional features that might be indicative of a threat or malicious file as well as turn off other features related to an evaluation of 'legal information.'" (Exhibit 16, '844 patent, 8:1-21.)

Accordingly, Defendants have failed to prove by clear and convincing evidence that the claim language referring to "character strings" being "within a predetermined range" is indefinite.

C. "within a predetermined range" ('844 patent, claims 1, 10, and 17)

Plaintiffs	Sophos/ForcePoint
Plain and ordinary meaning	"between a predefined lower and upper limit"

The claim language "within a predetermined range" should be given its plain and ordinary meaning and does not require construction. By contrast, the proposed construction from two of the Defendants should be rejected because it is inconsistent with the intrinsic evidence and improperly seeks to read limitations from the specification into the claims.

Initially, Defendants' construction is inconsistent with the plain and ordinary meaning of the claim language and the intrinsic record. Under Defendants' construction, the process requires at least three data points: (1) a numeric value at a lower limit; (2) a numeric value at an upper limit; and (3) the numeric values of static data points. But predetermined ranges are not restricted to numeric values on a spectrum. Indeed, the claim language specifically refers to "whether a value of one or more static data points…is within a predetermined range" without narrowing the "value" to a numeric value. For example, the '844 specification discloses an example where the classifiers are trained on static data points that "include but are not limited to: header information, section information, import and export information, certificate information, resource information, string and flag information, legal information, comments, and/or program information (*e.g.*, APIs and DLLs), among other examples. (Exhibit 16, '844 Patent, 5:63 –6:1.) These data points have a value that may or may not be a numerical value and do not have a numerical upper or lower limit.

The specification describes static data points based on the legal information as one example of a predetermined range that is not defined by a numerical upper or lower limit. The specification discloses, "[i]n an example where the learning classifier detects the presence of legal information does not exist in the executable file, this may trigger the learning classifier to adaptively check for additional features that might be indicative of a threat or malicious file as well as turn off other features related to an evaluation of the 'legal information.'" (*Id.* at 8:5–11.) The "legal information" detected by the learning classifier includes "any identifying information indicating data that conveys rights to one or more parties including: timestamp data, licensing information, copyright information, indication of intellectual property protection, etc." (*Id.* at 8:1–5.) Thus, the "predetermined range" in the case of "legal language" in the body of the executable file is that if such language is present, then the learning classifier allows certain other features of the vector to be turned off. Such a determination does not require three numeric data points, but rather the "predetermined range" in this disclosed instance is whether or not the executable file includes legal information.

Defendants' proposed construction is based entirely on an improper attempt to read limitations from the specification into the claims. Defendants do not point to any clear and unambiguous definition or disavowal in the intrinsic record in their effort to limit a "predetermined range" to numerical values between an upper and lower limit. Instead, Defendants seek to limit the claim language to what they (incorrectly) characterize as the "sole example" of a predetermined range. ECF 86 at 64. The Federal Circuit has repeatedly held that it is improper to limit the claims to embodiments disclosed in the specification. *Phillips* 415 F.3d at 1323; *Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 906 (Fed. Cir. 2004) (requiring "words or expressions of manifest exclusion or restriction" before limiting claims to a specific embodiment). This canon holds even

when the patent discloses only a single embodiment. Accordingly, Defendants' construction is inconsistent with the plain meaning of the claim language and the intrinsic evidence and should be rejected.

D. "support vector processing" ('844 patent, claims 1-20)

Plaintiffs	CrowdStrike/Sophos/Trend Micro/ForcePoint/Kaspersky
Plain and ordinary meaning	"processing with a support vector machine"

This Court should accord the term "support vector processing" its plain and ordinary meaning, which is simply "processing that involves a support vector." (Exhibit 7, Malek Decl. at ¶ 37.) Contrary to Defendants' construction, in the art of machine learning, "[S]upport vectors" are distinct from support vector machines. Specifically, "support vectors" are the "data points considered most difficult to classify," and make positive decisions about, for a given machine learning problem, and can be processed both by support vector machines and by other machine learning classifiers, such as neural networks. (*Id.* at ¶ 38.)

Defendants' proposed construction departs from the plain and ordinary meaning by adding a requirement that the "support vector processing" has to use a "support vector machine." Defendants' construction impermissibly reads out specifically disclosed embodiments of the '844 patent because it limits every independent claim of the '844 patent to the use of a "support vector machine" when the claims use the term "machine learning classifier." *Anchor Wall*, 340 F.3d 1298, 1308 (Fed. Cir. 2003) ("[I]t is axiomatic that a claim construction that excludes a preferred embodiment . . . 'is rarely, if ever correct and would require highly persuasive evidentiary support.'") (quoting *Vitronics*, 90 F.3d at 1583). The specification of the '844 patent discloses, however, that any number of machine learning classifiers may embody this limitation, including an "artificial neural network (ANN), a decision tree, association rules, inductive logic, a support

vector machine, clustering analysis, and Bayesian networks." (Exhibit 16, '844 patent, 8:17–21; Exhibit 7, Malek Decl. at ¶ 36.)

Finally, Defendants' citations to excerpts of the prosecution history in which applicants "did not rebut" certain of the Examiner's statements about prior art reciting a support vector machine do not legally affect a disclaimer of claim scope. It is black-letter law that "[a] applicant's silence in response to an examiner's characterization of a claim does not reflect the applicant's clear and unmistakable acquiescence to that characterization if the claim is eventually allowed on grounds unrelated to the examiner's unrebutted characterization . . . . After all, the applicant has disavowed nothing." *Salazar v. Procter & Gamble Co.*, 414 F.3d 1342, 1345 (Fed. Cir. 2005) (quoting *3M Innovative Props. v. Avery Dennison Corp.*, 350 F.3d 1373, 1374 (Fed. Cir. 2003)); *see also Biogen Idec, Inc. v. GlaxoSmithKline LLC*, 713 F.3d 1090, 1099-100 (Fed. Cir. 2013) ("[I]t is the applicant, not the examiner, who must give up or disclaim subject matter that would otherwise fall within the scope of the claims, and an applicant's silence regarding statements made by the examiner during prosecution cannot amount to a clear and unmistakable disavowal of claim scope.") Accordingly, Defendants' attempt to construe "support vector processing" in a manner that is inconsistent with its plain and ordinary meaning and the intrinsic record should be rejected.

E. "at least one processor …" ('844 patent, claims 17-20)

Plaintiffs	Trend Micro
Not subject to § 112(f); Not indefinite	Subject to § 112(f); Indefinite

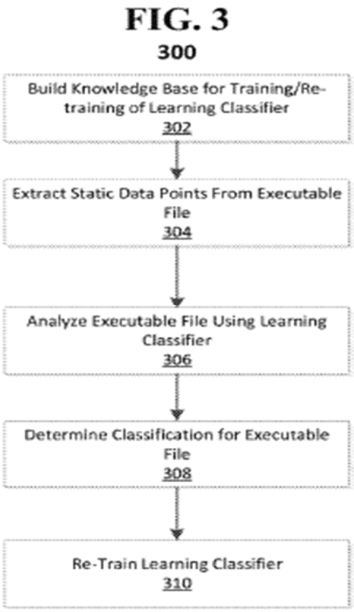
The absence of the word "means" creates a presumption against invoking § 112(f). *See Williamson,* 792 F.3d at 1348; *Media Rights Techs., Inc. v. Capital One Fin. Corp.*, 800 F.3d 1366, 1371 (Fed. Cir. 2015). Trend Micro has failed to overcome the presumption that means-plus-function construction is inapplicable for much the same reasons stated in connection with the "processor" claim language in the '591 patent.

Initially, Trend Micro cannot show that the "at least one processor…" claim language fails to recite sufficiently definite structure or else recites function without reciting sufficient structure for performing that function. *Williamson*, 792 F.3d at 1349. "Claim terms 'need not connote a single, specific structure,' and may instead 'describe a class of structures' and still recite 'sufficiently definite structure' to not invoke § 112, ¶6." *Dyfan*, 28 F.4th at 1356 citing *Apple Inc. v. Motorola, Inc.*, 757 F.3d 1286, 1300 (Fed. Cir. 2014). Here, a POSITA would have understood the claimed "processor" to refer to a class of structures well known in the art. (Exhibit 7, at ¶ 42-49); (Ex. 4, Medvidovic Decl., ¶¶68-69.) "'[P]rocessor' is not a 'nonce' term but rather connotes a class of structures." *Cellular Commc'ns Equip. LLC v. AT&T, Inc.*, No. 2:15-cv-576, 2016 WL 7364266, at *15-16 (E.D. Tex. Dec. 19, 2016) (internal citation omitted).; *see, e.g., Cellular Commc'ns*, 2016 WL 7364266, at *15-16. "A 'processor' in the computer arts is commonly understood to refer to the component of a computer that executes software instructions and performs computations.") *Typemock,* 2018 WL 4189692 at *7. "[A]s …courts have noted, a processor generally refers to a tangible object that can be purchased and that can perform certain functions even without specific instructions. Thus, unlike terms such as 'means,' 'element,' and 'device' that typically do not connote structure, 'processor' can on its own recite at least some structure to persons of ordinary skill in the art." *Fisher-Rosemount*, 2019 WL 6830806, at *16.

More particularly, the "at least one processor…" claim language connotes specific in view of the claims and specification. They describe the inputs and outputs to the processor, that the processor performs particular operations, and that the processor is configured to perform those particular operations. (Exhibit 7, at ¶ 42-49). Claim 17 recites "at least one processor operatively connected with the memory and configured to perform operation of [1] extracting a plurality of predefined character strings from an executable file without decrypting or unpacking the

executable file; [2] generating a feature vector from the plurality of predefined character strings using a classifier trained to classify the plurality of predefined character strings based on a collection of data comprising known malicious executable files, known benign executable files and known unwanted executable files…and [3] evaluating the feature vector using support vector processing to determine whether the executable file is harmful." Claim 17 thus describes the claimed "at least one processor" as being one configured to perform particular operations and a POSITA would understand the claimed processor to be a processor programmed to perform the claimed operations as structure. (Exhibit 7, at ¶ 45-49.)

The specification further includes a flowchart that describes the processor programmed to perform the claimed operations.



(Exhibit 16, '844 patent, Fig. 3.) The specification further describes the processor configured to perform the "extracting," "generating" and "evaluation" operations. (*See, e.g.,* Exhibit 16, '844 patent, 7:66-8:11; 9:2–40.) For example, with regard to the "extracting" operation, the

specification discloses that "[w]hen an executable file is identified for evaluation, flow proceeds to operation 304 …[in which] an executable file is analyzed using machine learning processing…to collect/extract static data points from an executable file for evaluation." (Exhibit 16, '844 patent, 12:6–23.) It further discloses that those static data points, which include the pre-defined character "strings" appearing in Claim 17, may be extracted through "Programming operations …to collect file data[], parse the file data, and store extracted data points." (*Id.* at 5:8–13; Table 1.1.) Such data extraction, parsing, and storage programming techniques would have been well-known to a POSITA. (Exhibit 7, Malek Decl. at ¶ 46.)

The '844 patent also describes a processor that is configured to generate "a feature vector from the plurality of predefined character strings using a classifier trained to classify the plurality of predefined character [.]" (Exhibit 16, '844 patent, cl. 17.) For example, "operation 306" may "generate the feature vector for the learning classifier" by "par[sing] and encod[ing]" the extracted data for "machine learning processing." (*Id.* at 12:48–53.) The specification describes the details of this processing as well as how that processing generates a feature vector from the predefined character strings extracted from an executable file: "As an example, string sequence data as well as byte-sequence data may be parsed and processed as n-grams and/or n-gram word prediction (e.g. word-grams).… The resulting counts of the unique n-grams are then used as input to a generated feature vector.… All of the different encoded data may be combined into one of more final feature vectors, for example after a complex vector coding/processing is performed *(e.g.*, L2 norm)." (*Id.* at 8:22–47.)

Contrary to Defendants' unsupported assertions, the '844 patent describes to a POSA a processor that operates to selectively turn on or off a feature and how a predetermined range of character strings is obtained. For example, Table 1.1 of the '844 patent explicitly describes 22

categories of "Strings" that may be extracted from the executable file, including the "company name," "file description," "Comments," "product name" and "legal copyright." (Exhibit 16, '844 patent at Table 1.1.). The specification describes when features of a feature vector are "selectively turned on or off based on whether a value of one or more predefined strings" "is within a predetermined range." For example, when the learning classifier detects "that the file does not contain 'legal information'," such as "timestamp data, licensing information, copyright information, etc.," it may "adaptively check for additional features that might be indicative of a threat" while "turn[ing] off features related to an evaluation of the "legal information." (*See id.* at 7:66–8:11.) Given this disclosure, a POSITA would readily ascertain that the claimed "at least one processor…" as a "processor[s]" structure that operates to generate a feature vector "wherein one or more features of the feature vector are selectively turned on or off based on whether a value of one or more predefined character strings from the plurality of predefined character strings is within a predetermined range." (Exhibit 7, Malek Decl. at ¶ 48.)

The '844 Patent likewise describes a processor structure that is configured to perform the claimed "evaluating" operation. (Exhibit 16, '844 patent, 9:2-40; Exhibit 7, Malek Decl. at ¶ 49). For example, the specification states, "[i]n evaluation using the classifier, the features related to the data points may be weighted by the particular data point being evaluated or by category, for example. For instance, the training of the learning classifier may indicate that legal information (e.g. "legal copyright") provides a better indication that a file may be malicious than that of a section name." (Exhibit 16, '844 patent, 9:2–40.) The specification describes several possible machine learning classification algorithms that a POSITA can use to process and evaluate the feature vector appearing in Claim 17, such as support vector machines and "decision tree[s] based on three-class logistic gradient boosting." (*Id.* at 9:40–66; *see also* Exhibit 7, Malek Decl. at ¶ 49.)

Like the "at least one processor…" language in claim 17, the claims and specification also describe the inputs and outputs to the processor, that the processor performs particular operations, and that the processor is configured to perform those particular operations for dependent claims 18-20. **(**Exhibit 7, Malek Decl. at ¶ 50-52.) For example, claims 18 and 20 add a "preventing execution" operation and claim 19 adds a "classifying" operation. In each case, the claims and specification describe a processor programmed to perform the claimed operation. *See e.g.,* (Exhibit 16, '844 Patent 13:20-32; Exhibit 7, Malek Decl. at ¶¶ 50, 52 ("preventing execution" claims 18, 20); Exhibit 16, '844 Patent, 12:26-35, 8:22-47; Exhibit 7, Malek Decl. at ¶ 51 ("classifying" claim 19).

Defendants' expert, Dr. Almeroth, ignores the intrinsic evidence in his conclusory assessment. ECF 86, Ex. U-5 (Almeroth Decl.) at ¶¶ 116, 118, 119. Given the '844 Patent claims and specification, a POSITA would understand the "at least one processor …" claim language to connote sufficiently definite structure that performs the claimed operations. (Exhibit 7, Malek Decl. at ¶ 47.) Accordingly, Trend Micro has failed to rebut the presumption that the "at least one processor…" claim language in the '844 patent is subject to §112(f) construction.

DATED: November 18, 2022

Respectfully submitted,

By*:/s/ Jeffrey D. Mills*

Jeffrey D. Mills
Texas Bar No. 24034203
KING & SPALDING LLP
500 West Second St.
Suite 1800
Austin, Texas 78701
Telephone: (512) 457-2027
Facsimile: (512) 457-2100
jmills@kslaw.com

Mark D. Siegmund
STECKLER WAYNE
CHERRY & LOVE, PLLC
8416 Old McGregor Rd.
Waco, Texas 76712
Telephone: (254) 651-3690
Facsimile: (254) 651-3689
mark@swclaw.com

Steve Sprinkle
Texas Bar No. 00794962
SPRINKLE IP LAW GROUP, P.C.
1301 W. 25th Street, Suite 408
Austin, Texas 78705
TEL: 512-637-9220
ssprinkle@sprinklelaw.com

Christopher C. Campbell (DC Bar No. 444262)
Patrick M. Lafferty (pro hac vice)
KING & SPALDING LLP
1700 Pennsylvania Avenue, NW
Suite 200
Washington, DC 20006
Telephone: (202) 626-5578
Facsimile: (202) 626-3737
ccampbell@kslaw.com
plafferty@kslaw.com

Britton F. Davis (pro hac vice)
Brian Eutermoser (pro hac vice)
KING & SPALDING LLP
1401 Lawrence Street
Suite 1900.
Denver, CO 80202
Telephone: (720) 535-2300
Facsimile: (720) 535-2400
bfdavis@kslaw.com
beutermoser@kslaw.com

Attorneys for Plaintiffs Open Text, Inc. and Webroot, Inc.

CERTIFICATE OF SERVICE

The undersigned certifies that on November 18, 2022, all counsel of record who are deemed to have consented to electronic service are being served with a copy of this document through the Court's CM/ECF system under Local Rule CV-5(b)(1).

/s/ Jeffrey D. Mills
Jeffrey D. Mills