FILED
October 24, 2022
CLERK, U.S. DISTRICT COURT
WESTERN DISTRICT OF TEXAS
BY: CV
DEPUTY

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

Webroot, Inc. and Open Text, Inc., ***Plaintiffs,*** **v.** **Sophos Ltd.,** ***Defendant.***	**Civil Action No.** **6:22-cv-00240-ADA** **JURY TRIAL DEMANDED**
Sophos Ltd., ***Counterclaim-Plaintiff,*** **v.** **Webroot, Inc., Open Text, Inc., and Open Text Corp.,** ***Counterclaim-Defendants.***	

SECOND AMENDED COMPLAINT FOR PATENT INFRINGEMENT

Pursuant to Federal Rule of Civil Procedure 15(a) (1) and United States District Court for the Western District of Texas Local Rule CV-15, Plaintiffs Webroot, Inc. ("Webroot") and Open Text, Inc. ("Open Text") (collectively "Plaintiffs") hereby submit their First Amended Complaint for Patent Infringement against Defendant Sophos Ltd. ("Sophos" or "Defendant") and allege the following:

1. This case involves patented technologies that helped to revolutionize, and have become widely adopted in, the fields of malware detection, network security, and endpoint protection. Endpoint protection involves securing endpoints or entry points of end-user devices

PGR2023-00031
Open Text Inc. EX-2011

(*e.g.*, desktops, laptops, mobile devices, etc.) on a network or in a cloud from cybersecurity threats, like malware.

2. Before Plaintiffs' patented technologies, security platforms typically relied on signatures (*i.e.*, unique identifiers) of computer objects (*e.g.*, computer programs) that were analyzed and identified as "bad" by teams of threat researchers. This approach required antivirus companies to employ hundreds to thousands of threat analysts to review individual programs and determine if they posed a threat.

3. The "bad" programs identified by researchers were compiled into a library and uploaded to an antivirus software program installed on each endpoint device. To detect threats, a resource intensive "virus scan" of each endpoint device was conducted. These virus scans could take hours to complete and substantially impact productivity and performance.

4. Despite substantial investments in resources and time, the conventional systems still were unable to identify and prevent emerging ("zero-day") threats from new or unknown malware. New threats persisted and were free to wreak havoc until a team of threat analysts could identify each one and upload these newly identified threats to an update of the "bad" program library. The updated "bad" program library, including signatures to identify new threats as well as old, then had to be disseminated to all of the endpoint computers, which required time and resource consuming downloads of the entire signature library to every computer each time an update was provided.

5. By the early-to-mid 2000s, new threats escalated as network connectivity became widespread, and programs that mutate slightly with each new copy (polymorphic programs) appeared. These events, and others, rendered the traditional signature-based virus scan systems ineffective for these modern environments.

6. Plaintiffs' patented technology helped transform the way malware detection and network security is conducted, reducing and often even eliminating the shortcomings that plagued signature-based security products that relied on human analysts.

7. Instead of relying on human analysts, Plaintiffs' patented technology enabled the automatic and real-time analysis, identification, and neutralization of previously unknown threats, including new and emerging malware, as well as advanced polymorphic programs.

8. For example, Plaintiffs' patented technology uses information about the computer objects being executed—including, for example, information about the object's behavior and information collected from across a network—along with machine learning technology and novel system architectures, to provide security systems that are effective in identifying and blocking new security threats in real-time in real-world, commercial systems.

9. Plaintiffs' patented technology further includes new methods of "on execution" malware analysis; new architectures that efficiently and effectively distribute workloads across the network; new forensic techniques that enable fast, efficient, and accurate analysis of malware attacks; and new advanced memory scanning techniques.

10. Plaintiffs' patented technology makes security software, platforms, and appliances better at detecting malware by, for example, reducing false positives/negatives and enabling the identification and mitigation of new and emerging threats in near real-time. These improvements are accomplished while at the same time reducing the resource demands on the endpoint computers (*e.g.*, not requiring downloading and using full signature databases and time-consuming virus scans).

11. Plaintiff Webroot has implemented this technology in its security products like Webroot SecureAnywhere AntiVirus, which identifies and neutralizes unknown and undesirable

computer objects in the wild in real-time.

12. Over the years, Plaintiff Webroot has also received numerous accolades and awards for its products and services. For example, Webroot has received 22 PC Magazine Editor's Choice Awards, including "Best AntiVirus and Security Suite 2021." That same year, Webroot also received the Expert Insights Best-of-Endpoint Security award.

13. Plaintiffs currently own more than 70 patents describing and claiming these and other innovations, including U.S. Patent No. 8,418,250 (the "'250 Patent"), U.S. Patent No. 8,726,389 (the "'389 Patent"), U.S. Patent No. 9,578,045 (the "'045 Patent"), U.S. Patent No. 10,257,224 (the "'224 Patent"), U.S. Patent No. 10,284,591 (the "'591 Patent"), U.S. Patent No. 10,599,844 (the "'844 Patent"), 11, 409, 869 (the "'869 patent), and U.S. Patent No. 9,413,721 (the "'721 Patent"). (Exhibits 1-8.)

14. Plaintiffs' patented technology represents such a vast improvement on the traditional malware detection and network security systems that it has become a widely adopted and accepted approach to providing endpoint security in real-time.

15. Defendant Sophos Ltd (collectively, "Sophos") is a direct competitor of Plaintiffs and provides security software and systems that, without authorization, implement Plaintiffs' patented technologies. Sophos's infringing security software includes, but is not limited to, Intercept X Advanced with EDR and XDR, Sophos Web Appliance, Sophos XG Firewall, and Sophos Synchronized Security, (collectively, "Sophos Security Suite" or "Accused Products").

16. Plaintiffs bring this action to seek damages for, and to ultimately stop, Defendant's continued infringement of Plaintiffs' patents, including in particular the '250 Patent, the '389 Patent, the '224 Patent, the '045 Patent, the '591 Patent, the '844 Patent, and the '721 Patent (collectively, the "Asserted Patents." (Exhibits 1-7.)) As a result of Sophos's unlawful competition

in this District and elsewhere in the United States, Plaintiffs have lost sales and profits and suffered irreparable harm, including lost market share and goodwill.

NATURE OF THE CASE

17. Plaintiffs bring claims under the patent laws of the United States, 35 U.S.C. § 1, *et seq.*, for infringement of the Asserted Patents. Defendant has infringed and continues to infringe each of the Asserted Patents under at least 35 U.S.C. §§271(a), 271(b) and 271(c).

THE PARTIES

18. Plaintiff Webroot, Inc., is the owner by assignment of each of the Asserted Patents.

19. Webroot has launched multiple cybersecurity products incorporating its patented technology, including for example Webroot SecureAnywhere and Evasion Shield.

20. Webroot is a registered business in Texas with multiple customers in this District. Webroot also partners with several entities in this District to resell, distribute, install, and consult on Webroot's products.

21. Plaintiff OpenText holds an exclusive license to the Asserted Patents. OpenText is registered to do business in the State of Texas.

22. OpenText is a Delaware corporation and maintains three business offices in the state of Texas, two of which are located in this District, including one in Austin and another in San Antonio. Over 60 employees work in this District, including employees in engineering, customer support, legal and compliance teams, IT, and corporate development. OpenText also has a data center located in this District. OpenText is in the computer systems design and services industry. OpenText sells and services software in the United States.

23. Defendant Sophos Ltd. is a foreign corporation with its global headquarters at The Pentagon, Abingdon, OX14 3YP, United Kingdom.

JURISDICTION & VENUE

24. This action arises under the Patent Laws of the United States, 35 U.S.C. § 1, *et seq*. The Court has subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a).

25. This Court has personal jurisdiction over Defendant because Defendant regularly conducts business in the State of Texas and in this district, including operating systems, using software, providing services and/or engaging in activities in Texas and in this district that infringe one or more claims of the Asserted Patents.

26. Defendant Sophos has further, either directly or through its extensive network of reseller and OEM partnerships, purposefully and voluntarily placed its infringing products and/or services into the stream of commerce with the intention and expectation that they will be purchased and used by customers in this District, as detailed below.

27. Venue is proper in this District pursuant to 28 U.S.C. §§ 1391(b) and (c) and 28 U.S.C. § 1400(b) because, upon information and belief, Defendant Sophos is a foreign entity. Sophos has also committed acts of infringement within this District.

28. On information and belief, Sophos is a foreign corporation with significant contacts with this District. As an example, Sophos has entered into license agreements with end-users in Texas covering the Accused Products and their operation in this District. The Sophos Security Suite End User License Agreements all reference Sophos Limited as the rights-holder under the contract. (*See, e.g.*, https://www.sophos.com/en-us/legal/sophos-end-user-license-agreement.aspx.) Thus, Sophos has entered into license agreements with end-users covering the Accused Products and their operation in Texas and in this District.

29. On information and belief, Sophos relies on a network of partnerships with "resellers, managed service providers and cybersecurity experts" to sell Accused Products,

including Intercept X, to its customers in this District, and to instruct and teach customers how to use the Accused Products. (*See, e.g.*, https://www.sophos.com/en-us/products/endpoint-antivirus/how-to-buy.aspx ("Sophos products and services are sold via trusted partners who recommend and implement the right solutions to meet your unique needs.").)

30. On information and belief, Sophos sells, offers for sale, advertises, makes, installs, and/or otherwise provides endpoint security software and security services, including the Accused Products, the use of which infringes the Asserted Patents in this District. Sophos performs these acts directly and/or through its partnerships with resellers and managed service providers in this District. Those partners include, but are not limited to, "Gold" and "Silver" partners consisting of resellers and managed service providers in this District. (*See* https://partners.sophos.com/english/directory/search?lat=30.267153&lng=-97.7430608&dMI=100&p=1.)

31. On information and belief, Sophos generates sales to end users within the United States and within this District through its partnerships with resellers and managed service providers. (*Id.*)

32. Sophos has sold infringing endpoint security software and provided infringing endpoint security services to customers who have regular and established places of business in this District, which, on information and belief, deploy Sophos's endpoint security software to their endpoint devices and encourages others to install Sophos's antivirus software on their own devices. (*See, e.g.*, https://security.utexas.edu/education-outreach/anti-virus.)

33. As further detailed below, Sophos's use, provision of, offer for sale, sales, installation, maintenance, support, and advertising of endpoint security software within this District infringe the Asserted Patents. Sophos' partners infringe the Asserted Patents by using, installing, offering for sale, selling, providing support for, and/or advertising Sophos's endpoint

security software within this District. Sophos' customers infringe the Asserted Patents by using Sophos' endpoint security software within this District.

34. Sophos and its partners encourage and induce its partners and customers to use the Accused Products in an infringing way at least by making Sophos's endpoint security services available on its website, widely advertising those services, providing applications that allow partners and users to access those services, provides instructions for installing, and maintaining those products, and/or provides technical support to users, and engaging in activities that aid and abet infringement of the Asserted Patents by end-users. (*See* https://www.sophos.com/en-us/products/endpoint-antivirus.aspx.).

35. Sophos's partners also infringe (directly or indirectly) the Asserted Patents by installing, maintaining, operating, providing instructions and technical support, and/or advertising the Sophos Security Suite including the Accused Products within this District. End-users and Sophos's partner customers infringe the Asserted Patents at least by installing and using Sophos Security Suite software, which performs the claimed methods in the Asserted Patents within this District.

36. Sophos also contributes to infringement of the Asserted Patents by customers and end users of the Accused Products by offering within the United States or importing the Accused Products into the United States, which are for use in practicing, and under normal operation practice, one or more of the methods claimed in the Asserted Patents, constituting a material part of the inventions claimed, and not a staple article or commodity of commerce suitable for substantial non-infringing uses. Indeed, the Accused Products and the example functionality described below have no substantial non-infringing uses but are specifically designed to practice the methods claimed in the Asserted Patents.

37. Sophos' infringement adversely impacts Plaintiffs and their employees who live in this district, as well as Plaintiffs' partners and customers who live and work in and around this District. On information and belief, Sophos actively targets and offers Accused Products to customers served by Plaintiffs, including in particular customers/end-users in this District.

PLAINTIFFS' PATENTED INNOVATIONS

38. Plaintiff Webroot and its predecessors were all pioneers and leading innovators in developing and providing modern end point security protection, including "community-based" signatureless threat detection process using AI-driven behavior analysis across the entire network to provide "zero-day" protection against unknown threats.

39. The Asserted Patents discussed below capture technology, features, and processes that reflect these innovations, and improve on traditional anti-Malware and network security systems.

Advanced Malware Detection Patents
U.S. Patent Nos. 8,418,250 and 8,726,389

40. The '250 and '389 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. These patents are collectively known as the "Advanced Malware Detection" Patents. Plaintiff Webroot owns by assignment the entire right, title, and interest in and to the '250 and '389 Patents. Webroot has granted Plaintiff OpenText an exclusive license to the '250 and '389 Patents.

41. The '250 Patent is entitled "Methods and Apparatus for Dealing with Malware," was filed on June 30, 2006, and was duly and legally issued by the United States Patent and Trademark Office ("USPTO") on April 9, 2013. The '250 Patent claims priority to Foreign Application No. 0513375.6 (GB), filed on June 30, 2005. A true and correct copy of the '250 Patent is attached as Exhibit 1.

42. The '389 Patent is also entitled "Methods and Apparatus for Dealing with Malware," was filed on July 8, 2012, and was duly and legally issued by the USPTO on May 13, 2014. The '389 Patent claims priority to the same Foreign Application as the '250 Patent. A true

and correct copy of the '389 Patent is attached as Exhibit 2.

43. 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.

44. 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 manually or "semi-manually" by subjecting the file to detailed analysis, for example by emulation or interpretation, which can take days given the human involvement that is typically required. (*See*, *e.g.*, Exhibit 2, '389 Patent, 2:9-17.)

45. 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.

46. 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 involvement, 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.)

47. By contrast, the methods and systems disclosed and claimed in the ’250 and ’389 Patents perform automatic, sophisticated review (*e.g.*, “pattern analysis”) 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.

48. 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.) This provides a significant improvement to the operation of the computer network because monitoring behavior or other information about the object or process, rather than code or signature matching, allows the system to rapidly determine the nature of the object (*e.g.*, malware), without requiring a detailed manual analysis of the code of the object itself as in conventional anti-virus software. (*See* Exhibit 1, ’250 Patent, 3:11-18.)

49. The approaches in the Advanced Malware Detection Patents are generally focused on receiving *information about the behavior* of objects or processes on remote computers at a base computer. This information is analyzed automatically by, for example, mapping the behavior and attributes of objects known across the community in order to identify suspicious behavior and to identify malware at an early stage. This approach allows, among other advantages, the number of human analysts needed to be massively reduced. It also improves the computer network by reducing the latency involved with identifying new threats and responding to objects exhibiting new, potentially malevolent behavior. (’250 Patent Prosecution History, 2010-09-07 Amendment

at 16-17.)

50. Each of the claimed inventions of the Advanced Malware Detection Patents is necessarily rooted in computer technology—in other words, the identification of malicious computer code in computer networks is fundamentally and inextricably a problem experienced with computer technology and networks— and addresses this fundamental computer technology problem with a computer technology solution. Furthermore, the Advanced Malware Detection Patents improve the technical functioning of the computer network using techniques—such as analyzing behavioral information about or associated with computer objects and processes—to improve network security by identifying malware more quickly and with less resources. These technical improvements address identified weaknesses in conventional systems and processes. (*See, e.g.*, Exhibit 1, '250 Patent, 2:5-3:18.)

51. In particular, the '250 Patent describes and claims methods and systems that include receiving *behavioral data about or associated with a computer object* from remote computers on which the object or similar objects are stored; comparing in a base computer the data about the computer object received from the remote computers; and, classifying the computer object as malware on the basis of said comparison if the data indicates the computer object is malware. In effect, this process builds a central picture of objects and their interrelationships and activities across the entire community and allows automation of the process of identifying malware by aggregating and comparing the activity of objects running across the community (*i.e.*, on multiple remote computers).

52. The '250 Patent further provides that a mask is automatically generated for an object that defines "acceptable behavior" for the object. The operation of the computer object is then monitored and if the actual monitored behavior extends beyond that permitted by the mask,

the object is disallowed from running and reclassified as malware.

53. The claimed methods and systems of the ’250 Patent constitute technical improvements over the traditional anti-malware systems and provide numerous advantages to computer systems and the process of detecting malware. In addition to the advantages set forth above, the methods and systems claimed in the ’250 Patent provide additional advantages in dealing with objects that do not initially exhibit suspicious behavior, but later start to exhibit malevolent behavior. Traditional malware systems could only mark a computer object as good or bad (*i.e.*, a binary decision), and did so by examining the signature of the object itself against a database of “known bad” signatures. This approach does not permit the system to automatically deal with the case where an object does not initially exhibit suspicious behavior but starts to exhibit malevolent behavior in the future.

54. By contrast, the ’250 Patent improves these systems by generating an appropriate behavior mask for the object and then continuing to monitor the behavior of the object. If the object operates out of bounds of the permitted behavior, then an appropriate action is taken, such as disallowing the computer object from running and reclassifying the object as malware. Thus, the systems and methods described and claimed further the operation and security of the network by stopping an object from running and changing the classification of an object in real-time when unacceptable behavior is identified. (*See* Exhibit 1, ’250 Patent, 3:47-50; 4:19-30.)

55. Furthermore, the methods and systems claimed in the ’250 Patent, including generating a “mask” of acceptable behavior, allowing an object to run, continuing to monitor the object, and disallowing/reclassifying the object if the behavior extends beyond that permitted by the mask, are not routine or conventional. For example, while a “safe,” mask-permitted version of notepad.exe “would not be expected to perform a wide variety of events, such as transmitting data

to another computer or running other programs or running other programs" a "modified" and potentially "malevolent" version of notepad.exe could perform those unexpected events. (*See* Exhibit 1, '250 Patent, 11:27-41.) Unlike traditional malware systems that would have already made a binary determination that the notepad.exe object is safe, the methods and systems of the '250 Patent re-classify that version of notepad.exe as malware when its behavior becomes unexpected and "extends beyond that permitted by the mask." (*Id.* at 4:19-30.)

56. The applicants provided another example illustrating the unconventional nature and technical advantages and improvements, offered by the claimed systems and methods during prosecution:

> As an example, suppose a new version of Internet Explorer appeared. This could be a legitimate update to Internet Explorer released by Microsoft or alternatively it could be a file infected with a virus. In the prior art, the new object would have an unknown signature, so an in-house analyst would laboriously analyse the new object and determine whether or not it was safe. Whilst this analysis is carried out, the object would either be blocked, which would cause huge inconvenience to users of the new object, or allowed to run, in which case there is a risk of the object performing malevolent acts. In contrast, the present invention would collect data at the base computer from remote computers running the new version of Internet Explorer. Using the information collected, the system could determine that the new object purports to be a new version of Internet Explorer. However, it may not be apparent at this point whether or not the new object is capable of malevolent behaviour. In this scenario the present invention generates an appropriate behavioural mask for the object, e.g. by using a profile of behaviour of previous versions of Internet Explorer that are known not to be malware, or by using a profile for the behaviour appropriate for a web browser. The remote computers are allowed to let the new version run whilst monitoring its behaviour against the mask. The instant the new object exhibits some new, malevolent behaviour, this can be stopped at the remote computer, as well as being flagged to the base computer and used at the base computer to change the classification of the object. Thus, the present invention allows an instant response to an object changing its behaviour to exhibit malevolent behaviour in the future. (*See* '250 Patent Prosecution History, 2010-09-07 Amendment at 18, 19.)

57. Similarly, the '389 Patent describes and claims deploying an unconventional "event" based model that classifies a particular object as malicious or safe by analyzing real-time

data sent by remote computers on the events, or actions, that a particular software "object," and other objects deemed similar to it, initiate or perform on those computers. (*See* Exhibit 2, '389 Patent, 3:14-55.) This information is collected from across the network, correlated and used for subsequent comparisons to new or unknown computer objects to identify relationships between the correlated data and the new or unknown computer objects. The objects may be classified as malware based on this comparison.

58. Through continuous aggregate analysis of events involving computer objects as they occur across network endpoints, the methods and systems described and claimed in the '389 Patent maintain up-to-date information about computer objects (including malicious objects) seen across the network, identify relationships between those previously identified objects and any new or unknown objects, and make malware determinations based on those relationships. "For example, a new object that purports to be a version of notepad.exe can have its behavior compared with the behav[io]r of one or more other objects that are also known as notepad.exe…In this way, new patterns of behav[io]r can be identified for the new object." (*Id.* at 10:58-65.)

59. The methods and systems described and claimed in the '389 Patent can rapidly determine "the nature of the object," (*e.g.*, whether it is malicious or not) based on information such as the behavior of the object or effects the object has, without requiring "detailed analysis of the object itself as such" (manually reviewing the object's code) or reliance on matching an extensive database of known malicious "signatures." (*Id.* at 3:14-24; Exhibit 1, '250 Patent, 3:7-18.)

60. The Advanced Malware Detection Patents provide systems and methods that necessarily address issues unique to computer networks and computer network operation; namely the identification of "bad" software (*e.g.*, malware, viruses, etc.). These patents all provide unique

network security enhancement that solves the technical problem of rapidly identifying newly arising and emerging malware by reviewing information about the object and processes (*e.g.*, the behaviors and events associated with software objects and processes running on computers within the network).

61. The systems and methods claimed in the Advanced Malware Detection Patents improve the operation of computer networks by identifying malicious objects in real-time and taking action to remove or eliminate the threat posed by the malware object or process once it has been identified. The claimed inventions in these patents provide a technological solution to a technological problem--the inability of conventional code or signature matching solutions to identify new or unknown malware objects or processes at or near the runtime of the objects or processes themselves without the extensive delay and resource use associated with traditional systems.

Forensic Visibility Patents
U.S. Patent No. 9,578,045 and U.S. Patent No. 10,257,224

62. The ’045 and ’224 Patents are part of the same patent family and are each generally directed to providing forensic visibility into computing devices in a communication network by analyzing network events and creating audit trails. Plaintiff Webroot owns by assignment the entire right, title, and interest in and to the ’045 and ‘224 Patents. Webroot has granted OpenText an exclusive license to the ’045 and ’224 Patents.

63. The ’045 Patent is entitled “Method and Apparatus for Providing Forensic Visibility into Systems and Networks,” was filed on May 5, 2014, and was duly and legally issued by the USPTO on February 21, 2017. The ’045 Patent claims priority to provisional application 61/819,470 filed on May 3, 2013. A true and correct copy of the ’045 Patent is attached as Exhibit 3.

64. The ’224 Patent is also entitled “Method and Apparatus for Providing Forensic Visibility into Systems and Networks,” was filed on February 20, 2017 and was duly and legally issued by the USPTO on April 9, 2019. The ’224 Patent claims priority to the ’045 Patent and also to provisional application 61/819,470 filed on May 3, 2013. A true and correct copy of the ’224 Patent is attached as Exhibit 4.

65. The ’045 and ’224 Patents describe and claim inventive and patentable subject matter that significantly improves on traditional network forensic tools used to discover or identify security issues on computer networks. Network forensics generally relates to intercepting and analyzing network events to discover the source of security attacks. (*See* Exhibit 3, ’045 Patent, 1:22-24; Exhibit 4, ’224 Patent, 1:24-26.)

66. The ’045 and ’224 Patents improved on prior art network forensics tools by providing a technical solution to a technical problem experienced by computer networks and computer network operation. 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 3, ’045 Patent, 2:36-38; Exhibit 4, ’224 Patent, 2:38-40.)

67. 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 3, ’045 Patent, cl. 1; Exhibit 4, ’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.

The claimed systems and methods of the Forensic Visibility Patents address an identified weakness in conventional systems and processes; namely the ability to monitor, capture and/or analyze what is occurring *at* computing devices on a computer network, thereby providing an improved way to address the technical problem of discovering security attacks or security problems within a computer network.

68. 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 event and computer objects of interest. (*See* Exhibit 3, '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.

69. The 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.

70. The 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 3, '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* Exhibit 3, '045 Patent, 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.

71. Thus, the '224 and '045 Patents describe and claim systems and methods that provide technical advantages and improvements over traditional network security and forensic systems, including more efficient and accurate identification of malware (*e.g.*, the contextual and global perspective information reduced false negative and positives for malware detection). The described systems and methods also improved the identification of other malware (and corresponding events) that might otherwise go undetected in prior systems, thereby improving system performance and reducing the number of resources required.

72. Indeed, the described systems and methods enable end-to-end forensic visibility into event occurrences across a networked environment and from the bottom of the stack to the top, thereby improving upon conventional network forensic products. (*See* Exhibit 3, '045 Patent,

2:31-38, 3:49-55; Exhibit 4, '224 Patent, 2:33-40, 3:52-59; *see also* Exhibit 3, '045 Patent, 4:36-41; Exhibit 4, '224 Patent, 4:39-44.)

73. Applicant further explained during prosecution how the generation of contextual state information and obtaining a global perspective—including for objects and events other than those that were detected, such as the originating object—are unconventional steps in the areas of malware detection and network forensics. For example, Applicant explained how the described systems and methods improve the system performance of computing devices:

> In this case, the claimed invention provides for determining correlations between events and objects and creating an audit trail for each individual event. For example, a context analyzer may correlate an actor, victim, and/or event type to one or more originating processes, devices, and users. After the analysis is complete, a sensor agent may use the correlated data to generate a global perspective for each event such that an administrator is able to forensically track back any event which occurs to what triggered it. Thus, the global perspective represents a drastic transformation of raw event data into a comprehensive, system-wide forensic audit trail. ('045 Patent Prosecution History, 2016-03-16 Amendment at 11-12.)

> In this case, examples of the claimed systems and methods provide low level system filters which intercept system events "in a manner such that the operation of the system filter does not impact system performance." Specification, para. [0008]. For example, on an average system, because tens of millions of events take place every minute, the noise ratio can prevent forensic solutions from being able to provide sufficient value to the end consumer of their data due to the inability to quickly find important events. A product which impacts system performance will have considerably diminished value to an administrator and can negatively affect the results of an analysis undertaken. Examples of the present systems and methods address this shortcoming by providing a system filter that substantially improves the system performance of the computing devices in the system. (*See* '045 Patent Prosecution History, 2016-03-16 Amendment at 12.)

74. During prosecution, Applicant further explained how the claims are directed to solving a technical problem and a specific improvement in computer functionality relating to computer security:

> ***[T]he claims are directed to solving a technical problem***. Typically, network forensic systems use network forensic tools (e.g., network sniffers and packet capture tools) to detect and capture information associated with communication

> sessions. Although such network forensic tools are operable to passively collect network traffic, the tools reside at a network edge (e.g., outside of a system or hosts). As a result, the network forensic tools have no ability to obtain useful information within a host or to establish any sort of context from within a host that is generating and/or receiving network events. To address this, aspects of the present disclosure enable methods for providing forensic visibility into systems and networks. For example, a local aggregator/interpreter, context analyzer and sensor agent may provide visibility into occurrences across an environment to ensure that a user (e.g., an administrator) is aware of any system change and data communications in and out of the computing devices residing on the network. During this process, identified events may be correlated to objects, thus creating an audit trial [sic] for each individual event. (*See* '045 Patent Prosecution History, 2016-03-16 Amendment at 9-10. (emphasis added))
>
> Here, ***the claims are directed to a specific improvement in computer functionality relating to computer security, and more specifically to providing end-to-end visibility of events within a system and/or network***. (*See* '224 Patent Prosecution History, 2018-08-29 Amendment at 10-11 (citing '224 Patent specification) (emphasis added).)
>
> The Specification subsequently discusses a variety of ways in which the claimed subject matter solves the above-described problem. For example: "It is, therefore, one aspect of the present disclosure to provide a system and method whereby events occurring within a computing device are captured and additional context and a global perspective is provided for each capture
> event. For example, a sensor agent may provide visibility into occurrences across an environment, such as a networked environment, to ensure that an administrator is aware of any system changes and data communication in and out of computing devices residing on the network." (*See* '224 Patent Prosecution History, 2018-08-29 Amendment at 11-12 (citing '224 Patent specification).)

75. In response to these arguments, the Examiner withdrew a rejection based on 35 U.S.C. §101 and allowed the claims of the Forensic Visibility Patents to issue. As recognized by the USPTO Examiner, the claimed inventions of the '045 and '224 Patents provide a technical solution to the technical problem of forensic visibility regarding events in a computer network.

US. Patent No. 10,284,591

76. U.S. Patent No. 10,284,591 is entitled "Detecting and Preventing Execution of Software Exploits," was filed on January 27, 2015 and was duly and legally issued by the USPTO on May 7, 2019. The '591 patent claims priority to provisional application 61/931,772 filed January 27, 2014. A

true and correct copy of the ’591 Patent is attached as Exhibit 5. Plaintiff Webroot owns by assignment the entire right, title, and interest in and to the ’591 Patent. Webroot has granted Plaintiff OpenText an exclusive license to the ’591 Patent.

77. The ’591 Patent describes and claims an “anti-exploit” technique to prevent undesirable software and/or other computer exploits from executing. (*See* Exhibit 5, ’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 monitors 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.)

78. The ’591 Patent describes and claims unconventional “stack walk processing” techniques for detecting and preventing unwanted software exploits during which memory checks are performed before an address of an originating caller function is reached. 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.” (*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.)

79. The “stack walking” and “memory checks” described and claimed in the ’591 Patent are fundamentally rooted in computer technology—in fact, they are processes only performed within a computer context. The techniques described and claimed in the ’591 Patent addresses a problem that specifically arises in the realm of computer technology (namely, computer exploit identification) by,

inter alia, performing memory checks and detection specified behavior during stack walking.

80. The ’591 Patent further describes and claims unconventional techniques that address identified weaknesses in conventional exploit prevention technologies. For example, unlike exploit prevention technologies that try to prevent an exploit from ever starting its own shellcode to execute a malicious payload, the ’591 Patent describes and claims techniques that prevent shellcode from executing a malicious payload even if the shellcode has been started. (*Id.* at 6:24-30; *see also* 7:56-62.) Thus, these unconventional techniques address an identified weakness in conventional exploit prevention systems and provide technical advantages including enhanced security protection, improved detection of potential security exploits, reduction in error rate identifying and marking suspicious behavior (*e.g.*, false positives), and improved usability and interaction for users who are not required to continuously monitor for security exploits. (*Id.* at 2:44-51.) As such, the ’591 Patent describes and claims specific computer-related technological steps to accomplish an improvement in computer security and functionality and is directed to a specific technological solution to a problem unique to computers.

U.S. Patent Nos. 10,599,844 and 11,409,869

81. The ’844 Patent is entitled “Automatic Threat Detection of Executable Files Based on Static Data Analysis,” was filed May 12, 2015 and was duly and legally issued by the USPTO on March 24, 2020. A true and correct copy of the ’844 Patent is attached as Exhibit 6. Plaintiff Webroot owns by assignment the entire right, title, and interest in and to the ’844 Patent. Webroot has granted Plaintiff OpenText an exclusive license to the ’844 Patent.

82. The ’844 Patent is entitled “Automatic Threat Detection of Executable Files Based on Static Data Analysis,” was filed May 12, 2015 and was duly and legally issued by the USPTO on March 24, 2020. A true and correct copy of the ’844 Patent is attached as Exhibit 6. Plaintiff

Webroot owns by assignment the entire right, title, and interest in and to the ’844 Patent. Webroot has granted Plaintiff OpenText an exclusive license to the ’844 Patent.

83. The ’869 Patent, entitled “Automatic Threat Detection of Executable Files Based on Static Data Analysis,” is a continuation of the ’844 Patent and claims priority to the application of the ’844 Patent. The ’869 Patent was filed on February 14, 2020 and was duly and legally issued by the USPTO on August 9, 2022. A true and correct copy of the ’869 Patent is attached as Exhibit 7. Plaintiff Webroot owns by assignment the entire right, title, and interest in and to the ’869 Patent. Webroot has granted Plaintiff OpenText an exclusive license to the ’869 Patent.

84. The ’844 and ’869 Patents address and improve upon conventional approaches to malware detection in computer networks and computer network operation. Every day, an uncountable number of new executable files are created and distributed across computer networks. Some of those files are unknown, and malicious. It is, thus, vital to accurately and immediately diagnose those files for any potential threat, while also efficiently using resources (e.g., processing power). (See Exhibit 6, ’844 Patent, 1:7-13.)

85. 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 6, ’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.)

86. Another approach for protecting against potential threats from unknown executable files involved wavelet decomposition to determine software entropy. (See WBRT007115 at 7491, ’844 Patent Prosecution History, April 24, 2019 Applicant Remarks, at 8.) Wavelet decomposition is a process where an original image is decomposed into a sequence of new images, usually called wavelet planes. (Id.) In this method, each data file in a set of data files is split into random, non-overlapping file chunks of a fixed length. (Id.) Those file chunks are then represented as an entropy time-series, which measures the time it takes for each chunk to decompose. (Id.) Said differently, this approach measured how much time it took a data file to decompose. (Id.) Once the file decomposition rate, or entropy time-series, had been calculated, that rate would be compared to decomposition rates of “known bad” files to identify files that contain malware. (Id. at WBRT007492.) This process required significant computing resources—typically taking hours to complete—and was not sufficiently accurate in identifying malware.

87. The ’844 and ’869 Patents significantly improve upon and address shortcomings associated with these prior approaches. The Patents describe and claim methods and systems that detect threats in executable files without the need to decrypt or unpack those 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 6, ’844 Patent, 1:20-21; cl. 1.) The described systems 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. By using this unconventional approach to determine whether a file executable on a computer poses a threat, the

’844 and ’869 Patents improve on the operation of the computer network associated with the computer by enhancing security, including by increasing detection of new threats, reducing the error rates in identifying suspicious files, and improving efficiency in detecting malicious files. (See Exhibit 6, ’844 Patent, 2:46-56.)

88. The ’844 and ’869 Patents describe and claim techniques that employ a learning classifier (e.g., a machine-learning classifier) to determine whether an executable file is malicious, for example by using the classifier to classify data into subgroups and identify and analyze specific data points to which those subgroups correspond. (See Exhibit 6, ’844 Patent, 4:33-41, 7:40-8:1.) The described and claimed techniques also selectively turn on or off features for evaluation by the learning classifier. (See id. at 7:57-66.) Doing so accelerates analysis and reduces false positives by testing those features of a file likely to be relevant to a determination of its maliciousness. For example, the learning classifier “may detect that the file does not contain ‘legal information’,” such as “timestamp data, licensing information, copyright information, etc.” (See id. at 7:66-8:5.) In this example, given the lack of legal protection information in the file, the learning classifier would “adaptively check” the file for additional features that might be indicative of a threat,” while “turn[ing] off,” and thus not use processing time unnecessarily checking features related to an evaluation of “legal information.” (Id. at 8:5-10.)

89. Second, the ’844 and ‘869 Patents describe and claim techniques that use character strings extracted from within the executable file to generate a feature vector and then evaluate that feature vector using support vector processing to classify executable files. (See Exhibit 6, ’844 Patent, 9:2-11.) The classifier provides, for example, the ability to leverage the indicia of “benign” files, which use “meaningful words” in certain data fields, versus “malicious” files, which leave such fields empty or full of “random characters,” to build meaningful feature vectors that are

analyzed to make faster and more identifications of malware (See, e.g., Exhibit 6, '844 Patent, 9:2-18.)

90. The '844 and '869 Patents are thus directed to specific solutions to problems necessarily rooted in computer technology, namely, the determination whether a file executable on a computer poses a threat. The '844 and '869 Patents improved upon the accuracy and efficiency of malware detection. (See Exhibit 6, '844 Patent, 2:15-45.)

91. By using some or all of the unconventional techniques described above to determine whether a file executable on a computer poses a threat, the '844 and '869 Patents address a problem necessarily involving computers and improve upon the operation of computer networks. In particular, the '844 and '869 Patents achieve a number of technical advantages over conventional approaches to malware detection including, for example:

- enhanced security protection including automatic detection of threats, reduction or minimization of error rates in identification and marking of suspicious behavior or files (*e.g.*, cut down on the number of false positives),
- ability to adapt over time to continuously and quickly detect new threats or potentially unwanted files/applications,
- improved efficiency in detection of malicious files, and
- improved usability and interaction for users by eliminating the need to continuously check for security threats.

(*See* Exhibit 6, '844 Patent, 2:15-57.)

U.S. Patent No. 9,413,721

92. The '721 Patent is entitled "Methods and Apparatus for Dealing with Malware," was filed on February 13, 2012, and duly and legally issued by the USPTO on February 5, 2013. A true and correct copy of the '721 Patent is attached as Exhibit 7. Plaintiff Webroot owns by assignment the entire right, title, and interest in and to the '721 Patent. Webroot has granted Plaintiff OpenText an exclusive license to the '721 Patent.

93. The systems and methods described and claimed in the '721 Patent are directed to improved techniques for detecting and classifying malware, a technological problem fundamentally and inextricably associated with computer technology and computer networks. The '721 Patent explains that prior anti-malware products used signature matching to detect malware, either locally or at a central server. (Exhibit 7, '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.)

94. To address these shortcomings, the '721 Patent describes and claims unconventional, novel distributed system architectures, such as remote computers that may be allocated to "threat" servers, with "central" servers sitting behind them. (Exhibit 7, '721 Patent, 9:16-57.) These enhanced computer architectures provide a technical solution to the technical

problem of detecting and classifying malware in a computer network environment, thus improving network security while identifying and classifying malware threats in real-time without delays engendered by use of human analysts. (*See, e.g.*, Exhibit 7, '721 Patent, 1:60-2:7)

95. In particular, the '721 Patent described and claims embodiments that 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 7, '721 Patent 12:28-54.) This novel network architecture improves the operation of the computer network over traditional networks because, for example and as described in the '721 Patent, "[t]his 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.)

96. Further, "by being able to query and analyze the collective view of an object, i.e., its metadata and behaviours, across all agents [] that have seen it, a more informed view can be derived, whether by human or computer, of the object. In addition, it is possible to cross-group objects based on any of their criteria, i.e. metadata and behaviour." (*Id.* at 18:17-22.) Thus, embodiments enable better malware identification than conventional systems (*e.g.*, using human analysis) in addition to providing an efficiency benefit. The patent explicitly notes that "the work in processing the raw data [] is too large of a task to be practical for a human operator to complete." (*Id.* at 18:50-52.)

97. The systems and methods described and claimed in the '721 patent provide further technical improvements. For example, the information collected at the central server includes

addition information about the object being classified as well as a count associated with the number of times that the first computer object has been seen to the central server. (*Id.* at cl. 1) As explained above, using information about the object (such as behavior information) being classified, the systems and methods described and claimed in the '721 Patent provide an approach that is more effective than traditional code or signature matching techniques for classifying objects as malicious. (*Id.* at 1:54-2:14.)

98. Prior methods of classifying malware had technical drawbacks when used on a distributed network. For example, a distributed network that required each server to maintain rules for determining what is malware required each server to deal with huge amounts of largely common data. (Exhibit 7, '721 Patent, 12:20-24.) It was also generally impractical to store the required data on each server because, for example, there were problems determining whether or not the data—which is both massive and constantly changing—is common and up-to-date in real-time. (*Id.* at 12:24-27.) The 3-tiered architectures described and claimed in embodiments of the '721 Patent provides a technical solution for distributed computer networks by, *inter alia*, reducing the workload across the network. (*Id.* at 12:28-59.)

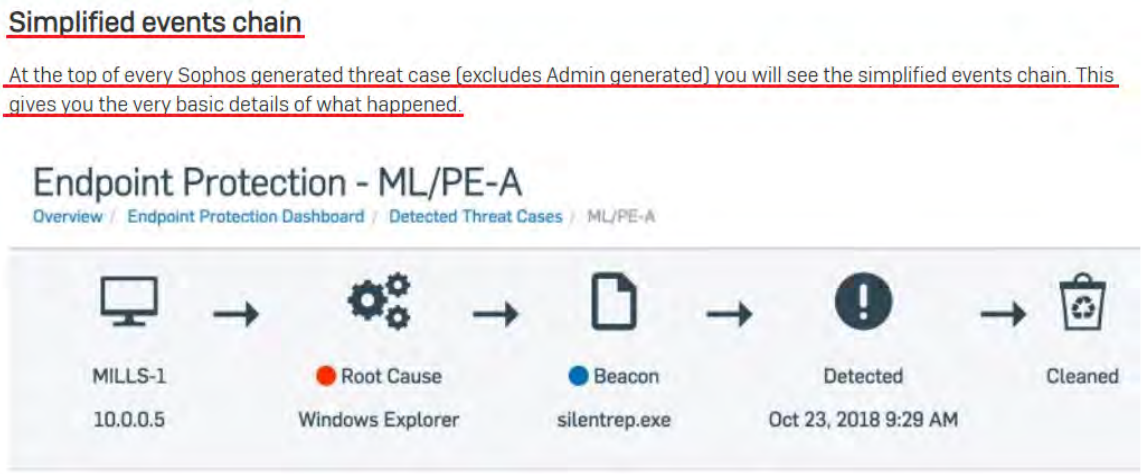
99. Accordingly, the '721 Patent discloses and claims, among other things, an unconventional technological solution to the inherently computer-network centric technical issue of identifying malware in computer systems. The solution implemented by the '721 Patent provides a specific and substantial improvement over prior malware classification systems, for example by introducing novel computer network architecture elements combined in an unconventional manner. These approaches improve the function and working of malware detection services by, for example, utilizing multiple threat servers and central servers and performing the analysis and communication carried out by each type of server in an unconventional and efficient

manner. (*See, e.g.*, Exhibit 7, '721 Patent, cl. 1.) These elements and their combination represent a marked improvement in the functioning of computer systems utilized to identify and detect malware in computers networks.

ACCUSED PRODUCTS

100. Defendant offers and sells the Accused Products including Sophos' Intercept X Advanced with EDR and XDR, Sophos Web Appliance, Sophos XG Firewall, and Sophos' Synchronized Security, as well as products and services with similar functionality. These products provide and implement malware detection, network security, and endpoint protection platforms for individuals and enterprises and incorporate Plaintiffs' patented technologies.

101. Sophos' Intercept X Advanced with XDR ("Intercept X"), formerly known as Intercept X Advanced with EDR, is an endpoint protection platform that establishes forensic attack chains for infections and leverages machine learning techniques to detect malware.



(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html.)

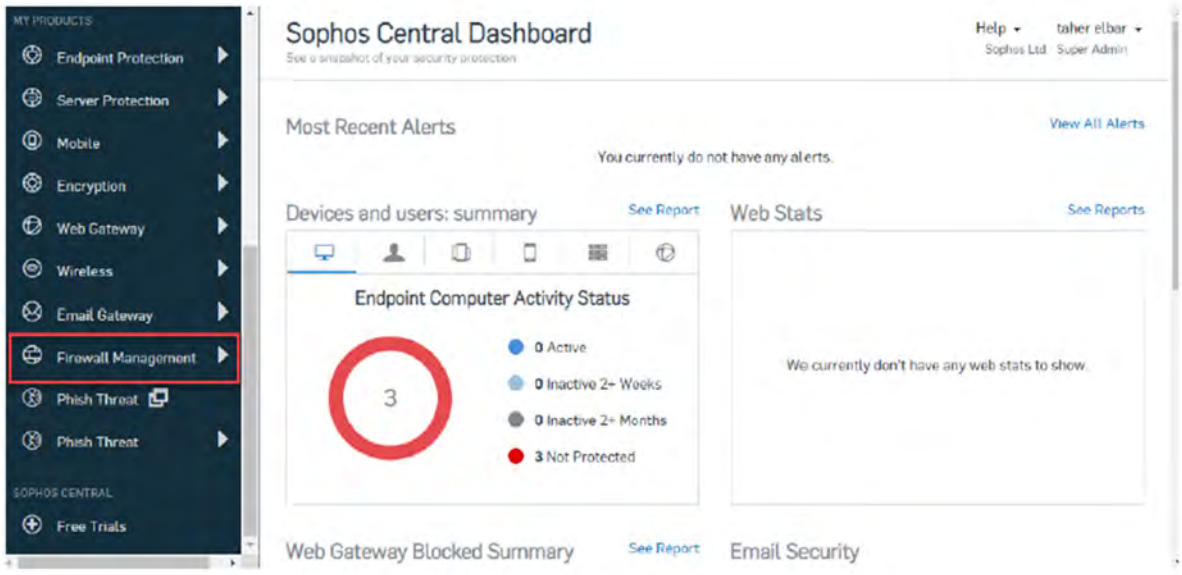
102. Sophos' Synchronized Security is an integrated cybersecurity platform that manages Sophos' endpoint security software products, such as XG Firewall and Intercept X, from

a central point via a cloud-based Control Center. It receives "Security Heartbeat[s]" that share "health, security and security information" from Sophos' endpoint security products installed on various endpoints across a network protected by Synchronized Security.

103. The Sophos Synchronized Security Control Center's dashboard displays information about endpoints in the network. The details of this information include the name of the computer, IP address, operating system, the Sophos security products installed on the endpoint, and the health status of the endpoint.

With Synchronized Security, Sophos products share threat, health, and security information in real time via the Security Heartbeat™ and respond automatically to threats.

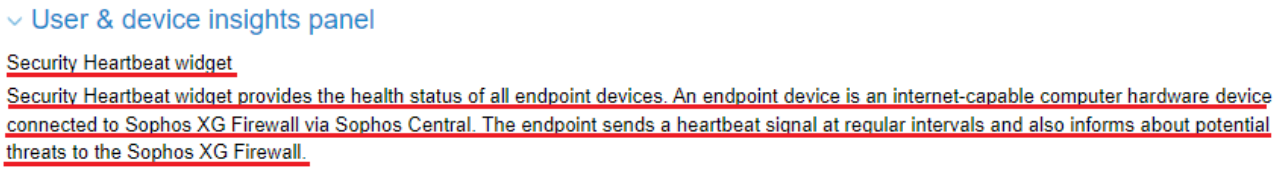
This quick start guide walks you through enabling Synchronized Security between Sophos XG Firewall and Sophos endpoint and server protection managed through Sophos Central (including Sophos Intercept X and Sophos Intercept X for Server).



Control center

The control center appears as soon as you sign in.

The control center provides a single screen snapshot of the state and health of the security system.



(*See* https://www.sophos.com/en-us/medialibrary/PDFs/factsheets/sophos-quick-start-guide-synchronized-security.pdf; *see also* https://docs.sophos.com/nsg/sophos-firewall/17.5/Help/en-us/webhelp/onlinehelp/nsg/sfos/concepts/SSLVPNLiveUsersManage.html.)

FIRST CAUSE OF ACTION
(INFRINGEMENT OF THE ’250 PATENT)

104. Plaintiffs reallege and incorporate by reference the allegations of the preceding paragraphs of this Complaint.

105. Defendant has infringed and continue to infringe one or more claims of the ’250 Patent in violation of 35 U.S.C. § 271 in this District and elsewhere in the United States and will continue to do so unless enjoined by this Court. the Accused Products, including features such as Intercept X Advanced with XDR (“Intercept X”), at least when used for their ordinary and customary purposes, practice each element of at least claim 1 of the ’250 Patent as described below.

106. For example, claim 1 of the ’250 Patent recites:

> 1. A method of classifying a computer object as malware, the method comprising:
>
> at a base computer, receiving data about a computer object from each of plural remote computers on which the object or similar objects are stored, the data including information about the behaviour of the object running on one or more remote computers;
>
> determining in the base computer whether the data about the computer object received from the plural computers indicates that the computer object is malware;
>
> classifying the computer object as malware when the data indicates that the computer object is malware; when the determining does not indicate that the computer object is malware, initially classifying the computer object as not malware;

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;

running said object on at least one of the remote computers;

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

allowing the computer object to continue to run when behaviour of the computer object is permitted by the mask:

disallowing the computer object to run when the actual monitored behaviour of the computer object extends beyond that permitted by the mask; and,

reclassifying the computer object as malware when the actual monitored behaviour extends beyond that permitted by the mask.

107. The Accused Products perform each element of the method of claim 1 of the '250 Patent. To the extent the preamble is construed to be limiting, the Accused Products perform *a method for classifying a computer object as malware*, as further explained below. For example, Intercept X with XDR ("Intercept X") "scans across [the] entire environment and highlight[s] suspicious activity, anomalous behavior and other IT issues." Intercept X displays "threats," including processes classified as malware, in its "Sophos Central" and "Threat Analysis Center" dashboards.

XDR builds upon that solid foundation by adding even more data and context that both increases visibility and gives the user even more insight during an investigation. This results in faster and more accurate incident detection and response. Additional data sources can include firewall, email, cloud and mobile information. For example, adding in firewall data makes it simple to correlate a malicious traffic detection by the firewall with a compromised endpoint, or to see which application is causing the office network connection to run slowly.

One of the most valuable ways to use XDR is to begin with the 'macro' spotlight that gives you the tools to quickly scan across your entire environment and highlight suspicious activity, anomalous behavior and other IT issues. When an issue is identified you can then hone-in on a device of interest, pulling live data or remotely accessing the device in order to dig deeper and take remedial action.

(*See* https://secure2.sophos.com/en-us/medialibrary/Gated-Assets/white-papers/sophos-xdr-beginner-guide.pdf.)

108. The Accused Products perform a method that includes *receiving data at a base computer about a computer object from each of plural remote computers on which the object or similar objects are stored, the data including information about the behaviour of the object running on one or more remote computers*. For example, each endpoint on which Intercept X is installed sends data about the processes executing on it to the cloud-based "Sophos Central" computer which stores that data in a database and manages endpoints, which include remote computers, within a network. This data includes information about the behavior of an object running on one or more remote computers. For example, data can be queried from each endpoint using "Live Discover" SQL queries through the "Threat Analysis Dashboard," to detect, for example, processes (running objects) that have made "[u]nusual changes to the registry" or to "search devices for signs of a suspected or known threat if Sophos Central has found the threat elsewhere." Data about each process is automatically analyzed, marked as a "threat case" if appropriate, and displayed as such in the "Threat Analysis Center." Moreover, such data is also periodically uploaded by each endpoint to a cloud-based computer "Data Lake," which can be queried, for example, to obtain data about which processes executed on a given endpoint.

Live Discover

Live Discover allows you to check the devices that Sophos Central is managing, look for signs of a threat, or assess compliance.

You can use Live Discover queries to search devices for signs of threats that haven't been detected by other Sophos features. For example:

- Unusual changes to the registry.
- Failed authentications.
- A process running that is very rarely run.

You can also search devices for signs of a suspected or known threat if Sophos Central has found the threat elsewhere, or if a user reports suspicious behavior on their device.

You can also check the compliance of each device. For example, you can search for out-of-date software or browsers with insecure settings.

This page tells you how to use Live Discover. You can also familiarize yourself with it by completing the Sophos XDR Training.

How queries work

We provide a range of queries for you to use to check your devices. You can use them as they are, or edit them (you'll need to be familiar with osquery or SQL). You can also create queries.

You can run queries to get information from different sources:

- Endpoint queries get the latest information from devices that are currently connected.

(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/learningContents/LiveDiscover.html; *see also* https://community.sophos.com/intercept-x-endpoint/b/blog/posts/introducing-the-new-threat-analysis-center.)

Data Lake queries

Data Lake queries let you search security and compliance data that your devices upload to the cloud.

You can run Data Lake queries with Live Discover, a feature in our Threat Analysis Center.

Live Discover now lets you choose which data source you use when you set up and run a query:

- Endpoints that are currently connected.
- The Data Lake in the cloud.

For help with Live Discover see Live Discover.

How the Data Lake works

We host the Data Lake and provide scheduled "hydration queries" that define which data your endpoints upload to it.

However, before you use Data Lake queries, you must make sure that data is being uploaded. To turn on uploads of data, see Data Lake uploads.

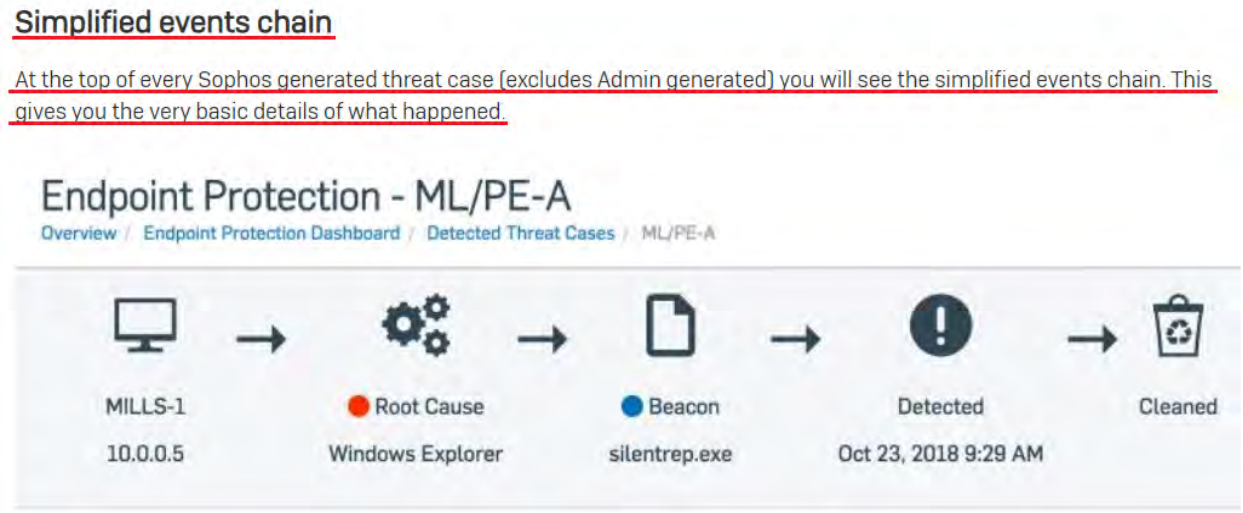
We store the data for 30 days.

We provide pre-prepared Data Lake queries you can run. You can use them as they are or edit them. You can also create your own queries.

(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/DataLakeQueries.html.)

109. The Accused Products perform a method that includes *determining in the base computer whether the data about the computer object received from the plural computers indicates that the computer object is malware*. For example, each endpoint on which Intercept X is installed sends data about the processes executing on it to the cloud-based "Sophos Central" computer,

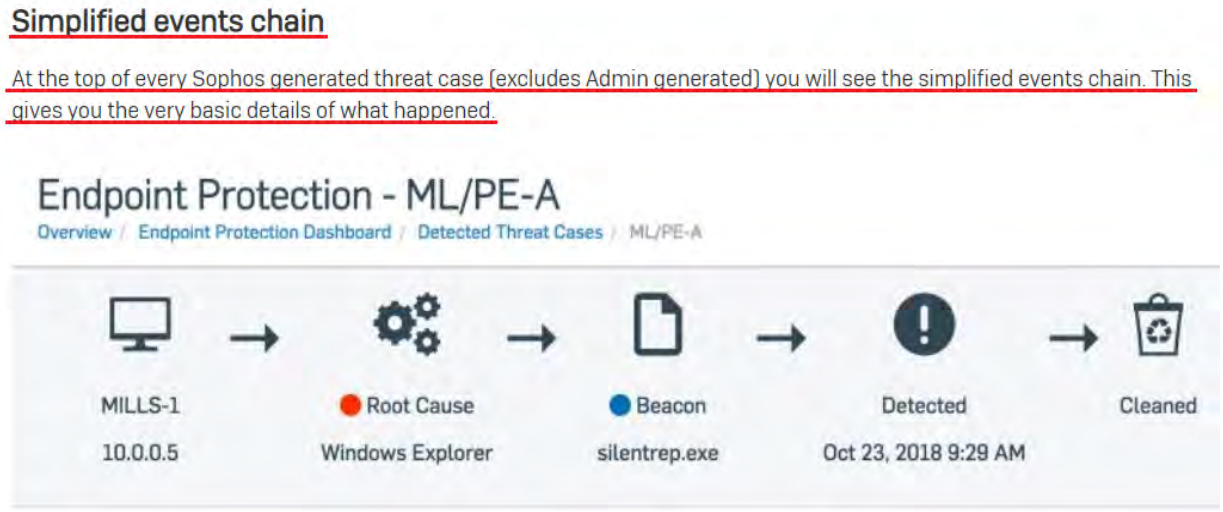
which stores that data in a database and manages endpoints (including remote computers), within a network. This data includes information about the behavior of an object running on one or more remote computers. For example, data can be queried from each endpoint using "Live Discover" SQL queries through the "Threat Analysis Dashboard," to detect, for example, processes (running objects) that have made "[u]nusual changes to the registry" or to "search devices for signs of a suspected or known threat if Sophos Central has found the threat elsewhere." Data about each process is automatically analyzed, marked as a "threat case" if appropriate, and displayed as such in the "Threat Analysis Center." Moreover, such data is also periodically uploaded by each endpoint to a cloud-based computer "Data Lake," which can be queried, for example, to obtain data about which processes executed on a given endpoint.



(*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_USl.)

110. The Accused Products perform a method that includes *classifying the computer object as malware when the data indicates that the computer object is malware; when the determining does not indicate that the computer object is malware, initially classifying the computer object as not malware*. For example, as explained above, the Accused Products use the data on processes that each endpoint sends to Sophos Central to determine whether those processes

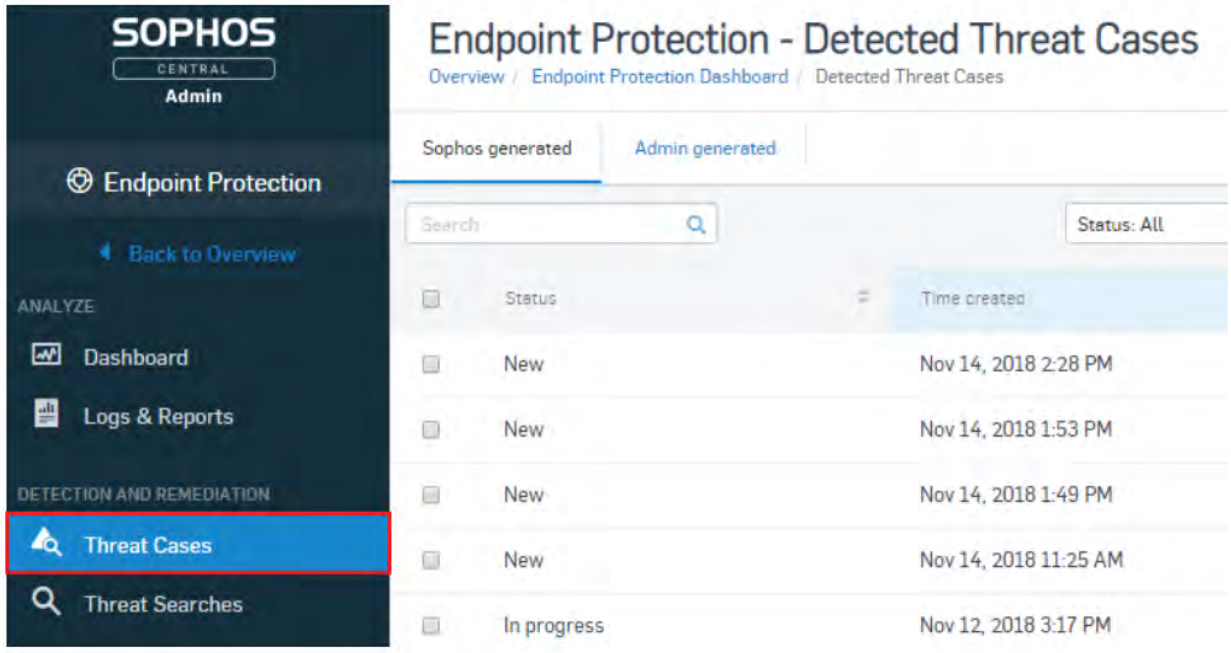
constitute malware. In the example below, Sophos Central has identified that the process "silentrep.exe" is a "threat case" and a variant of the malware class "ML/PE-A." When the data that Sophos Central receives does not indicate that the process is malware or potentially malicious, that process is not initially marked as a "threat case."



(*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_USl; *see also* https://community.sophos.com/intercept-x-endpoint/b/blog/posts/introducing-the-new-threat-analysis-center.)

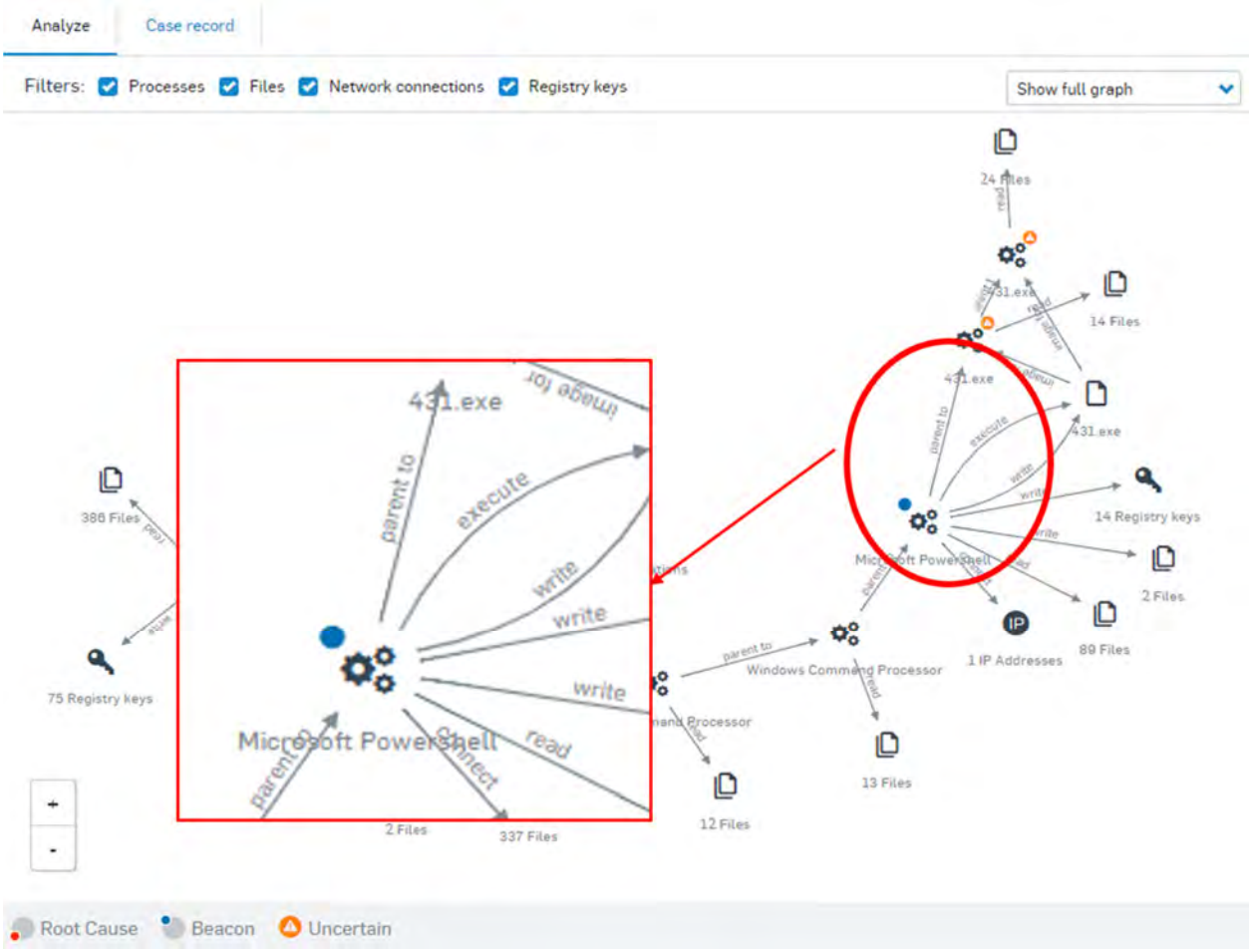
For customers with Sophos EDR, the full list of Threat Cases can be found in the below locations:

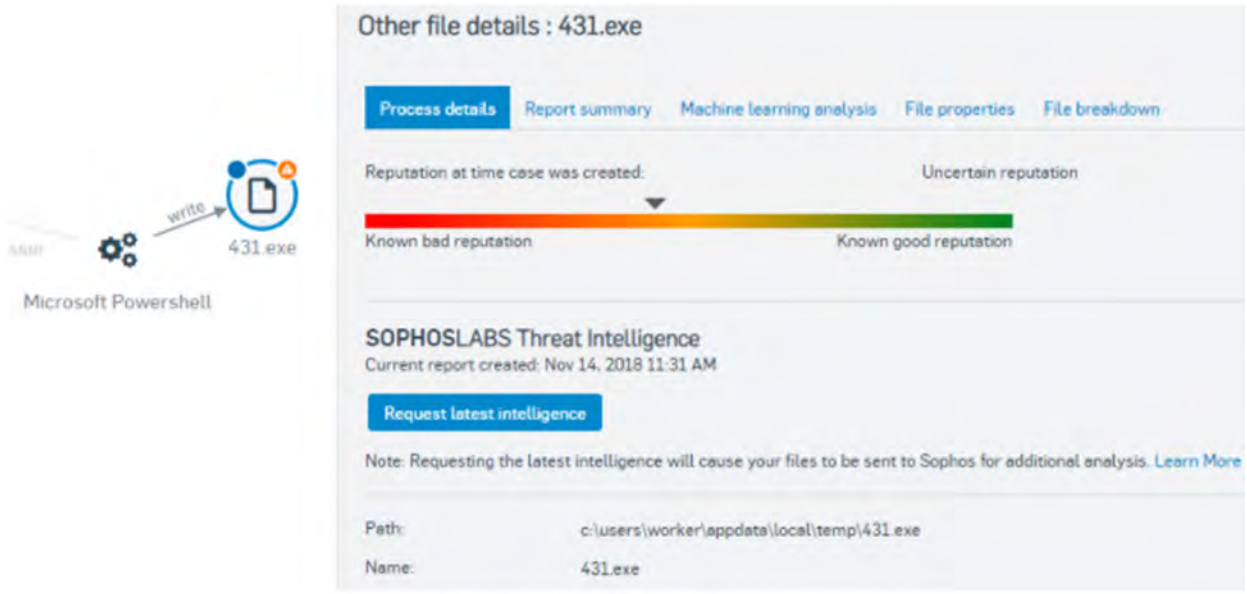
- Endpoint Protection > Detection and Remediation > Threat Cases
- Server Protection > Analyze > Threat Cases



To view a Threat Case click on the detection **Name**:

111. The Accused Products perform a method that includes *automatically generating a mask for the computer object that defines acceptable behavior for the computer object, wherein the mask is generated in accordance with normal behavior of the object determined from said received data*. For example, the Accused Products assess "threat cases" by detecting illicit behaviors (*e.g.*, behaviors that are not normal for the object) such as the modification of "registry keys." In particular, in the example shown below, the "Analyze" tab of a "threat case" displayed in the "Threat Analytics Center" illustrates the illicit execution, on an infected endpoint device, of the suspicious process "431.exe" by Microsoft Powershell, which is marked as the "Beacon," and the subsequent modification of "registry keys" by the file "431.exe" on the infected endpoint. The process "431.exe" is quarantined after its behavior is detected as being malicious.


Analyze
Case record
Filters:
Processes
Files
Network connections
Registry keys
Show full graph
24 Files
431.exe
14 Files
431.exe
431.exe
14 Registry keys
2 Files
89 Files
1 IP Addresses
386 Files
75 Registry keys
431.exe
parent to
execute
write
write
write
read
Microsoft Powershell
2 Files
337 Files
Windows Command Processor
13 Files
12 Files
Root Cause
Beacon
Uncertain



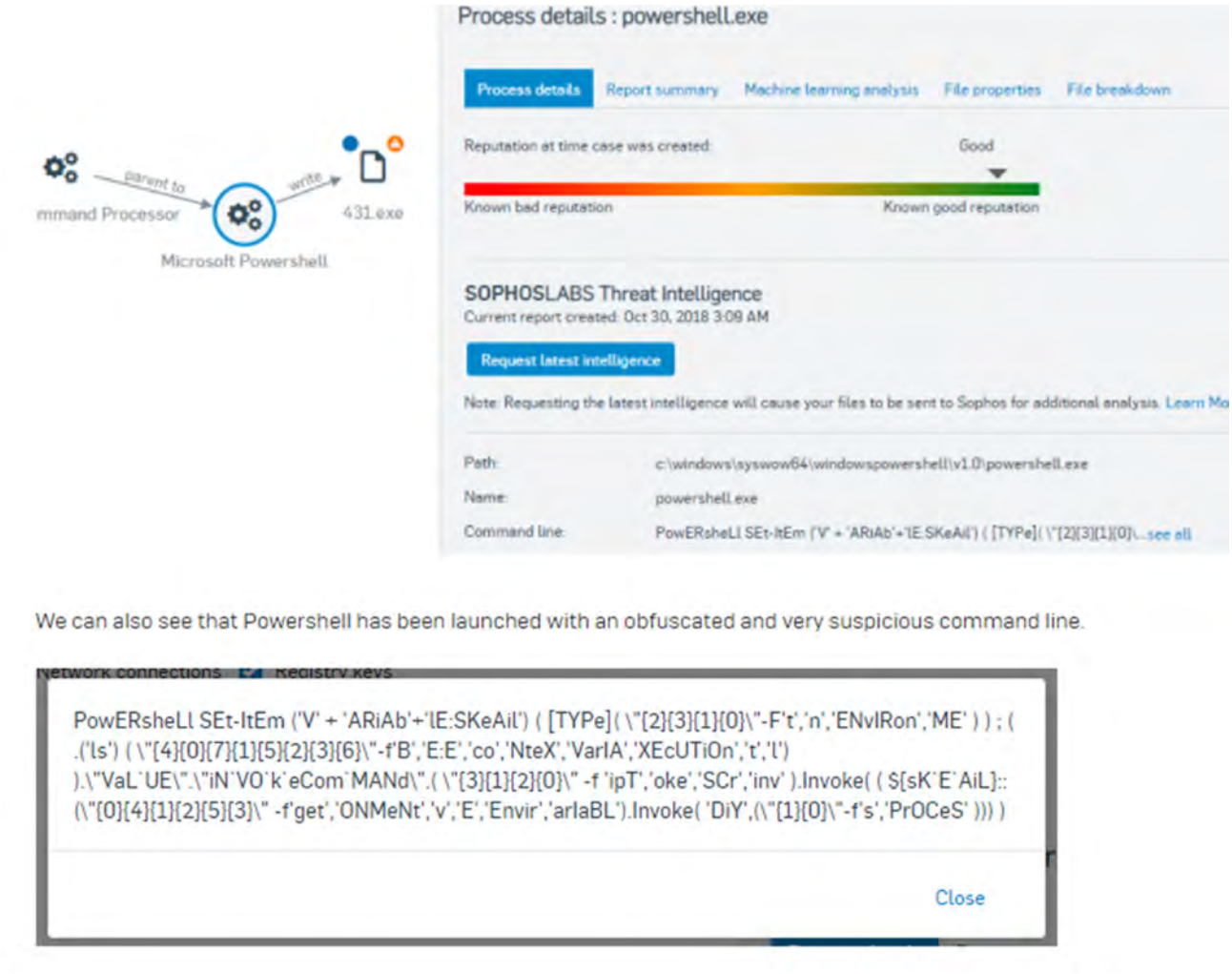
Important: Customers who are **not** using Sophos XDR, you will only see the one **Process details** tab.

For customers using Sophos EDR, by pressing the **Request latest intelligence** button, the file will be retrieved out of the Sophos quarantine and submitted to SophosLabs. A couple of minutes later the four other tabs (Report summary, Machine learning analysis, File properties, File breakdown) pictured will be displayed. The purpose of these these additional tabs is to help display the various properties of the file in a simple way. This can be useful for various reasons, one of them is to feel confident that the file is indeed malicious and not something you want in your environment. For more information on SophosLabs Threat Intelligence, please see: Sophos Central: Threat intelligence overview.

(*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_US.)

112. Based on the data that it received about the behavior of processes (*e.g.*, "431.exe" and the behavior of the process "Microsoft PowerShell," which executed "431.exe"), Sophos Central determined that PowerShell exceeded the scope of normal behavior by being launched "with an obfuscated and very suspicious command line" and that "431.exe" illicitly modified registry keys and was written to "the users AppData location," which is "typically meant for data not executable," resulting in its quarantine. As such, Sophos Central defines what is typical or normal behavior for both malware and non-malware. For example, it explains that "[o]bfuscation is very typical in malicious code and is designed to hide the true goal behind the code." In another example, when one command shell launches anther command shell, that behavior is deemed suspicious, meaning that when one command shell does not invoke another, it is deemed non-

malware behavior: "We can also see that when CMD was launched it then launched another copy of CMD, this one with a similar suspicious command line."



(*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_US.)

113. The Accused Products perform a method that includes *running said object on at least one of the remote computers and automatically monitoring operation of the object on the at least one of the remote computers*. As explained above, the Accused Products identify "threat cases" as they occur on each endpoint. In particular, in the example shown above, the "Analyze" tab of a "threat case" displayed in the "Threat Analytics Center" illustrates the illicit execution, on an infected endpoint device, of the suspicious process "431.exe" by Microsoft Powershell, which

is marked as the "Beacon," and the subsequent modification of "registry keys" by the file "431.exe" on the infected endpoint. The process "431.exe" was initially allowed to execute until it suspiciously modified registry keys, at which point it was classified as malware and quarantined.

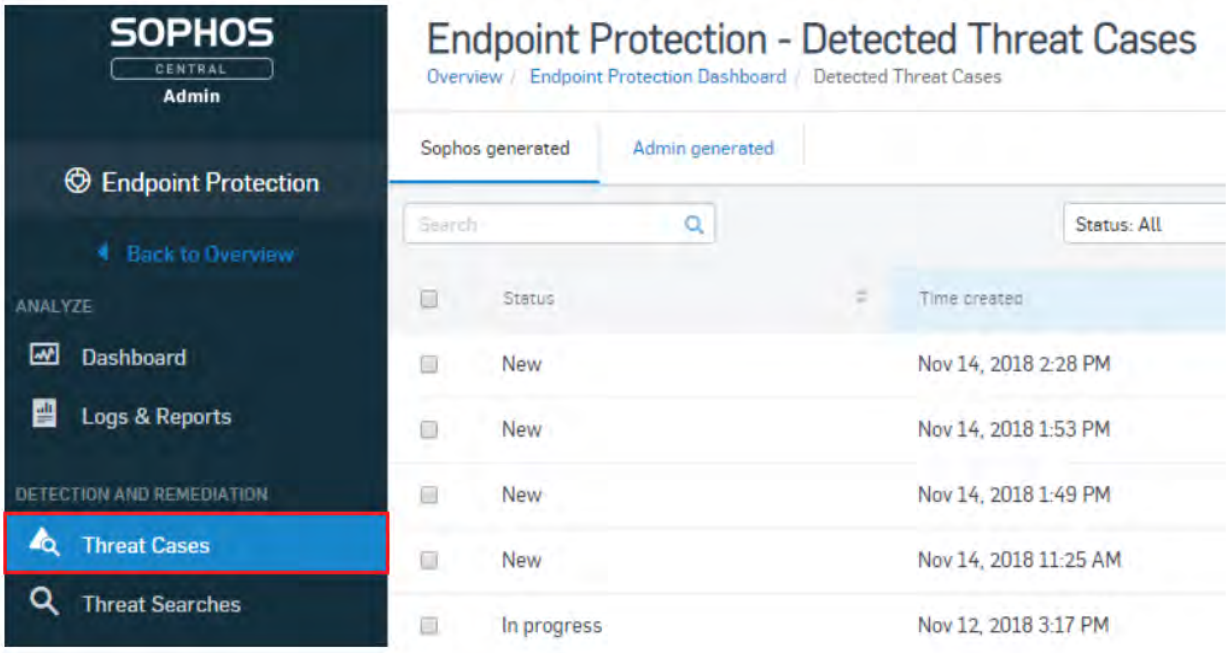
114. The Accused Products perform a method that includes *allowing the computer object to continue to run when behaviour of the computer object is permitted by the mask and disallowing the computer object to run when the actual monitored behaviour of the computer object extends beyond that permitted by the mask*. As explained and shown in the example above, the "Analyze" tab of a "threat case" displayed in the "Threat Analytics Center" illustrates the illicit execution, on an infected endpoint device, of the suspicious process "431.exe" by Microsoft Powershell, which is marked as the "Beacon," and the subsequent modification of "registry keys" by the file "431.exe" on the infected endpoint. The process "431.exe" was allowed to execute until it suspiciously modified registry keys, at which point it was quarantined.

115. The Accused Products perform a method that includes *reclassifying the computer object as malware when the actual monitored behaviour extends beyond that permitted by the mask*. As explained above, and as shown in the example above, the "Analyze" tab of a "threat case" displayed in the "Threat Analytics Center" illustrates the illicit execution, on an infected endpoint device, of the suspicious process "431.exe" by Microsoft Powershell, which is marked as the "Beacon," and the subsequent modification of "registry keys" by the file "431.exe" on the infected endpoint. The process "431.exe" was initially allowed to execute until it suspiciously modified registry keys, at which point it was classified as malware and quarantined.

116. In another example, the Accused Products detected that the process "silentrep.exe" was a variant of the malware class "ML/PE-A" and re-classified it as such.

For customers with Sophos EDR, the full list of Threat Cases can be found in the below locations:

- Endpoint Protection > Detection and Remediation > Threat Cases
- Server Protection > Analyze > Threat Cases



To view a Threat Case click on the detection **Name**:

Simplified events chain

At the top of every Sophos generated threat case (excludes Admin generated) you will see the simplified events chain. This gives you the very basic details of what happened.

(*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_USl.)

117. Each claim in the '250 Patent recites an independent invention. Neither claim 1, described above, nor any other individual claim is representative of all claims in the '250 Patent.

118. Defendant has been aware of the ’250 Patent since at least the filing of this Complaint. Further, Plaintiffs have marked their products with the ’250 Patent, including on its web site, since at least July 2020.

119. Defendant directly infringes at least claim 1 of the ’250 Patent, either literally or under the doctrine of equivalents, by performing the steps described above. For example, on information and belief, Defendant performs the claimed method in an infringing manner as described above by running this software and system to protect its own computer and network operations. On information and belief, Defendant also performs the claimed method in an infringing manner when testing the operation of the Accused Products’ and corresponding systems. As another example, Defendant performs the claimed method when providing or administering services to third parties, customers, and partners using the Accused Products.

120. Defendant’s partners, customers, and end users of its Accused Products and corresponding systems and services directly infringe at least claim 1 of the ’250 Patent, literally or under the doctrine of equivalents, at least by using the Accused Products and corresponding systems and services, as described above.

121. Defendant has actively induced and is actively inducing infringement of at least claim 1 of the ’250 Patent with specific intent to induce infringement, and/or willful blindness to the possibility that its acts induce infringement, in violation of 35 U.S.C. § 271(b). For example, Defendant encourages and induces customers with specific intent to use Sophos’ security software in a manner that infringes claim 1 of the ’250 Patent at least by offering and providing software that performs a method that infringes claim 1 when installed and operated by the customer, and by engaging in activities relating to selling, marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including the activities described below.

122. Defendant encourages, instructs, directs, and/or requires third parties—including their certified partners and/or customers—to perform the claimed method using the software, services, and systems in infringing ways, as described above.

123. Defendant further encourages and induces its customers to infringe claim 1 of the '250 Patent: 1) by making its security services available on its website, providing applications that allow users to access those services, widely advertising those services, and providing technical support and instructions to users, and 2) through activities relating to marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including their Sophos security software, and services in the United States. (*See* https://www.sophos.com/en-us/products/endpoint-antivirus/how-to-buy.aspx; https://partners.sophos.com/english/directory/search?lat=30.267153&lng=-97.7430608&dMI=100&p=1; https://secure2.sophos.com/en-us/security-news-trends/whitepapers/gated-wp/cybersecurity-system-buyers-guide.aspx%23formFrame; *see also* https://www.sophos.com/en-us/medialibrary/PDFs/factsheets/sophos-synchronized-security-ds.pdf.)

124. For example, on information and belief, Defendant shares instructions, guides, and manuals, which advertise and instruct third parties on how to use the software as described above, including at least customers and partners. (*Id.*) On further information and belief, Defendant also provides customer service and technical support to purchasers of the Accused Products and corresponding systems and services, which directs and encourages customers to perform certain actions that use the Accused Products in an infringing manner. (*Id.*)

125. Defendant and/or its partners recommend and sell the Accused Products and provide technical support for the installation, implementation, integration, and ongoing operation of the Accused Products for each individual customer. On information and belief, each customer

enters into a contractual relationship with Defendant and/or one of its partners, which obligates each customer to perform certain actions in order to use the Accused Products. Further, in order to receive the benefit of Defendant and/or its partner's continued technical support and their specialized knowledge and guidance of the operability of the Accused Products, each customer must continue to use the Accused Products in a way that infringes the '250 Patent.

126. Further, as the entity that provides installation, implementation, and integration of the Accused Products in addition to ensuring the Accused Product remains operational for each customer through ongoing technical support, on information and belief, Defendant and/or its partners affirmatively aid and abet each customer's use of the Accused Products in a manner that performs the claimed method of, and infringes, the '250 Patent.

127. Defendant also contributes to the infringement of its partners, customers, and end-users of the Accused Products by providing within the United States or importing the Accused Products into the United States, which are for use in practicing, and under normal operation practice, methods claimed in the Asserted Patents, constituting a material part of the inventions claimed, and not a staple article or commodity of commerce suitable for substantial non-infringing uses.

128. Indeed, as shown above, the Accused Products have no substantial non-infringing uses because the accused functionality, including the behavioral "threat analysis" and related functionality described above, is an integral part of the Accused Products and must be performed for the Accused Products to perform their intended purpose. For example, without compiling and analyzing data about the behavior of an object running on one or more remote computers, the Accused Products could not detect processes (running objects) that have made "[u]nusual changes to the registry" or to "search devices for signs of a suspected or known threat." (*See*

https://docs.sophos.com/central/Customer/help/en-us/central/Customer/learningContents/LiveDiscover.html). These process are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer suitably function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice the '250 patent, that functionality could not be performed.

129. Additionally, the accused functionality, including the behavioral threat analysis and related functionality described above, itself has no substantial non-infringing uses because the components, modules and methods identified above are a necessary part of that functionality. For example, without a "Threat Analysis Center" that compiles and correlates behavioral data from processes that run on the endpoints protected by the Accused Products, the Accused Products could not assess which behaviors on any given endpoint constitute a threat. These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice the '250 patent, that functionality could not be performed.

130. In addition, as shown in the detailed analysis above, the products, systems, modules, and methods provided by Defendant constitute a material part of the invention—indeed, they provide all the components, modules, and features that perform the claimed methods and systems. For example, the Accused Products and accused functionalities (including the behavioral threat analysis functionality) constitute a material part of the inventions claimed because such analysis is integral to the processes identified above (such as to generating a "mask for the

computer object that defines acceptable behaviour for the computer object") as recited in the claims of the '250 patent. None of these products are staple goods—they are sophisticated and customized cyber security and malware detection products, methods, and systems.

131. Defendant's infringing actions have continued after the original Complaint was filed. Defendant had knowledge of the '250 patent and of the specific conduct that constitutes infringement of the '250 patent at least based on the original Complaint, but has continued to engage in the infringing activity, including by selling, making, configuring, and installing the Accused Products and performing the accused functionality, and by engaging in activities that constitute inducing infringement and contributory infringement as described above.

132. On information and belief, the infringing actions of each partner, customer, and/or end-user of the Accused Products are attributable to Defendant. For example, on information and belief, Defendant directs and controls the activities or actions of its partners or others in connection with the Accused Products by contractual agreement or otherwise requiring partners or others to provide information and instructions to customers who acquire the Accused Products which, when followed, results in infringement. Defendant further directs and controls the operation of devices executing the Accused Products by programming the software which, when executed by a customer or end user, perform the claimed method of at least claim 1 of the '250 Patent.

133. Plaintiffs have suffered and continue to suffer damages, including lost profits, as a result of Defendant's infringement of the '250 Patent. Defendant is therefore liable to Plaintiffs under 35 U.S.C. § 284 for damages in an amount that adequately compensates Plaintiffs for Defendant's infringement, but no less than a reasonable royalty.

134. Plaintiffs will continue to suffer irreparable harm unless this Court preliminarily and permanently enjoins Defendant, its agents, employees, representatives, and all others acting

in concert with Defendant from infringing the ’250 Patent.

135. Defendant’s infringement of the ’250 Patent is knowing and willful. Defendant had actual knowledge of the ’250 Patent at least by the time Plaintiffs filed this lawsuit and had constructive knowledge of the ’250 Patent from at least the date Plaintiffs marked their products with the ’250 Patent and/or provided notice of the ’250 Patent on their website.

136. On information and belief, despite Defendant’s knowledge of the Asserted Patents and Plaintiffs’ patented technology, Defendant made the deliberate decision to sell products and services that they knew infringe these patents. Defendant’s continued infringement of the ’250 Patent with knowledge of the ’250 Patent constitutes willful infringement.

137. Plaintiffs’ allegations of infringement, indirect infringement, and willful infringement with respect to this patent are further set forth in Plaintiffs’ Disclosure of Asserted Claims and Preliminary Infringement Contentions Pursuant to the Court’s Standing Order Governing Patent Proceedings served July 12, 2022.

SECOND CAUSE OF ACTION
(INFRINGEMENT OF THE ’389 PATENT)

138. Plaintiffs reallege and incorporate by reference the allegations of the preceding paragraphs of this Complaint.

139. Sophos has infringed and continues to infringe one or more claims of the ’389 Patent in violation of 35 U.S.C. § 271 in this District and elsewhere in the United States and will continue to do so unless enjoined by this Court. The Accused Products, including features such as Intercept X Advanced with XDR (“Intercept X with XDR”), at least when used for their ordinary and customary purposes, practice each element of at least claim 1 of the ’389 Patent as described below.

140. For example, claim 1 of the ’389 Patent recites:

> 1. A method of classifying a computer object as malware, the method comprising:
>
> at a base computer, receiving data about a computer object from a first remote computer on which the computer object or similar computer objects are stored, wherein said data includes information about events initiated or involving the computer object when the computer object is created, configured or runs on the first remote computer, said information including at least an identity of an object initiating the event, the event type, and an identity of an object or other entity on which the event is being performed;
>
> at the base computer, receiving data about the computer object from a second remote computer on which the computer object or similar computer objects are stored, wherein said data includes information about events initiated or involving the computer object when the computer object is created, configured, or runs on the second remote computer, said information including at least an identity of an object initiating the event, the event type, and an identity of an object or other entity on which the event is being performed;
>
> storing, at the base computer, said data received from the first and second remote computers;
>
> 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; and
>
> classifying, by the base computer, the computer object as malware on the basis of said comparison.

141. The Accused Products perform each of the method steps of claim 1 of the ’389 Patent. To the extent the preamble is construed to be limiting, the Accused Products perform a *method of classifying a computer object as malware*, as further explained below. For example, Intercept X with XDR (“Intercept X”) “scans across [the] entire environment and highlight[s] suspicious activity, anomalous behavior and other IT issues.” Intercept X displays “threats,”

including processes classified as malware, in its "Sophos Central" and "Threat Analysis Center" dashboards.

> XDR builds upon that solid foundation by adding even more data and context that both increases visibility and gives the user even more insight during an investigation. This results in faster and more accurate incident detection and response. Additional data sources can include firewall, email, cloud and mobile information. For example, adding in firewall data makes it simple to correlate a malicious traffic detection by the firewall with a compromised endpoint, or to see which application is causing the office network connection to run slowly.
>
> One of the most valuable ways to use XDR is to begin with the 'macro' spotlight that gives you the tools to quickly scan across your entire environment and highlight suspicious activity, anomalous behavior and other IT issues. When an issue is identified you can then hone-in on a device of interest, pulling live data or remotely accessing the device in order to dig deeper and take remedial action.

(*See* https://secure2.sophos.com/en-us/medialibrary/Gated-Assets/white-papers/sophos-xdr-beginner-guide.pdf; *see also* https://community.sophos.com/intercept-x-endpoint/b/blog/posts/introducing-the-new-threat-analysis-center.)

142. The Accused Products perform a method that includes *at a base computer, receiving data about a computer object from a first remote computer on which the computer object or similar computer objects are stored*. For example, each endpoint on which Intercept X is installed sends data about the processes executing on it to the cloud-based "Sophos Central," which stores that data in a database and manages endpoints within a network. For example, data can be queried from each endpoint using "Live Discover" SQL queries through the "Threat Analysis Dashboard," to detect, for example, processes that have made "[u]nusual changes to the registry" or to "search devices for signs of a suspected or known threat if Sophos Central has found the threat elsewhere." Data about each process is automatically analyzed, marked as a "threat case," and displayed as such in the "Threat Analysis Center." Moreover, such data is also periodically uploaded by each endpoint to a cloud-based "Data Lake," which can be queried, for example, to obtain data about which processes executed on a given endpoint when it is offline.

Live Discover

Live Discover allows you to check the devices that Sophos Central is managing, look for signs of a threat, or assess compliance.

You can use Live Discover queries to search devices for signs of threats that haven't been detected by other Sophos features. For example:

- Unusual changes to the registry.
- Failed authentications.
- A process running that is very rarely run.

You can also search devices for signs of a suspected or known threat if Sophos Central has found the threat elsewhere, or if a user reports suspicious behavior on their device.

You can also check the compliance of each device. For example, you can search for out-of-date software or browsers with insecure settings.

This page tells you how to use Live Discover. You can also familiarize yourself with it by completing the Sophos XDR Training.

How queries work

We provide a range of queries for you to use to check your devices. You can use them as they are, or edit them (you'll need to be familiar with osquery or SQL). You can also create queries.

You can run queries to get information from different sources:

- Endpoint queries get the latest information from devices that are currently connected.

(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/learningContents/LiveDiscover.html; *see also* https://community.sophos.com/intercept-x-endpoint/b/blog/posts/introducing-the-new-threat-analysis-center.)

Data Lake queries

Data Lake queries let you search security and compliance data that your devices upload to the cloud.

You can run Data Lake queries with Live Discover, a feature in our Threat Analysis Center.

Live Discover now lets you choose which data source you use when you set up and run a query:

- Endpoints that are currently connected.
- The Data Lake in the cloud.

For help with Live Discover see Live Discover.

How the Data Lake works

We host the Data Lake and provide scheduled "hydration queries" that define which data your endpoints upload to it.

However, before you use Data Lake queries, you must make sure that data is being uploaded. To turn on uploads of data, see Data Lake uploads.

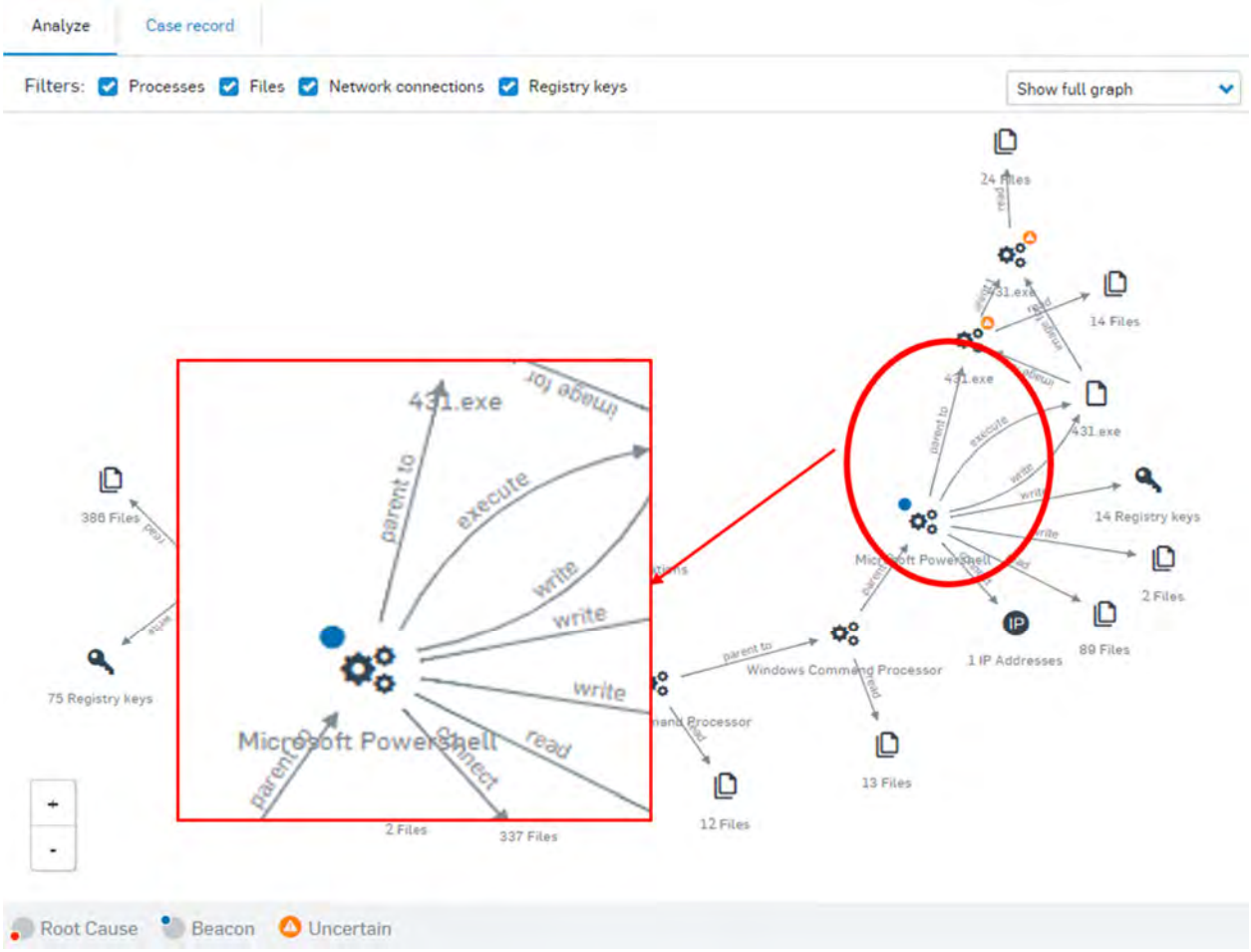
We store the data for 30 days.

We provide pre-prepared Data Lake queries you can run. You can use them as they are or edit them. You can also create your own queries.

(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/DataLakeQueries.html.)

143. The Accused Products perform a method that includes *wherein the data received from a first remote computer about a computer object includes information about events initiated or involving the computer object when the computer object is created, configured or runs on the first remote computer, said information including at least an identity of an object initiating the event, the event type, and an identity of an object or other entity on which the event is being performed*. As shown above, each endpoint on which Intercept X is installed sends data about the processes executing on it to the cloud-based "Sophos Central," which stores that data in a database and manages endpoints within a network.

144. As further evidence, the event data sent to Sophos Central by each endpoint includes event data generated when the file or process is created, configured or executed. The event data also includes incident details that describe the identity of objects and entities on which each event is performed. In particular, in the example shown below, the "Analyze" tab of a "threat case" displayed in the "Threat Analytics Center" illustrates the illicit execution, on an infected endpoint device, of the suspicious process "431.exe" by Microsoft Powershell, which is marked as the "Beacon," and the subsequent modification of "registry keys" by the file "431.exe" on the infected endpoint.



(*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_US.)

145. The Accused Products perform a method that includes wherein *data about the computer object from a second remote computer on which the computer object or similar computer objects are stored, wherein said data includes information about events initiated or involving the computer object when the computer object is created, configured, or runs on the second remote computer, said information including at least an identity of an object initiating the event, the event type, and an identity of an object or other entity on which the event is being performed.*

146. For example, as explained above, each endpoint on which Intercept X is installed sends data about the processes executing on it to the cloud-based "Sophos Central," which stores that data in a database and manages endpoints within a network. For example, data can be queried

using "Live Discover" SQL queries through the "Threat Analysis Dashboard," to detect, *e.g.*, processes that have made "[u]nusual changes to the registry" or to "search devices for signs of a suspected or known threat if Sophos Central has found the threat elsewhere." Data about each process is automatically analyzed, marked as a "threat case," and displayed as such in the "Threat Analysis Center." Moreover, such data is also periodically uploaded by each endpoint to a cloud-based "Data Lake," which can be queried, for example, to obtain data about which processes executed on a given endpoint when it is offline. The Accused Products protect endpoints associated with "more than 500,000 organizations and millions of consumers in more than 150 countries." (*See* https://www.sophos.com/en-us/press-office/press-releases/2020/11/sophos-intercept-x-named-best-endpoint-security-solution.aspx.)

Live Discover

Live Discover allows you to check the devices that Sophos Central is managing, look for signs of a threat, or assess compliance.

You can use Live Discover queries to search devices for signs of threats that haven't been detected by other Sophos features. For example:

- Unusual changes to the registry.
- Failed authentications.
- A process running that is very rarely run.

You can also search devices for signs of a suspected or known threat if Sophos Central has found the threat elsewhere, or if a user reports suspicious behavior on their device.

You can also check the compliance of each device. For example, you can search for out-of-date software or browsers with insecure settings.

This page tells you how to use Live Discover. You can also familiarize yourself with it by completing the Sophos XDR Training.

How queries work

We provide a range of queries for you to use to check your devices. You can use them as they are, or edit them (you'll need to be familiar with osquery or SQL). You can also create queries.

You can run queries to get information from different sources:

- Endpoint queries get the latest information from devices that are currently connected.

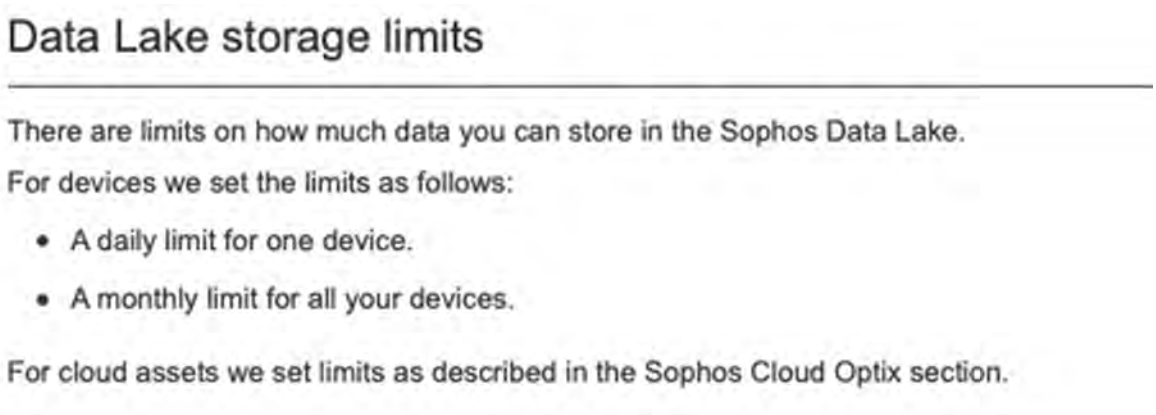
(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/learningContents/

LiveDiscover.html; *see also* https://community.sophos.com/intercept-x-endpoint/b/blog/posts/introducing-the-new-threat-analysis-center; https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/DataLake Queries.html.)

147. As further evidence, the event data sent to Sophos Central by each endpoint includes event data generated when the file or process is created, configured and executed. The event data also includes incident details that describe the identity of objects and entities on which each event is performed. In particular, in the example shown above, the "Analyze" tab of a "threat case" displayed in the "Threat Analytics Center" illustrates the illicit execution, on an infected endpoint device, of the suspicious process "431.exe" by Microsoft Powershell, which is marked as the "Beacon," and the subsequent modification of "registry keys" by the file "431.exe" on the infected endpoint.

148. The Accused Products perform a method that includes *storing, at the base computer, said data received from the first and second remote computers*. For example, as explained above, Sophos Central stores data received from every endpoint and organizes it in a database. As another example, data from endpoints is also organized into a "Data Lake," which can be queried, for example, to obtain data about which processes executed on a given endpoint when it is offline.

149. Sophos Central stores data from remote computers in a centralized "Data Lake."



Data Lake storage limits

There are limits on how much data you can store in the Sophos Data Lake.

For devices we set the limits as follows:

- A daily limit for one device.
- A monthly limit for all your devices.

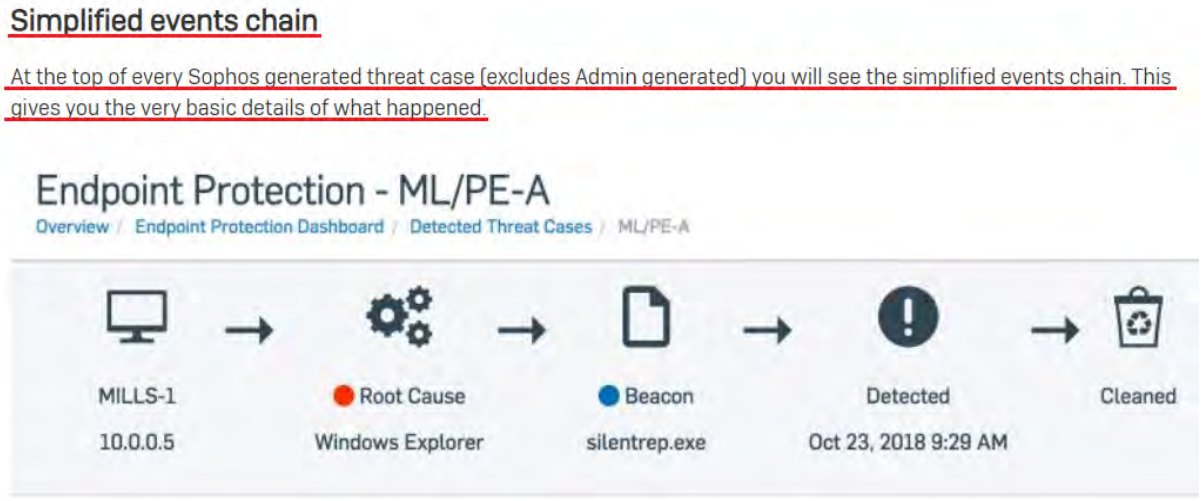
For cloud assets we set limits as described in the Sophos Cloud Optix section.

(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/DataLakeStorageLimits.html.)

150. The Accused Products perform a method that includes *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*. For example, each endpoint on which Intercept X is installed sends data about the processes executing on it to the cloud-based "Sophos Central," which stores and organizes that data in a database and manages endpoints within a network. Data about those processes, such as which actions of events they have initiated on their endpoints, and which other endpoints also ran processes initiating such actions, are correlated within that database, and can be queried on the basis of those correlations to, for example, "search devices for signs of a suspected or known threat if Sophos Central has found the threat elsewhere."

151. As another example, by using "Live Discover" SQL queries to a correlated database through the "Threat Analysis Dashboard," a system administrator can obtain a list of processes, across all connected endpoints, that have made certain "[u]nusual changes to the registry" or "Failed Authentications" in a particular way. (*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/learningContents/ LiveDiscover.html.)

152. The Accused Products perform a method that includes *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.* As explained above, the Accused Products use the data on processes that each endpoint sends to Sophos Central to identify relationships between those processes and malware to identify "threat cases." These comparisons allow the Accused Products to search for signs of a suspected or known threat if Sophos Central has found the threat elsewhere. In the example below, Intercept X has identified that the process "silentrep.exe" is a variant of the malware "ML/PE-A."



(*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_USl.)

153. The Accused Products perform a method that includes *classifying, by the base computer, the computer object as malware based on said comparison*. For example, as explained above, Intercept X identified that the data Sophos Central received about the process "silentrep.exe" indicated that it was a variant of malware "ML/PE-A" and on the basis of that comparison, classified it as malware.

154. Each claim in the '389 Patent recites an independent invention. Neither claim 1,

described above, nor any other individual claim is representative of all claims in the ’389 Patent.

155. Sophos has been aware of the ’389 Patent since at least the filing of this Complaint. Further, Plaintiffs have marked its products with the ’389 Patent, including on its web site, since at least July 2020.

156. Defendant directly infringes at least claim 1 of the ’389 Patent, literally or under the doctrine of equivalents, by performing the steps described above. For example, on information and belief, Defendant performs the claimed method in an infringing manner as described above by running this software and corresponding systems to protect its own computer and network operations. On information and belief, Defendant also performs the claimed method as described above when testing the operation of the Accused Products and corresponding systems. As another example, Defendant performs the claimed method as described above when providing or administering services to third parties, customers, and partners using the Accused Products.

157. Defendant’s partners, customers, and end users of the Accused Products and corresponding systems and services directly infringe at least claim 1 of the ’389 Patent, literally or under the doctrine of equivalents, at least by using the Accused Products and corresponding systems and services, as described above.

158. Defendant has actively induced and are actively inducing infringement of at least claim 1 of the ’389 Patent with specific intent to induce infringement, and/or willful blindness to the possibility that its acts induce infringement, in violation of 35 U.S.C. § 271(b). For example, Defendant encourages and induces customers with specific intent to use Sophos’s security software in a manner that infringes claim 1 of the ’389 Patent at least by offering and providing software that performs a method that infringes claim 1 when installed and operated by the customer, and by engaging in activities relating to selling marketing, advertising, promotion,

installation, support, and distribution of the Accused Products, including the activities described below.

159. Defendant encourages, instruct, direct, and/or require third parties—including its certified partners and/or customers—to perform the claimed method using the software, services, and systems in infringing ways, as described above.

160. Defendant further encourages and induces its customers to infringe claim 1 of the '389 Patent: 1) by making its security services available on its website, providing applications that allow users to access those services, widely advertising those services, and providing technical support and instructions to users, and 2) through activities relating to marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including its Sophos security software and services in the United States. (*See* https://www.sophos.com/en-us/products/endpoint-antivirus/how-to-buy.aspx; https://partners.sophos.com/english/directory/search?lat=30.267153&lng=-97.7430608&dMI=100&p=1; https://secure2.sophos.com/en-us/security-news-trends/whitepapers/gated-wp/cybersecurity-system-buyers-guide.aspx%23formFrame; *see also* https://www.sophos.com/en-us/medialibrary/PDFs/factsheets/sophos-synchronized-security-ds.pdf.)

161. For example, on information and belief, Defendant shares instructions, guides, and manuals, which advertise and instruct third parties on how to use the software as described above, including at least customers and partners. (*Id.*) On further information and belief, Defendant also provides customer service and technical support to purchasers of the Accused Products and corresponding system and services, which directs and encourages customers to perform certain actions that use the Accused Products in an infringing manner. (*Id.*)

162. Defendant and/or its partners recommend and sell the Accused Products and

provide technical support for the installation, implementation, integration, and ongoing operation of the Accused Products for each individual customer. On information and belief, each customer enters into a contractual relationship with Defendant and/or one of its partners, which obligates each customer to perform certain actions in order to use the Accused Products. Further, in order to receive the benefit of Defendant and/or its partner's continued technical support and their specialized knowledge and guidance of the operability of the Accused Products, each customer must continue to use the Accused Products in a way that infringes the '389 Patent.

163. Further, as the entity that provides installation, implementation, and integration of the Accused Products in addition to ensuring the Accused Product remains operational for each customer through ongoing technical support, on information and belief, Defendant and/or its partners affirmatively aid and abet each customer's use of the Accused Products in a manner that performs the claimed method of, and infringes, the '389 Patent.

164. Defendant also contributes to the infringement of its partners, customers, and end-users of the Accused Products by providing within the United States or importing the Accused Products into the United States, which are for use in practicing, and under normal operation practice, methods claimed in the Asserted Patents, constituting a material part of the claimed methods, and not a staple article or commodity of commerce suitable for substantial non-infringing uses. Indeed, as shown above, the Accused Products and the example functionality described below have no substantial non-infringing uses but are specifically designed to practice the '389 Patent.

165. Indeed, as shown above, the Accused Products have no substantial non-infringing uses because the accused functionality, including the behavioral "threat analysis" and related functionality described above, is an integral part of the Accused Products and must be performed

for the Accused Products to perform their intended purpose. For example, without compiling and analyzing data about the behavior of an object running on one or more remote computers, the Accused Products could not detect processes (running objects) that have made "[u]nusual changes to the registry" or to "search devices for signs of a suspected or known threat." (*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/learningContents/LiveDiscover.html). These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer suitably function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice the '389 patent, that functionality could not be performed.

166. Additionally, the accused functionality, including the accused behavioral threat analysis functionality described above, itself has no substantial non-infringing uses because the components, modules and methods identified above are a necessary part of that functionality. For example, without a "Threat Analysis Center" that compiles and correlates behavioral data from processes that run on the endpoints protected by the Accused Products, the Accused Products could not assess which behaviors on any given endpoint constitute a threat. These process are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice the '389 patent, that functionality could not be performed.

167. In addition, as shown in the detailed analysis above, the products, systems, modules, and methods provided by Defendant constitute a material part of the invention—indeed,

it provides all the components, modules, and features that perform the claimed methods and systems. For example, the Accused Products and corresponding functionality (including the behavioral threat analysis functionality) constitute a material part of the inventions claimed because such analysis is integral to the processes identified above (such as to "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," and "classifying, by the base computer, the computer object as malware on the basis of said comparison"), as required by the claims of the '389 patent. None of these products are staple goods—they are sophisticated and customized cyber security and malware detection products and systems.

168. Defendant's infringing actions have continued after the original Complaint was filed. Defendant had knowledge of the '389 patent, and of the specific conduct that constitutes infringement of the '389 patent at least based on the original Complaint, but has continued to engage in the infringing activity, including by selling, making, configuring, and installing the Accused Products and performing the accused functionality, and by engaging in activities that constitute inducing infringement and contributory infringement as described above.

169. On information and belief, the infringing actions of each partner, customer, and/or end-user of the Accused Products are attributable to Defendant. For example, on information and belief, Defendant directs and controls the activities or actions of its partners or others in connection with the Accused Products by contractual agreement or otherwise requiring partners or others to

provide information and instructions to customers who acquire the Accused Products which, when followed, results in infringement. Defendant further directs and controls the operation of devices executing the Accused Products by programming the software which, when executed by a customer or end user, perform the method steps of at least claim 1 of the ’389 Patent.

170. Plaintiffs have suffered and continue to suffer damages, including lost profits, as a result of Defendant’s infringement of the ’389 Patent. Defendant is therefore liable to Plaintiffs under 35 U.S.C. § 284 for damages in an amount that adequately compensates Plaintiffs for Defendant’s infringement, but no less than a reasonable royalty.

171. Plaintiffs will continue to suffer irreparable harm unless this Court preliminarily and permanently enjoins Defendant, its agents, employees, representatives, and all others acting in concert with Defendant from infringing the ’389 Patent.

172. Defendant’s infringement of the ’389 Patent is knowing and willful. Defendant had actual knowledge of the ’389 Patent at least by the time Plaintiffs filed this lawsuit and had constructive knowledge of the ’389 Patent from at least the date Plaintiffs marked their products with the ’389 Patent and/or provided notice of the ’389 Patent on their website.

173. On information and belief, despite Defendant’s knowledge of the ’389 Patent, and Plaintiffs’ patented technology, Defendant made the deliberate decision to sell products and services that they knew infringe the ’389 Patent. Defendant’s continued infringement of the ’389 Patent with knowledge of the ’389 Patent constitutes willful infringement.

174. Plaintiffs’ allegations of infringement, indirect infringement, and willful infringement with respect to this patent are further set forth in Plaintiffs’ Disclosure of Asserted Claims and Preliminary Infringement Contentions Pursuant to the Court’s Standing Order Governing Patent Proceedings served July 12, 2022.

THIRD CAUSE OF ACTION
(INFRINGEMENT OF THE ’045 PATENT)

175. Plaintiffs reallege and incorporates by reference the allegations of the preceding paragraphs of this Complaint.

176. Sophos has infringed and continues to infringe one or more claims of the ’045 Patent in violation of 35 U.S.C. § 271 in this District and elsewhere in the United States and will continue to do so unless enjoined by this Court. The Accused Products, including features such as Sophos’ Intercept X Advanced with EDR (“Intercept X”), at least when used for their ordinary and customary purposes, practice each element of at least claim 1 of the ’045 Patent as described below.

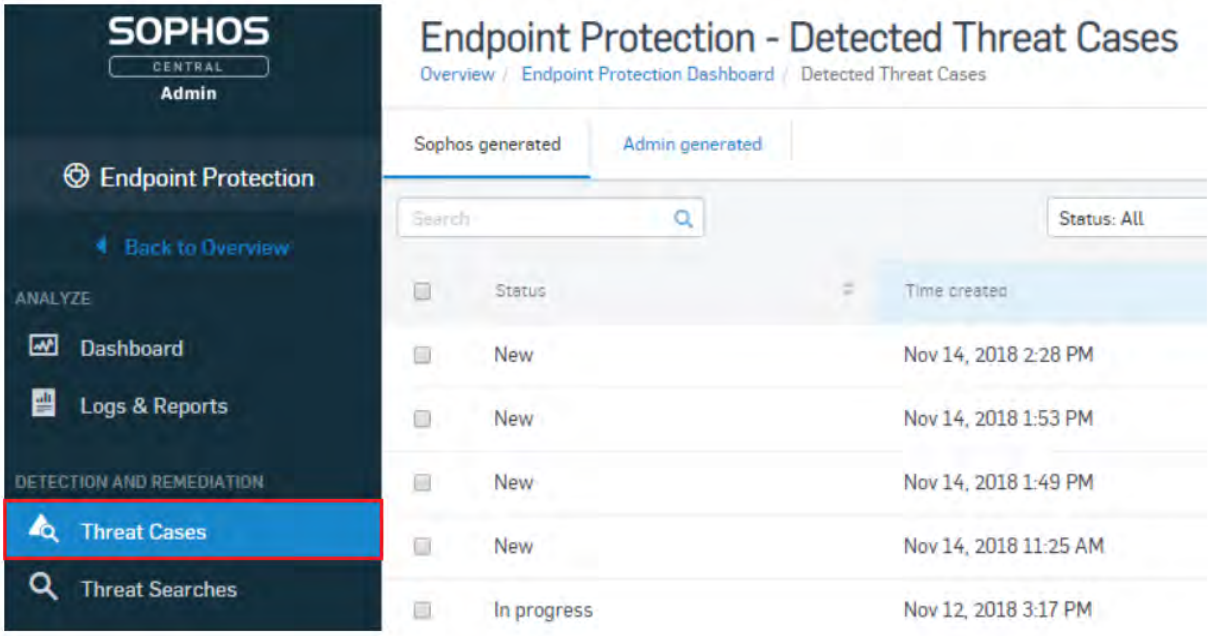
177. Claim 1 of the ’045 Patent recites:

> 1. A method comprising:
>
> gathering one or more events defining an action of a first object acting on a target;
>
> generating a contextual state for at least one of the one or more events by correlating the at least one event to an originating object, the contextual state including an indication of the originating object of the first object and an indication of at least one of a device on which the first object is executed and a user associated with the first object;
>
> obtaining a global perspective for the at least one event by obtaining information associated with one or more of the first object and the originating object, the information including at least one of age, popularity, a determination as to whether the first object is malware, a determination as to whether the originating object is malware, Internet Protocol (IP) Address, and Uniform Resource Locator (URL) information, wherein the global perspective for one or more related events to at least one event across a network;
>
> assembling an event line including details associated with the at least one event, the details including information uniquely identifying the first object, the action of the first object, the target, and the originating object; and
>
> transmitting the assembled event line.

178. The Accused Products perform each of the method steps of claim 1 of the '045 Patent. To the extent the preamble is construed to be limiting, the Accused Products perform a *method* as further explained below. For example, Intercept X performs a method for endpoint protection, wherein threat cases/attacks are analyzed in detail.

For customers with Sophos EDR, the full list of Threat Cases can be found in the below locations:

- Endpoint Protection > Detection and Remediation > Threat Cases
- Server Protection > Analyze > Threat Cases



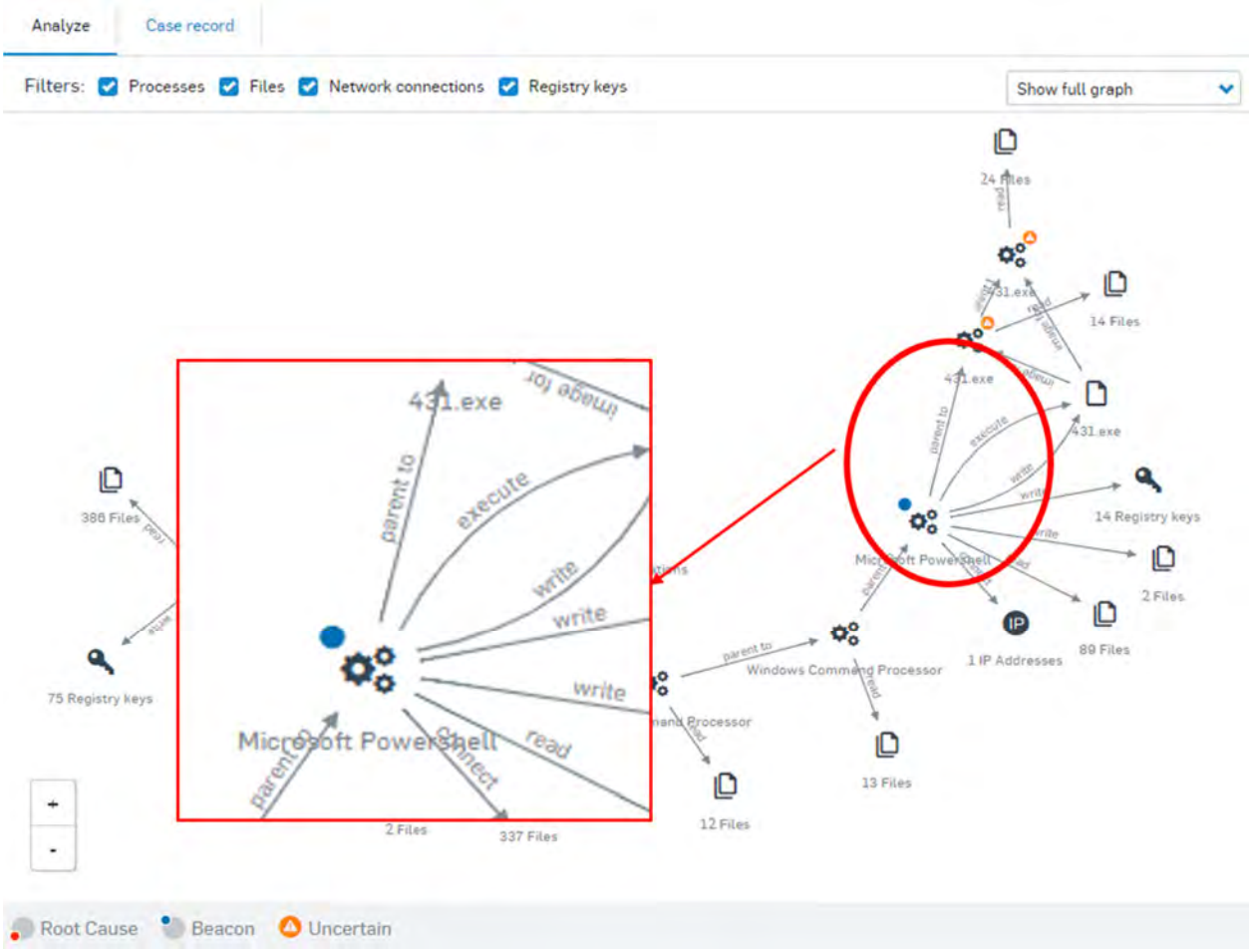
To view a Threat Case click on the detection **Name**:

Simplified events chain

At the top of every Sophos generated threat case (excludes Admin generated) you will see the simplified events chain. This gives you the very basic details of what happened.

(*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_USl.)

179. The Accused Products perform a method that includes *gathering one or more events defining an action of a first object acting on a target*. In the example shown below, the "Analyze" tab of Intercept X illustrates the illicit creation of the malicious file "431.exe" by Microsoft Powershell, which is marked as the "Beacon," and the subsequent actions of "431.exe" on the infected endpoint, such as modifying registry keys. The Analyze Tab describes the "attack chain," *i.e.*, the chain of events linking the "Beacon" to what Intercept X has identified as the "Root Cause," in this case the program "Outlook." In the example below, Outlook wrote a document called "rgnr-avr11205-85.doc" and used Microsoft Office to read the document, initiating a chain of events culminating in the illicit creation and execution of the malicious file "431.exe" via Microsoft Powershell. Events defining the attack chain are gathered from the endpoint device by Intercept X (*e.g.*, Sophos Central).



(*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_US.)

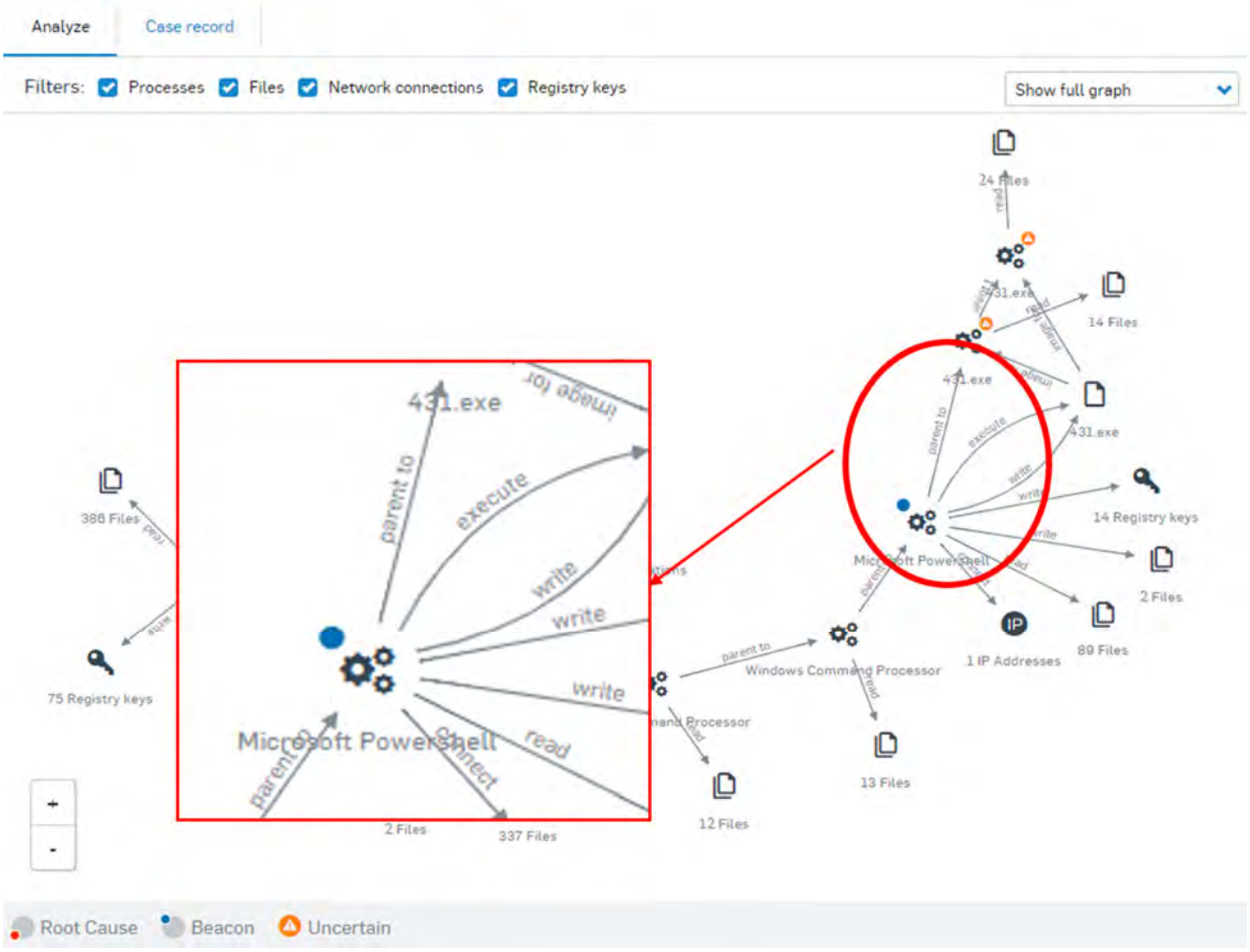
180. The root cause analysis performed by Intercept X, and illustrated by an attack chain, is further described in the video "Root Cause Analysis RCA in 2 minutes."

(*See* https://www.youtube.com/watch?v=AosjUjp4P7Q (showing the root cause as the process circled in red above).)

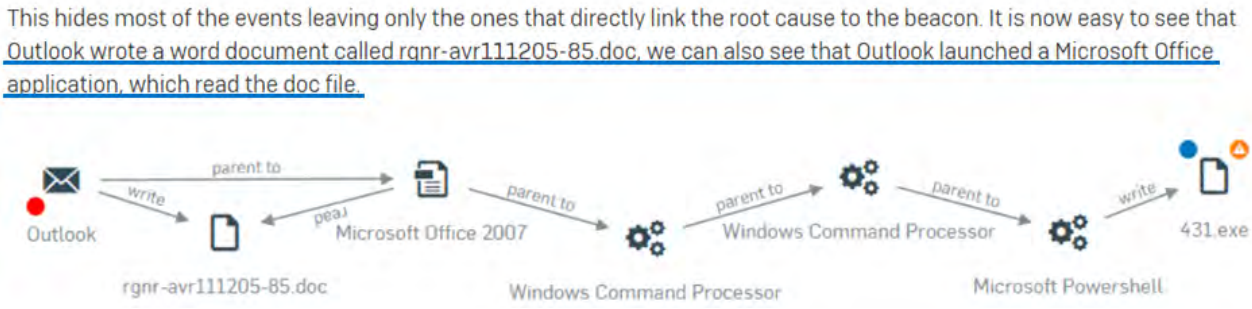
181. The Accused Products perform a method that includes *generating a contextual state for at least one of the one or more events by correlating the at least one event to an originating object, the contextual state including an indication of the originating object of the first object and an indication of at least one of a device on which the first object is executed and a user associated with the first object*. As explained above, Intercept X's "Analyze" tab illustrates the creation of a malicious file "431.exe" by Microsoft Powershell, which is marked as the "Beacon."

182. In the example below, the Analyze Tab describes the "attack chain," *i.e.*, the chain of associated events linking the "Beacon" to what Intercept X has identified as the "Root Cause," in this case the program "Outlook." Outlook wrote a document called "rgnr-avr11205-85.doc" and used Microsoft Office to read the document, initiating a chain of events culminating in the illicit creation and execution of the malicious file "431.exe" via Microsoft Powershell. An "attack chain," also known as an "event chain," is associated both with the endpoint device on which files

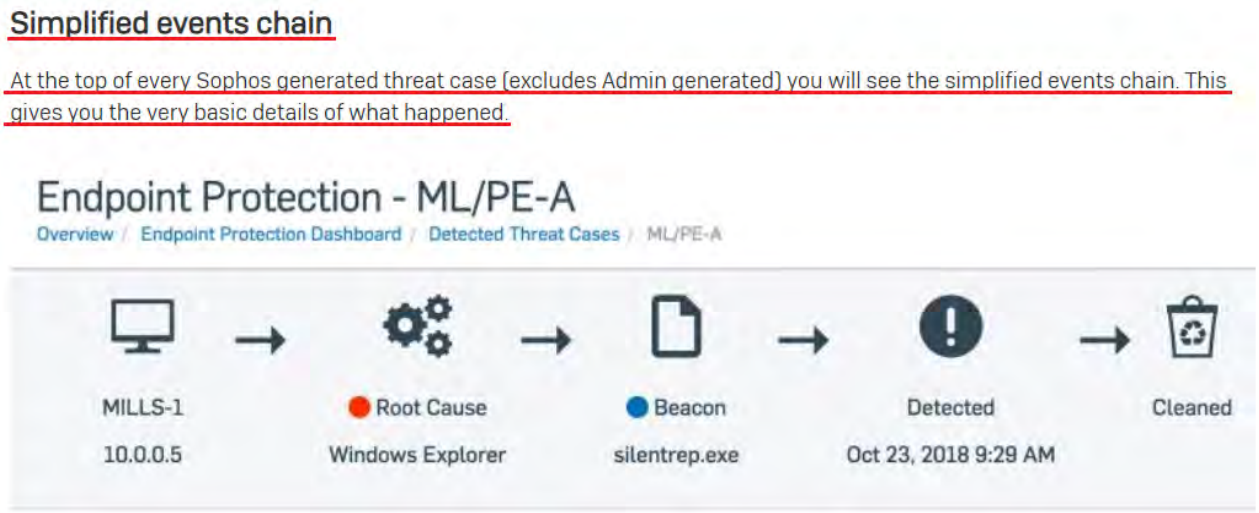
executed and the user of that device.



(*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_US.)



(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html.)



(*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_USl.)

183. The details of the attack chain, and the relationships it illustrates between the "Root Cause," the "Beacon," and the intervening files or processes between them, include the "*contextual state for at least one of the one or more events by correlating the at least one event to an originating object, the contextual state including an indication of the originating object of the first object.*" For example, the attack chain above shows each step in the attack (*e.g.*, files or registry keys read or written by any program, IP addresses accessed, caller-callee relationships, and more).

184. The Accused Products perform a method that includes *obtaining a global perspective for the at least one event by obtaining information associated with one or more of the first object and the originating object, the information including at least one of age, popularity, a determination as to whether the first object is malware, a determination as to whether the originating object is malware, Internet Protocol (IP) Address, and Uniform Resource Locator (URL) information, wherein the global perspective for one or more related events to the at least one event across a network.* The attack chain includes information obtained about the "Root Cause," the "Beacon," events involving them, as well as the intervening files or processes from associated events across a network. For example, Intercept X obtains information at least about

the age, popularity and URL information of the processes within the attack chain.

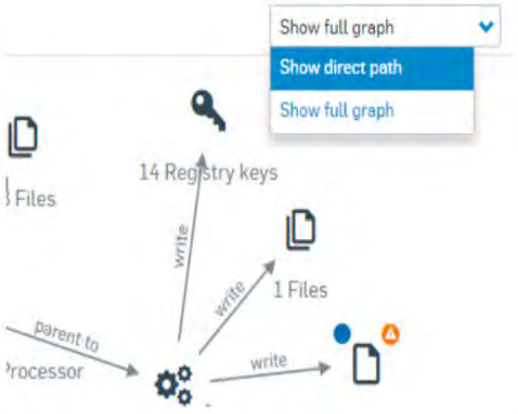
Report summary

Under **Report summary**, you can see the file's reputation and prevalence and the results of our machine learning analysis, which indicate how suspicious the file is.

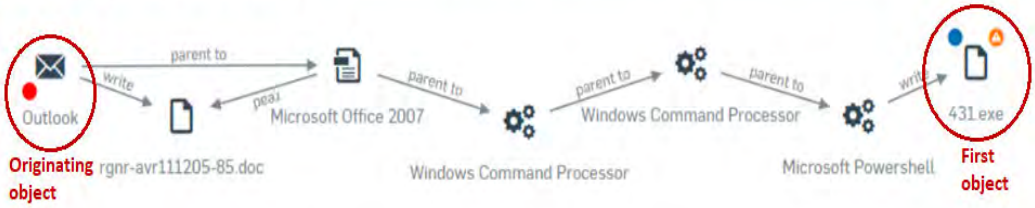
Setting	Description
Prevalence	Indicates how often SophosLabs has seen the file.
First seen	When SophosLabs first saw the file in the wild.
Last seen	When SophosLabs last saw the file in the wild
Machine learning analysis	Summarizes how suspicious the file is.

(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/

ProcessDetails.html.)



This hides most of the events leaving only the ones that directly link the root cause to the beacon. It is now easy to see that Outlook wrote a word document called rgnr-avr111205-85.doc, we can also see that Outlook launched a Microsoft Office application, which read the doc file.

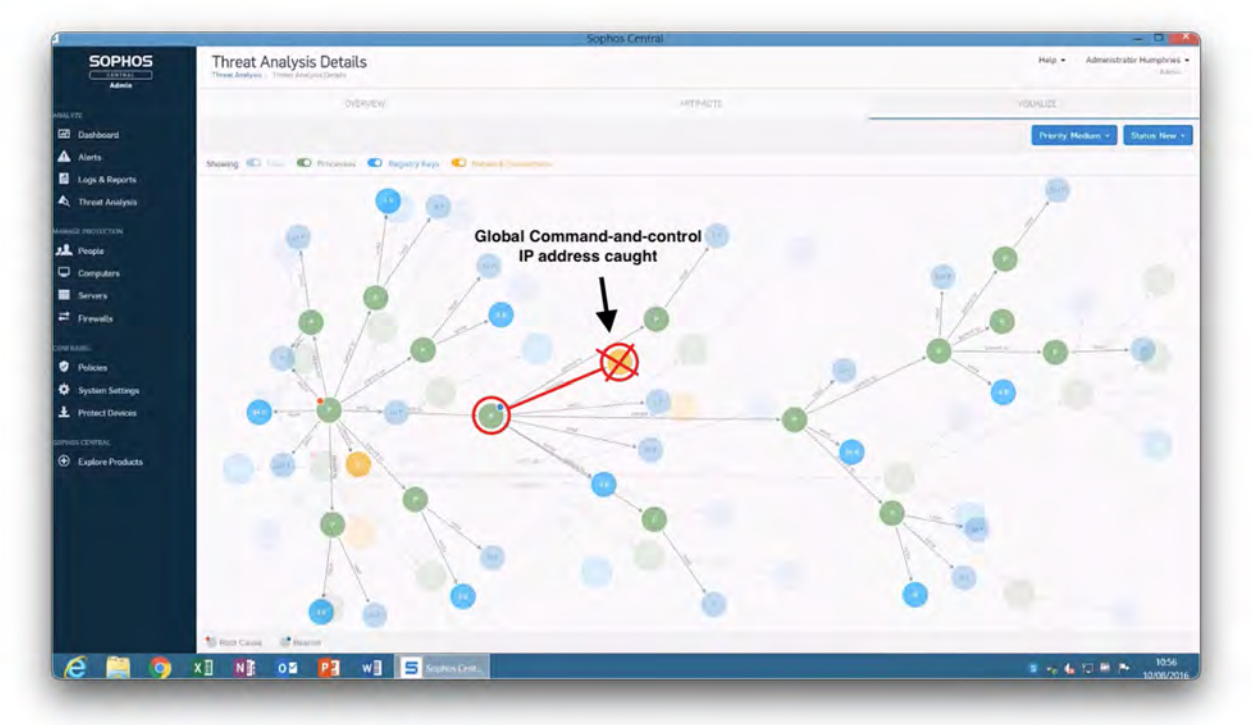


We can already see that the Microsoft Office event does not have a reputation icon, which means it has a good reputation.

Note: Reputation is only calculated for Portable Executable (PE) files, for example .exe, .dll. It is not shown for other file types such as .doc, .pdf, .png.

(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html; *see also* https://www.youtube.com/watch?v=_ujOT58ZvpI.)

185. Moreover, the Accused Products identify, for example, global IP addresses that are malware command-and-control centers: "Here's the beacon event, the thing that was caught, in this case a fake salary report. Here's the root, Google Chrome. We see here that the fake report reached out to an IP address that we happen to know to be a command-and-control site. This right here is the moment of conviction, when we discovered that what was executing was a piece of malware."

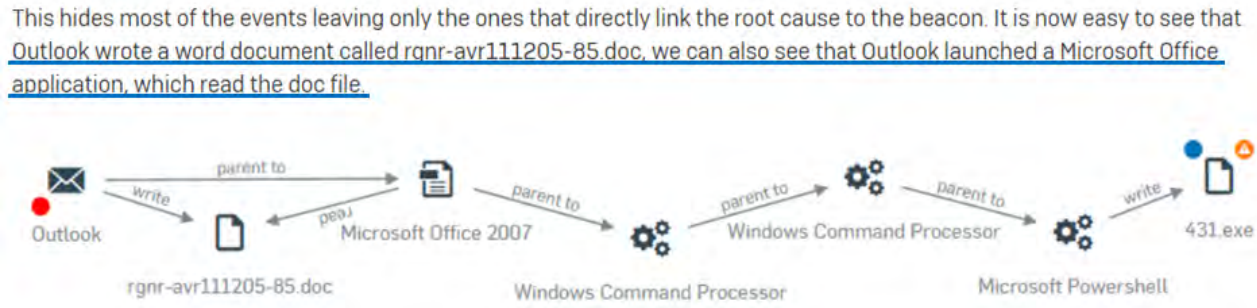


(*See* https://youtube/AOsjUjp4P7Q?t=48.)

186. The Accused Products perform a method that includes *assembling an event line including details associated with the at least one event, the details including information uniquely identifying the first object, the action of the first object, the target, and the originating object*. As

explained above, an attack chain created by Intercept X includes information associated with, and identifying the "Root Cause," the "Beacon," and the intervening files or processes in the attack chain. In the example included, Intercept X's "Analyze" tab illustrates the illicit creation of the malicious file "431.exe" by Microsoft Powershell, and the subsequent actions of "431.exe" on the infected endpoint, such as modifying registry keys.

187. In the example below, the Analyze Tab describes the "attack chain," *i.e.*, the chain of events linking the "Beacon" to what Intercept X has identified as the "Root Cause," in this case the program "Outlook." Outlook wrote a document called "rgnr-avr11205-85.doc" and used Microsoft Office to read the document, initiating a chain of events culminating in the illicit creation and execution of the malicious file "431.exe" via Microsoft Powershell. The attack chain, clearly seen as a chain of arrows connecting objects and their next targets, also illustrates the subsequent actions of "431.exe" on the infected endpoint, such as modifying registry keys. (*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_US.)

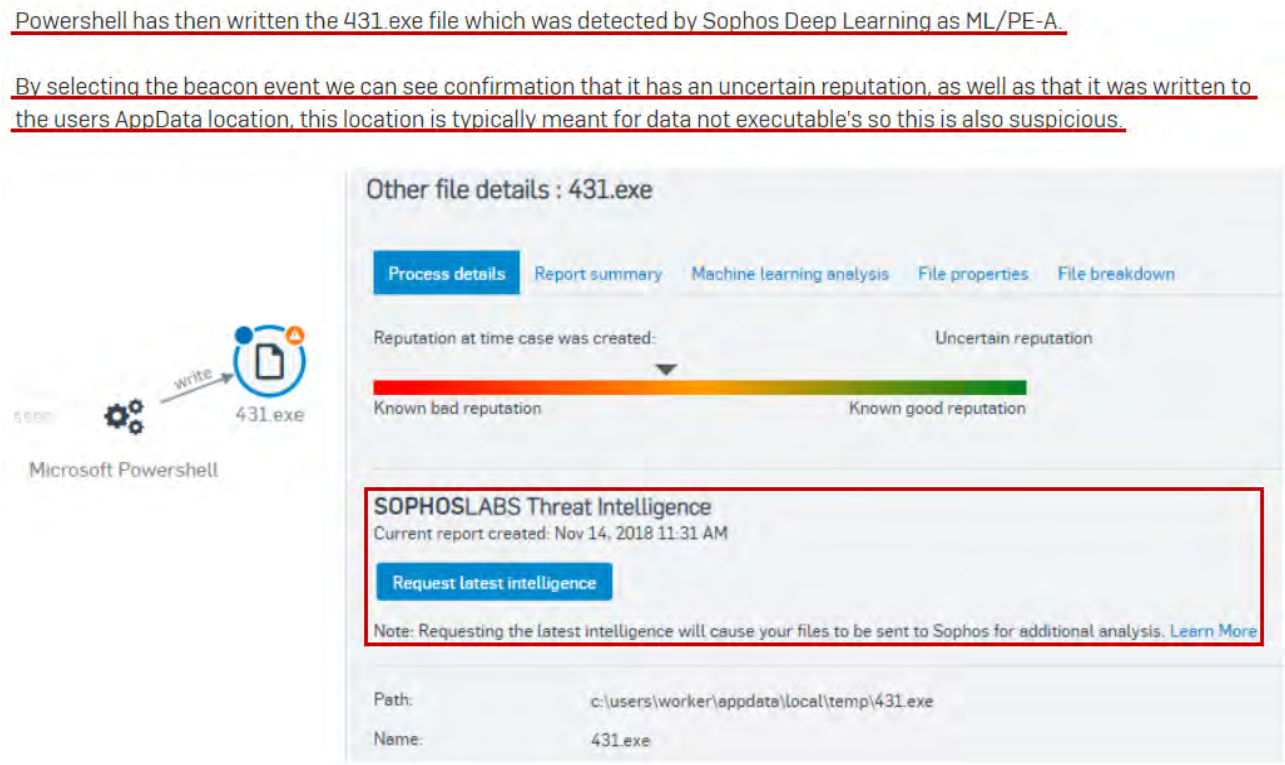


(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html.)

188. The Accused Products perform a method of *transmitting the assembled event line*. In the example below, and as explained above, Intercept X's "Analyze" tab illustrates the "attack chain" linking the illicit creation of the malicious file "431.exe" by Microsoft Powershell, *i.e.*, "the Beacon," to the "Root Cause." Intercept X thus transmits the event line such that the attack chain

can be generated, stored or displayed (*e.g.*, on a user or administrator's client-side web browser).

189. Furthermore, Intercept X transmits the event line to SophosLABS for "additional analysis."



(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html.)

190. Each claim in the '045 Patent recites an independent invention. Neither claim 1, described above, nor any other individual claim is representative of all claims in the '045 Patent.

191. Defendant has been aware of the '045 Patent since at least the filing of this Complaint. Further, Plaintiffs have marked its products with the '045 Patent, including on its web site, since at least July 2020.

192. Defendant directly infringes at least claim 1 of the '045 Patent, literally or under the doctrine of equivalents, by performing the steps described above. For example, on information and belief, Defendant performs the claimed method as described above by running this software

and corresponding systems to protect its own computer and network operations. On information and belief, Defendant also performs the claimed method as described above when testing the operation of the Accused Products and corresponding systems. As another example, Defendant performs the claimed method as described above when providing or administering services to third parties, customers, and partners using the Accused Products.

193. Defendant's partners, customers, and end users of the Accused Products and corresponding systems and services directly infringe at least claim 1 of the '045 Patent, literally or under the doctrine of equivalents, at least by using the Accused Products and corresponding systems and services, as described above.

194. Defendant has actively induced and is actively inducing infringement of at least claim 1 of the '045 Patent with specific intent to induce infringement, and/or willful blindness to the possibility that its acts induce infringement, in violation of 35 U.S.C. § 271(b). For example, Defendant encourages and induces customers with specific intent to use Sophos security software in a manner that infringes claim 1 of the '045 Patent at least by offering and providing software that performs a method that infringes claim 1 when installed and operated by the customer, and by engaging in activities relating to selling, marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including the activities described below.

195. Defendant encourages, instructs, directs, and/or requires third parties—including its certified partners and/or customers—to perform the claimed method using the software, services, and systems in infringing ways, as described above.

196. Defendant further encourages and induces its customers to infringe claim 1 of the '045 Patent: 1) by making its security services available on its website, providing applications that allow users to access those services, widely advertising those services, and providing technical

support and instructions to users, and 2) through activities relating to marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including its Sophos security software, and services in the United States. (*See* https://www.sophos.com/en-us/products/endpoint-antivirus/how-to-buy.aspx; https://partners.sophos.com/english/directory/search?lat=30.267153&lng=-97.7430608&dMI=100&p=1; https://secure2.sophos.com/en-us/security-news-trends/whitepapers/gated-wp/cybersecurity-system-buyers-guide.aspx%23formFrame.)

197. For example, on information and belief, Defendant shares instructions, guides, and manuals, which advertise and instruct third parties on how to use the software as described above, including at least customers and partners. (*Id.*) On further information and belief, Sophos also provides customer service and technical support to purchasers of the Accused Products and corresponding system and services, which directs and encourages customers to perform certain actions that use the Accused Products in an infringing manner. (*Id.*)

198. Defendant and/or its partners recommend and sell the Accused Products and provide technical support for the installation, implementation, integration, and ongoing operation of the Accused Products for each individual customer. On information and belief, each customer enters into a contractual relationship with Defendant and/or one of its partners, which obligates each customer to perform certain actions as a condition to use of the Accused Products. Further, in order to receive the benefit of Defendant and/or its partner's continued technical support and their specialized knowledge and guidance of the operability of the Accused Products, each customer must continue to use the Accused Products in a way that infringes the '045 Patent.

199. Further, as the entity that provides installation, implementation, and integration of the Accused Products in addition to ensuring the Accused Product remains operational for each

customer through ongoing technical support, on information and belief, Defendant and/or its partners affirmatively aid and abet each customer's use of the Accused Products in a manner that performs the claimed method of, and infringes, the '045 Patent.

200. Defendant also contributes to the infringement of their partners, customers, and end-users of the Accused Products by providing within the United States or importing the Accused Products into the United States, which are for use in practicing, and under normal operation practice, methods claimed in the Asserted Patents, constituting a material part of the claimed methods, and not a staple article or commodity of commerce suitable for substantial non-infringing uses. Indeed, as shown above, the Accused Products and the example functionality described below have no substantial non-infringing uses but are specifically designed to practice the '045 Patent.

201. Indeed, as shown above, the Accused Products have no substantial non-infringing uses because the accused functionality, including the "Root Cause Analysis," "Threat Anlaysis," and related functionality described above, is an integral part of the Accused Products and must be performed for the Accused Products to perform their intended purpose. For example, without analyzing data about the behavior and globally known information about the maliciousness of an object running on one or more remote computers, the Accused Products could not detect when that object had engaged in an illicit behavior, and could not create a forensic trail associating that object with its root cause. These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer suitably function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice in the '045 patent, that functionality could not be performed.

202. Additionally, the accused functionality, including the "Root Cause Analysis," "Threat Anlaysis," and related functionality described above, itself has no substantial non-infringing uses because the components, modules and methods identified above are a necessary part of that functionality. For example, without creating an "attack chain" that compiles and correlates behavioral data from processes that run on the endpoints protected by the Accused Products, the Accused Products could not describe how and why the illicit behaviors detected on any given endpoint constitute a threat. These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice the '045 patent, that functionality could not be performed.

203. In addition, as shown in the detailed analysis above, the products, systems, modules, and methods provided by Defendant constitute a material part of the invention--indeed, they provide all the components, modules, and features that perform the claimed methods and systems. For example, the Accused Products and accused functionalities (including the "Root Cause Analysis"/"Threat Anlaysis" functionality) constitute a material part of the inventions claimed at least because they are integral to the processes identified above (such as "generating a contextual state . . ."; "obtaining a global perspective . . ."; "assembling" and "transmitting" an event line"), as recited in the claims of the '045 patent. None of these products are staple goods—they are sophisticated and customized cyber security and malware detection products, methods, and systems.

204. Defendant's infringing actions have continued after the original Complaint was

filed. Defendant had knowledge of the ’045 patent, and of the specific conduct that constitutes infringement of the ’045 patent at least based on the original Complaint, but has continued to engage in the infringing activity, including by selling making, configuring, and installing the Accused Products and performing the accused functionality, and by engaging in activities that constitute inducing infringement and contributory infringement as described above.

205. On information and belief, the infringing actions of each partner, customer, and/or end-user of the Accused Products are attributable to Defendant. For example, on information and belief, Defendant directs and controls the activities or actions of its partners or others in connection with the Accused Products by contractual agreement or otherwise requiring partners or others to provide information and instructions to customers who acquire the Accused Products which, when followed, results in infringement. Defendant further directs and controls the operation of devices executing the Accused Products by programming the software which, when executed by a customer or end user, perform the method steps of at least claim 1 of the ’045 Patent.

206. Plaintiffs have suffered and continue to suffer damages, including lost profits, as a result of Defendant’s infringement of the ’045 Patent. Defendant is therefore liable to Plaintiffs under 35 U.S.C. § 284 for damages in an amount that adequately compensates Plaintiffs for Defendant’s infringement, but no less than a reasonable royalty.

207. Plaintiffs will continue to suffer irreparable harm unless this Court preliminarily and permanently enjoins Defendant, its agents, employees, representatives, and all others acting in concert with Defendant from infringing the ’045 Patent.

208. Defendant’s infringement of the ’045 Patent is knowing and willful. Defendant had actual knowledge of the ’045 Patent at least by the time Plaintiffs filed this lawsuit and had

constructive knowledge of the ’045 Patent from at least the date Plaintiffs marked their products with the ’045 Patent and/or provided notice of the ’045 Patent on their website.

209. On information and belief, despite Defendant’s knowledge of the Asserted Patents and Plaintiffs’ patented technology, Defendant made the deliberate decision to sell products and services that they knew infringe the ’045 Patent. Defendant’s continued infringement of the ’045 Patent with knowledge of the ’045 Patent constitutes willful infringement.

210. Plaintiffs’ allegations of infringement, indirect infringement, and willful infringement with respect to this patent are further set forth in Plaintiffs’ Disclosure of Asserted Claims and Preliminary Infringement Contentions Pursuant to the Court’s Standing Order Governing Patent Proceedings served July 12, 2022.

FOURTH CAUSE OF ACTION
(INFRINGEMENT OF THE ’224 PATENT)

211. Plaintiffs reallege and incorporate by reference the allegations of the preceding paragraphs of this Complaint.

212. Sophos has infringed and continues to infringe one or more claims of the ’224 Patent in violation of 35 U.S.C. § 271 in this District and elsewhere in the United States and will continue to do so unless enjoined by this Court. The Accused Products, including features such as Sophos’s Intercept X Advanced with EDR (“Intercept X”), at least when used for their ordinary and customary purposes, practice each element of at least claim 1 of the ’224 Patent as described below.

213. Claim 1 of the ’224 Patent recites:

> 1. A method comprising:
>
> gathering an event defining an action of a first object acting on a target, wherein the first object is executed on a device;

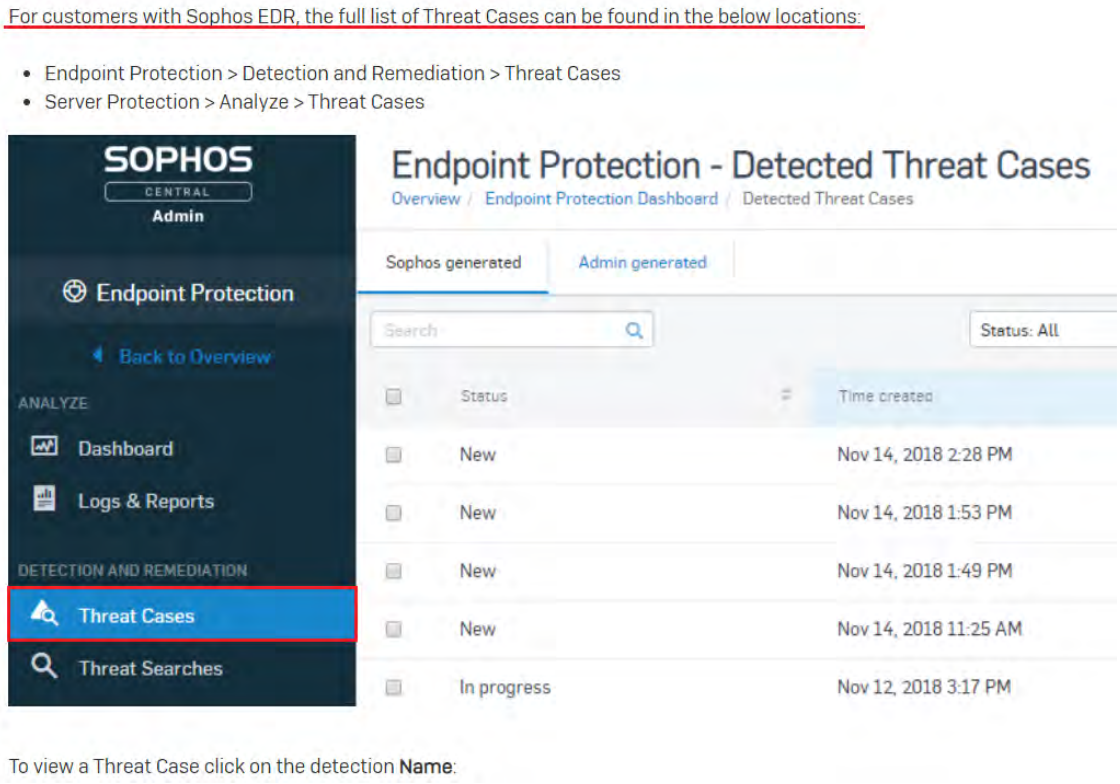
generating contextual state information for the event by correlating the event to an originating object of the first object;

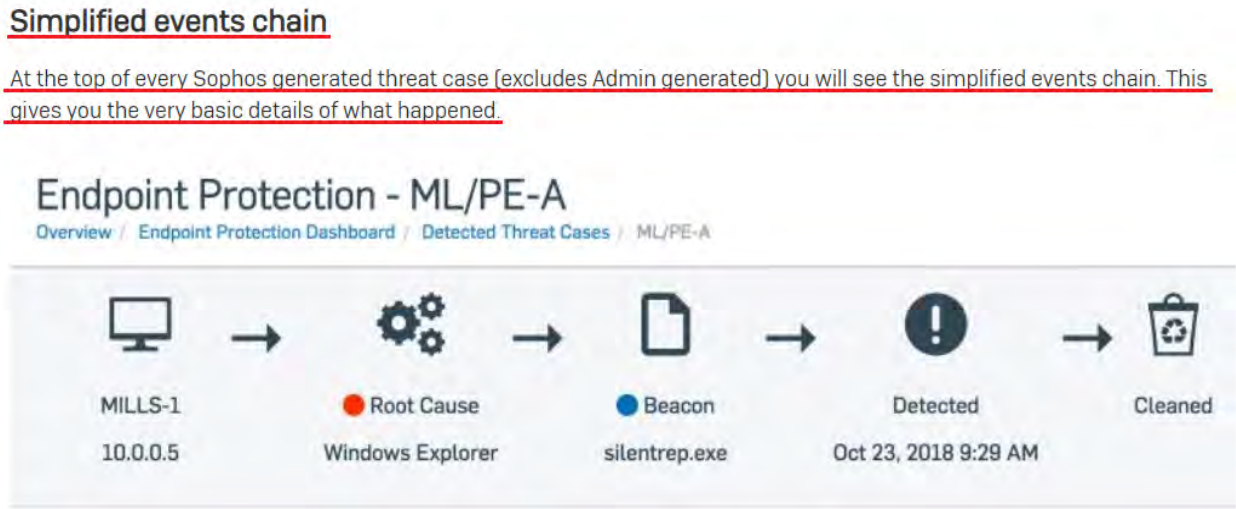
obtaining a global perspective for the event based on the contextual state information, wherein the global perspective comprises information associated with one or more of the first object and the originating object, and wherein the global perspective relates to one or more other events related to the event across a network;

generating an event line comprising information relating to the event, wherein the information relates to at least one of the first object, the action of the first object, the target, and the originating object; and

transmitting the generated event line.

214. To the extent the preamble is construed to be limiting, the Accused Products perform a *method* as further explained below. For example, the Accused Products perform a method for endpoint protection, wherein threat cases/attacks are analyzed in detail.





(*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_USl.)

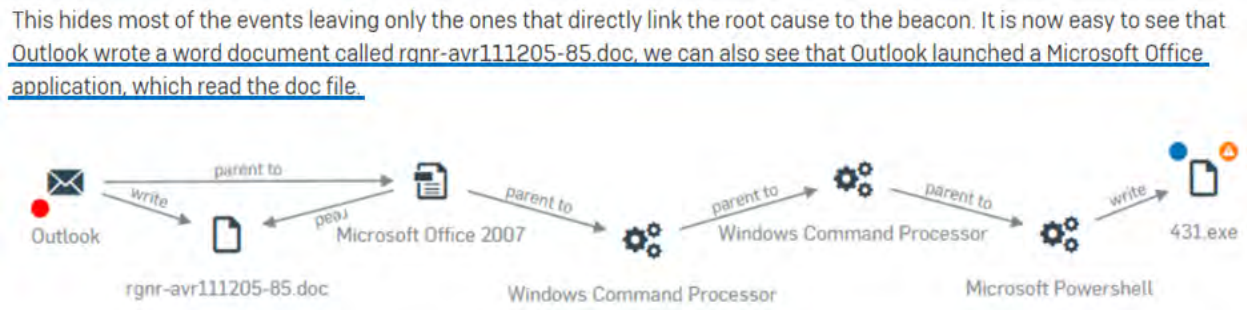
215. The Accused Products perform a method of *gathering an event defining an action of a first object acting on a target, wherein the first object is executed on a device*. In the example shown below, the "Analyze" tab of Intercept X illustrates the illicit creation of the malicious file "431.exe" by Microsoft Powershell, which is marked as the "Beacon," and the subsequent actions of "431.exe" on the infected endpoint, such as modifying registry keys. The Analyze Tab describes the "attack chain," *i.e.*, the chain of events linking the "Beacon" to what Intercept X has identified as the "Root Cause," in this case the program "Outlook." In the example below, Outlook wrote a document called "rgnr-avr11205-85.doc" and used Microsoft Office to read the document, initiating a chain of events culminating in the illicit creation and execution of the malicious file "431.exe" via Microsoft Powershell. Events defining the attack chain are gathered from the endpoint device by Intercept X (*e.g.*, Sophos Central). (*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_US.)

216. The root cause analysis performed by Intercept X, and illustrated by an attack chain, is further described in the video "Root Cause Analysis RCA in 2 minutes":

(*See* https://www.youtube.com/watch?v=AOsjUjp4P7Q (showing the root cause as the process circled in red above).)

217. The Accused Products perform a method of *generating contextual state information for the event by correlating the event to an originating object of the first object*. As explained above, Intercept X's "Analyze" tab illustrates the illicit creation and execution of the malicious file "431.exe" by Microsoft Powershell, which is marked as the "Beacon."

218. In the example below, the Analyze Tab describes the "attack chain," *i.e.*, the chain of associated events linking the "Beacon" to what Intercept X has identified as the "Root Cause," in this case the program "Outlook." Outlook wrote a document called "rgnr-avr11205-85.doc" and used Microsoft Office to read the document, initiating a chain of events culminating in the illicit creation and execution of the malicious file "431.exe" via Microsoft Powershell. The attack chain also illustrates the subsequent actions of "431.exe" on the infected endpoint, such as modifying registry keys. (*See* https://support.sophos.com/support/s/article/KB-000036359?language =en_US.)



(*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_US.)

219. The details of the attack chain, and the relationships it illustrates between the "Root Cause," the "Beacon," and the intervening files or processes between them, include the "*contextual state information for the event by correlating the event to an originating object of the first object*."

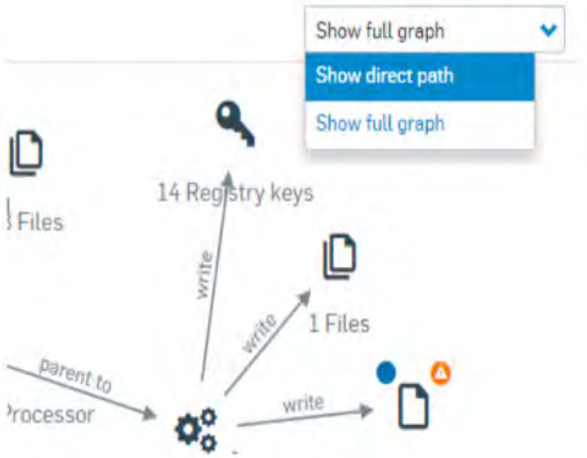
220. The Accused Products perform a method of *obtaining a global perspective for the event based on the contextual state information wherein the global perspective comprises information associated with one or more of the first object and the originating object, and wherein the global perspective relates to one or more other events related to the event across a network.* The attack chain includes information obtained about the "Root Cause," the "Beacon," events involving them, as well as the intervening files or processes from associated events across a network. For example, Intercept X obtains information at least about the age, popularity URL information of the processes within the attack chain.

Report summary

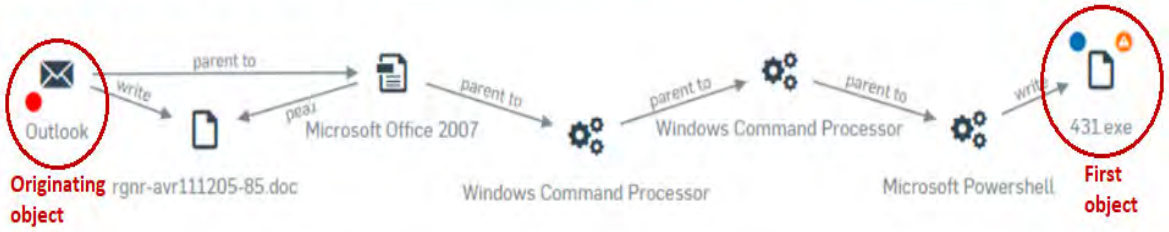
Under **Report summary**, you can see the file's reputation and prevalence and the results of our machine learning analysis, which indicate how suspicious the file is.

Setting	Description
Prevalence	Indicates how often SophosLabs has seen the file.
First seen	When SophosLabs first saw the file in the wild.
Last seen	When SophosLabs last saw the file in the wild
Machine learning analysis	Summarizes how suspicious the file is.

(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html.)



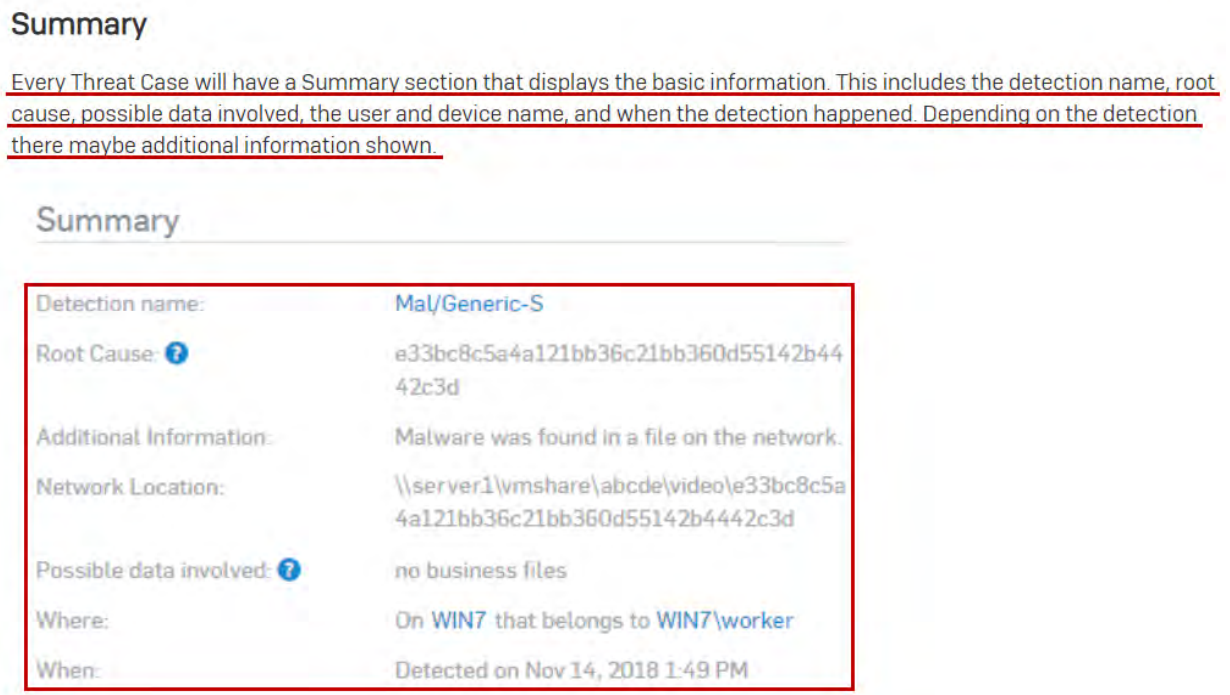
This hides most of the events leaving only the ones that directly link the root cause to the beacon. It is now easy to see that Outlook wrote a word document called rgnr-avr111205-85.doc, we can also see that Outlook launched a Microsoft Office application, which read the doc file.



We can already see that the Microsoft Office event does not have a reputation icon, which means it has a good reputation.

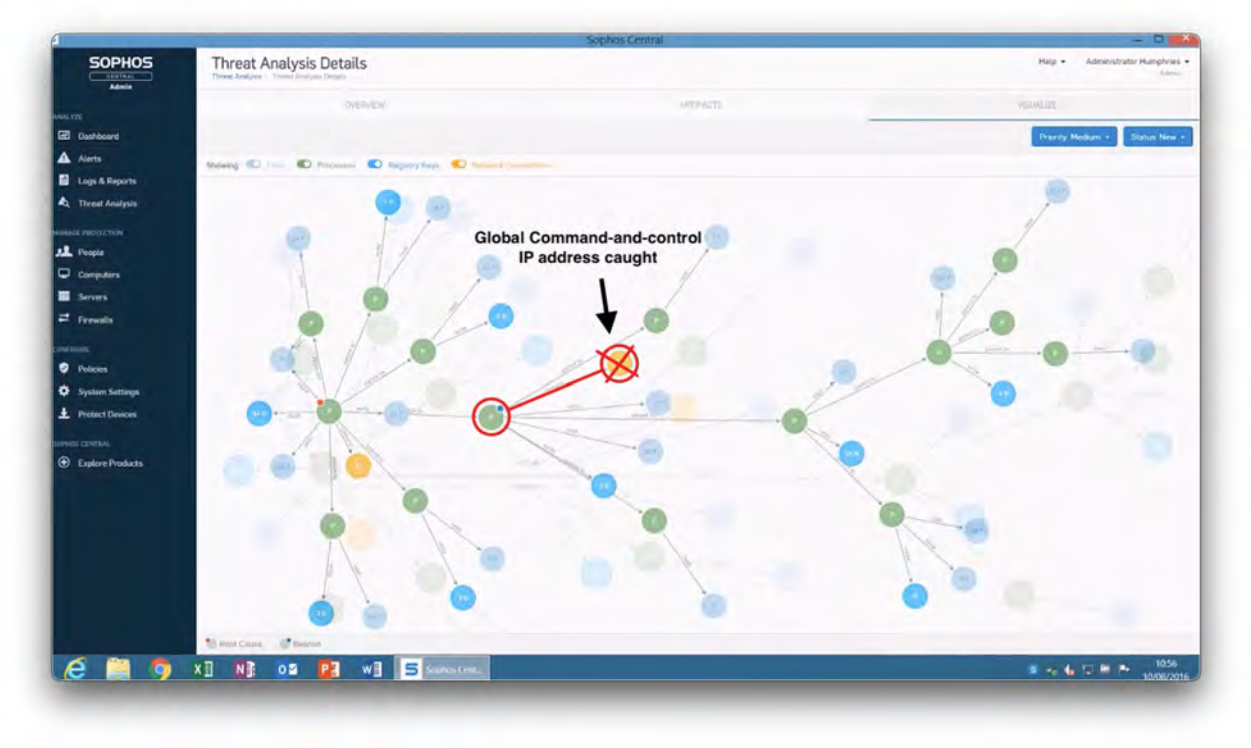
Note: Reputation is only calculated for Portable Executable (PE) files, for example .exe, .dll. It is not shown for other file types such as .doc, .pdf, .png.

(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html.)



(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html; *see also* https://www.youtube.com/watch?v=_ujOT58ZvpI.)

221. Moreover, the Accused Products identify, for example, global IP addresses that are malware command-and-control centers: "Here's the beacon event, the thing that was caught, in this case a fake salary report. Here's the root, Google Chrome. We see here that the fake report reached out to an IP address that we happen to know to be a command-and-control site. This right here is the moment of conviction, when we discovered that what was executing was a piece of malware."

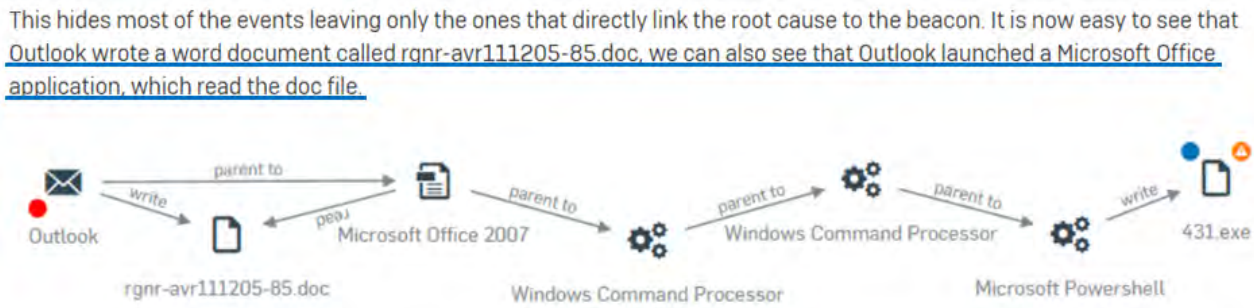


(*See* https://youtu.be/AOsjUjp4P7Q?t=48.)

222. The Accused Products perform a method of *generating an event line comprising information relating to the event, wherein the information relates to at least one of the first object, the action of the first object, the target, and the originating object*. As explained above, an attack chain created by Intercept X includes information associated with, and identifying the "Root Cause," the "Beacon," and the intervening files or processes in the attack chain. In the example included, Intercept X generates an attack chain illustrating the illicit creation, execution, and subsequent actions of the malicious file "431.exe."

223. In the example below, the "attack chain" links the "Beacon" to what Intercept X has identified as the "Root Cause," in this case the program "Outlook." Outlook wrote a document called "rgnr-avr11205-85.doc" and used Microsoft Office to read the document, initiating a chain of events culminating in the illicit creation and execution of the malicious file "431.exe" via Microsoft Powershell. The attack chain also illustrates the subsequent actions of "431.exe" on the

infected endpoint, such as modifying registry keys. (*See* https://support.sophos.com/support/s/article/KB-000036359?language=en_US.)



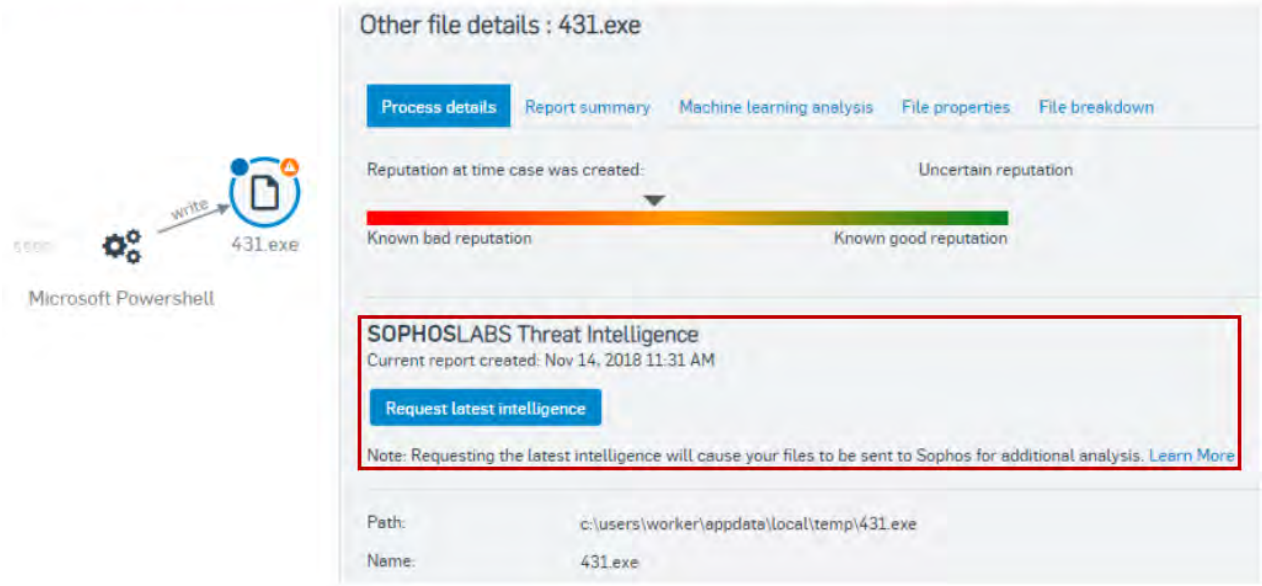
(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html.)

224. The Accused Products perform a method of *transmitting the generated event line*. In the example below, and as explained above, Intercept X's "Analyze" tab illustrates the "attack chain" linking the illicit creation, execution, and subsequent actions of the malicious file "431.exe" by Microsoft Powershell, *i.e.*, "the Beacon," to the "Root Cause." Intercept X thus transmits the event line such that the attack chain can be generated, stored or displayed.

225. Furthermore, Intercept X transmits the event line to SophosLABS for "additional analysis."

Powershell has then written the 431.exe file which was detected by Sophos Deep Learning as ML/PE-A.

By selecting the beacon event we can see confirmation that it has an uncertain reputation, as well as that it was written to the users AppData location, this location is typically meant for data not executable's so this is also suspicious.



(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html.)

For customers using Sophos EDR, by pressing the **Request latest intelligence** button, the file will be retrieved out of the Sophos quarantine and submitted to SophosLabs. A couple of minutes later the four other tabs (Report summary, Machine learning analysis, File properties, File breakdown) pictured will be displayed. The purpose of these these additional tabs is to help display the various properties of the file in a simple way. This can be useful for various reasons, one of them is to feel confident that the file is indeed malicious and not something you want in your environment. For more information on SophosLabs Threat Intelligence, please see: Sophos Central: Threat intelligence overview.

Now that we understand that the file was malicious and that it came from an email, which then used Microsoft Word, CMD and Powershell to execute the attack chain, we can decide what could be done to help prevent this type of attack happening again.

(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html.)

226. Each claim in the ’224 Patent recites an independent invention. Neither claim 1, described above, nor any other individual claim is representative of all claims in the ’224 Patent.

227. Defendant has been aware of the ’224 Patent since at least the filing of this Complaint. Further, Plaintiffs have marked its products with the ’224 Patent, including on its web

site, since at least July 2020.

228. Defendant directly infringes at least claim 1 of the ’224 Patent, literally or under the doctrine of equivalents, by performing the steps described above. For example, on information and belief, Defendant performs the claimed method as described above by running this software and corresponding systems to protect its own computer and network operations. On information and belief, Defendant also performs the claimed method as described above when testing the operation of the Accused Products and corresponding systems. As another example, Defendant performs the claimed method as described above when providing or administering services to third parties, customers, and partners using the Accused Products.

229. Defendant’s partners, customers, and end users of the Accused Products and corresponding systems and services directly infringe at least claim 1 of the ’224 Patent, literally or under the doctrine of equivalents, at least by using the Accused Products and corresponding systems and services, as described above.

230. Defendant has actively induced and is actively inducing infringement of at least claim 1 of the ’224 Patent with specific intent to induce infringement, and/or willful blindness to the possibility that its acts induce infringement, in violation of 35 U.S.C. § 271(b). For example, Defendant encourages and induces customers with specific intent to use Sophos’ security software in a manner that infringes claim 1 of the ’224 Patent at least by offering and providing software that performs a method that infringes claim 1 when installed and operated by the customer, and by engaging in activities relating to selling, marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including the activities described below.

231. Defendant encourages, instructs, directs, and/or requires third parties—including its certified partners and/or customers—to perform the claimed method using the software,

services, and systems in infringing ways, as described above.

232. Defendant further encourages and induces its customers to infringe claim 1 of the '224 Patent: 1) by making its security services available on its website, providing applications that allow users to access those services, widely advertising those services, and providing technical support and instructions to users, and 2) through activities relating to marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including its Sophos security software, and services in the United States. (*See* https://www.sophos.com/en-us/products/endpoint-antivirus/how-to-buy.aspx; https://partners.sophos.com/english/directory/search?lat=30.267153&lng=-97.7430608&dMI=100&p=1; https://secure2.sophos.com/en-us/security-news-trends/whitepapers/gated-wp/cybersecurity-system-buyers-guide.aspx%23formFrame.)

233. For example, on information and belief, Defendant shares instructions, guides, and manuals, which advertise and instruct third parties on how to use the software as described above, including at least customers and partners. (*Id.*) On further information and belief, Defendant also provides customer service and technical support to purchasers of the Accused Products and corresponding system and services, which directs and encourages customers to perform certain actions that use the Accused Products in an infringing manner. (*Id.*)

234. Defendant and/or its partners recommend and sell the Accused Products and provide technical support for the installation, implementation, integration, and ongoing operation of the Accused Products for each individual customer. On information and belief, each customer enters into a contractual relationship with Defendant and/or one of its partners, which obligates each customer to perform certain actions in order to use of the Accused Products. Further, in order to receive the benefit of Defendant and/or its partner's continued technical support and their

specialized knowledge and guidance of the operability of the Accused Products, each customer must continue to use the Accused Products in a way that performs the claimed method and infringes the ’224 Patent.

235. Further, as the entity that provides installation, implementation, and integration of the Accused Products in addition to ensuring the Accused Product remains operational for each customer through ongoing technical support, on information and belief, Defendant and/or its partners affirmatively aid and abet each customer’s use of the Accused Products in a manner that performs the claimed method of, and infringes, the ’224 Patent.

236. Defendant also contributes to the infringement of its partners, customers, and end-users of the Accused Products by providing within the United States or importing the Accused Products into the United States, which are for use in practicing, and under normal operation practice, methods claimed in the Asserted Patents, constituting a material part of the claimed methods, and not a staple article or commodity of commerce suitable for substantial non-infringing uses. Indeed, as shown above, the Accused Products and the example functionality described below have no substantial non-infringing uses but are specifically designed to practice the ’224 Patent.

237. Indeed, as shown above, the Accused Products have no substantial non-infringing uses because the accused functionality, including the “Threat Anlaysis,” “Root Cause Analysis,” and related functionality described above, is an integral part of the Accused Products and must be performed for the Accused Products to perform their intended purpose. For example, without analyzing data about the behavior and globally known information about the maliciousness of an object running on one or more remote computers, the Accused Products could not detect when that object had engaged in an illicit behavior, and could not create a forensic trail associating that object

with its root cause. These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer suitably function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice the '224 patent, that functionality could not be performed.

238. Additionally, the accused functionality, including the "Threat Anlaysis," "Root Cause Analysis," and related functionality, itself has no substantial non-infringing uses because the components, modules and methods identified above are a necessary part of that functionality. For example, without creating an "attack chain" that compiles and correlates behavioral data from processes that run on the endpoints protected by the Accused Products, the Accused Products could not describe how and why the illicit behaviors detected on any given endpoint constitute a threat. These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice the '224 patent, that functionality could not be performed.

239. In addition, as shown in the detailed analysis above, the products, systems, modules, and methods provided by Defendant constitute a material part of the invention—indeed, they provide all the components, modules, and features that perform the claimed methods and systems. For example, the Accused Products and accused functionalities (including the "Threat Anlaysis"/"Root Cause Analysis") constitute a material part of the inventions claimed at least because they are integral to the processes identified above (such as "generating contextual state information . . ."; "obtaining a global perspective . . ."; "generating" and "transmitting" an "event

line")), as recited in the claims of the '224 patent. None of these products are staple goods—they are sophisticated and customized cyber security and malware detection products, methods, and systems.

240. Defendant's infringing actions have continued after the original Complaint was filed. Defendant had knowledge of the '224 patent, and of the specific conduct that constitutes infringement of the '224 patent at least based on the original Complaint, but has continued to engage in the infringing activity, including by selling, making, configuring, and installing the Accused Products and performing the accused functionality, and by engaging in activities that constitute inducing infringement and contributory infringement as described above.

241. On information and belief, the infringing actions of each partner, customer, and/or end-user of the Accused Products are attributable to Defendant. For example, on information and belief, Defendant directs and controls the activities or actions of its partners or others in connection with the Accused Products by contractual agreement or otherwise requiring partners or others to provide information and instructions to customers who acquire the Accused Products which, when followed, results in infringement. Defendant further directs and controls the operation of devices executing the Accused Products by programming the software which, when executed by a customer or end user, perform the method steps of at least claim 1 of the '224 Patent.

242. Plaintiffs have suffered and continue to suffer damages, including lost profits, as a result of Defendant's infringement of the '224 Patent. Defendant is therefore liable to Plaintiffs under 35 U.S.C. § 284 for damages in an amount that adequately compensates Plaintiffs for Defendant's infringement, but no less than a reasonable royalty.

243. Plaintiffs will continue to suffer irreparable harm unless this Court preliminarily and permanently enjoins Defendant, its agents, employees, representatives, and all others acting

in concert with Defendant from infringing the ’224 Patent.

244. Defendant’s infringement of the ’224 Patent is knowing and willful. Defendant had actual knowledge of the ’224 Patent at least by the time Plaintiffs filed this lawsuit and had constructive knowledge of the ’224 Patent from at least the date Plaintiffs marked their products with the ’224 Patent and/or provided notice of the ’224 Patent on their website.

245. On information and belief, despite Defendant’s knowledge of the Asserted Patents and Plaintiffs’ patented technology, Defendant made the deliberate decision to sell products and services that they knew infringe the ’224 Patent. Defendant’s continued infringement of the ’224 Patent with knowledge of the ’224 Patent constitutes willful infringement.

246. Plaintiffs’ allegations of infringement, indirect infringement, and willful infringement with respect to this patent are further set forth in Plaintiffs’ Disclosure of Asserted Claims and Preliminary Infringement Contentions Pursuant to the Court’s Standing Order Governing Patent Proceedings served July 12, 2022.

FIFTH CAUSE OF ACTION
(INFRINGEMENT OF THE ’591 PATENT)

247. Plaintiffs reallege and incorporate the preceding paragraphs of this complaint.

248. Defendant has infringed and continues to infringe one or more claims of the ’591 Patent in violation of 35 U.S.C. § 271 in this District and elsewhere in the United States and will continue to do so unless enjoined by this Court. The Accused Products, including anti-exploit features such as those included in Intercept X, at least when used for their ordinary and customary purposes, practice each element of at least claim 1 of the ’591 Patent as described below.

249. For example, claim 1 of the ’591 Patent recites:

> 1. A computer-implemented method comprising:
>
> monitoring a memory space of a process for execution of at least one monitored function of a plurality of functions, wherein monitoring the memory

space comprises loading a component for evaluating the at least one monitored function in the memory space;

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; and

performing, during the executing of the stack walk processing before an address of an originating caller function is reached, a memory check for a plurality of stack entries identified during the stack walk processing to detect suspicious behavior, wherein an alert of suspicious behavior is triggered when the performing of the memory check detects at least one of the following:

code execution is attempted from non-executable memory,

a base pointer is identified as being invalid,

an invalid stack return address is identified,

attempted execution of a return-oriented programming technique is detected,

the base pointer is detected as being outside a current thread stack, and

a return address is detected as being inside a virtual memory area,

wherein when an alert of suspicious behavior is triggered, preventing execution of a payload for the invoked function from operating.

250. The Accused Products perform each of the method steps of claim 1 of the ’591 Patent. To the extent the preamble is construed to be limiting, the Accused Products perform *a computer-implemented method*, as further explained below. For example, the Accused Products include “[e]xploit prevention [to] stop[] the techniques used in file-less, malware-less, and exploit-based attacks.” (*See* https://www.sophos.com/en-us/products/endpoint-antivirus.aspx.)

251. The Accused Products perform a method that includes *monitoring a memory space of a process for execution of at least one monitored function of a plurality of functions, wherein monitoring the memory space comprises loading a component for evaluating the at least one*

monitored function in the memory space. For example, the Accused Products load a component for monitoring memory space when monitoring "sensitive API functions." In another example, the Accused Products include "Memory Scanning" for "defense against in-memory malware" and monitor "API call[s] (*e.g.*, VitrualAlloc)."

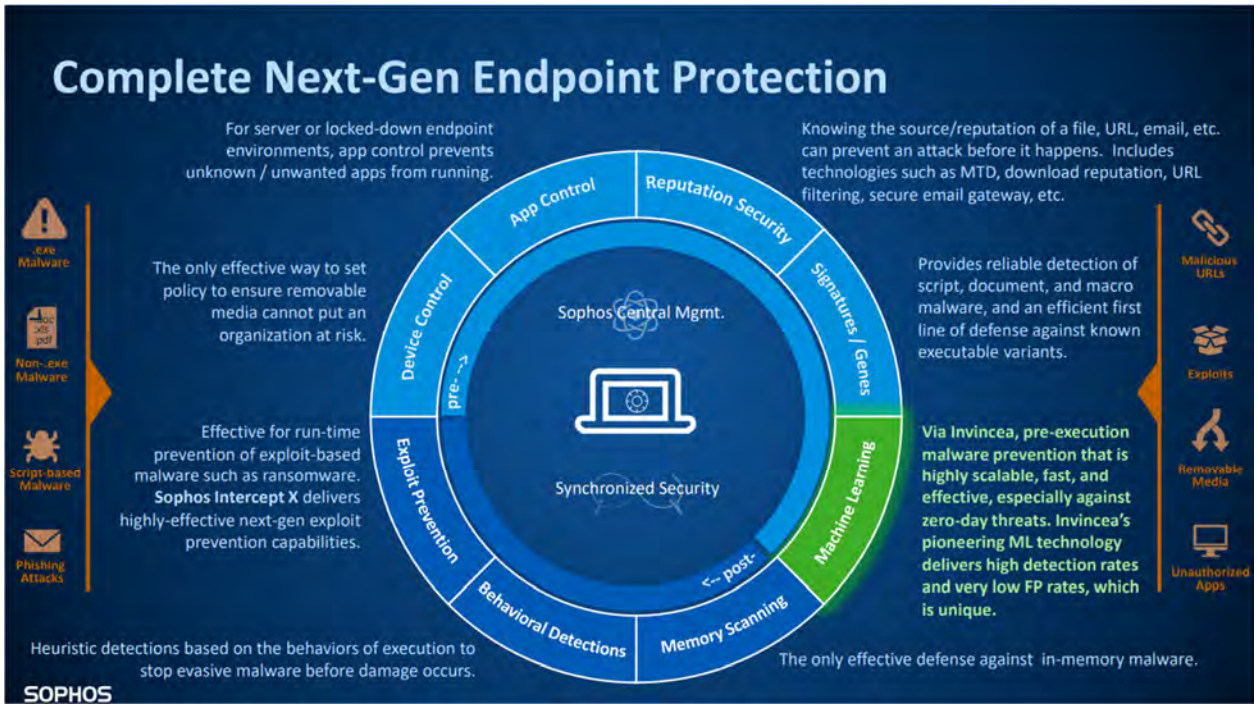
Stack-based ROP Mitigation (Caller)

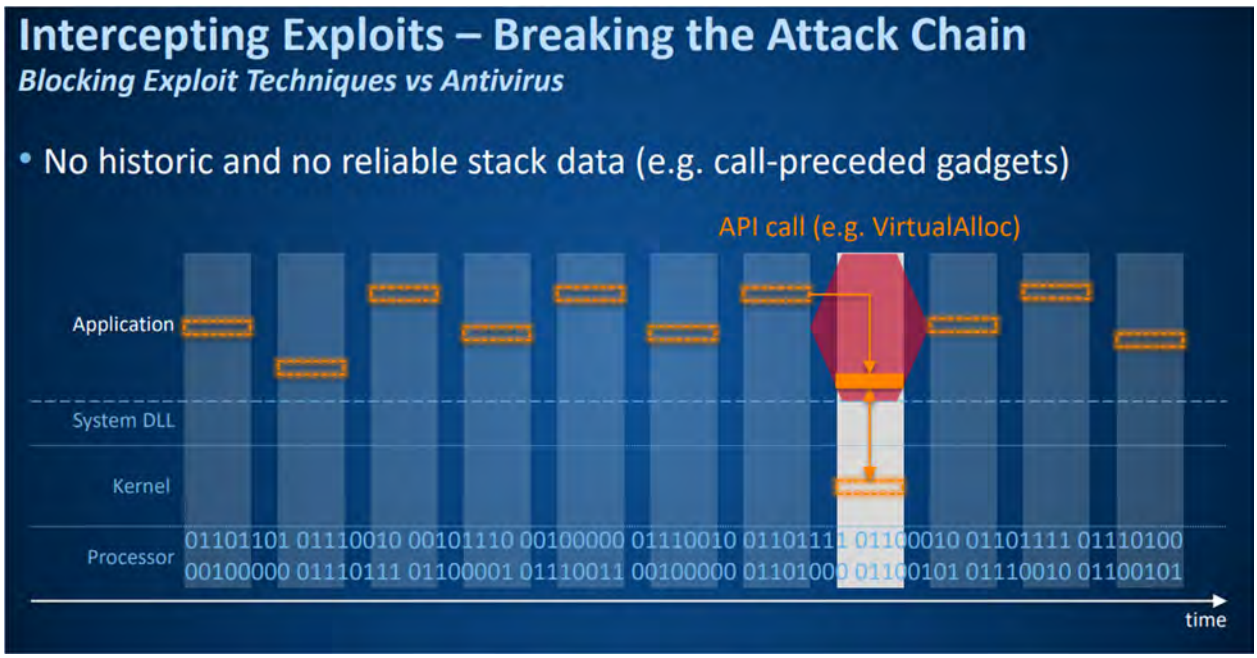
To defeat security technologies like data execution prevention (DEP) and address space layout randomization (ASLR), attackers typically resort to hijacking control-flow of vulnerable internet-facing applications. Such in-memory attacks are invisible to antivirus, most "next-gen" products, and other cyber defenses as there are no malicious files involved. Instead, the attack is constructed at run time by combining short pieces of benign code that are part of existing applications like Internet Explorer and Adobe Flash Player – a so-called code-reuse or return-oriented programming (ROP) attack.

During normal control-flow, sensitive API functions – like VirtualAlloc and CreateProcess – are invoked by the CALL instruction. Upon invoking a sensitive API, typical ROP defenses stop code execution to determine the API invoking address, using the 'return' address which is located on top of the stack. If the instruction of the API invoking address is not a CALL, the process is terminated.

Since the contents of the stack are writable, an attacker can write specific values on the stack to mislead the analysis of the stack-based ROP defense. The stack-based ROP defense cannot determine if the contents of the stack are benign or manipulated by an attacker.

(*See* https://www.sophos.com/en-us/medialibrary/Gated-Assets/white-papers/Sophos-Comprehensive-Exploit-Prevention-wpna.pdf.)





(*See* https://secure2.sophos.com/it-it/medialibrary/PDFs/other/end-of-ransomware/MarkLomanSophosInterceptX.ashx.)

252. The Accused Products perform a method that includes *invoking one of the plurality of functions as a result of receiving a call from an application programming instance*. For example,

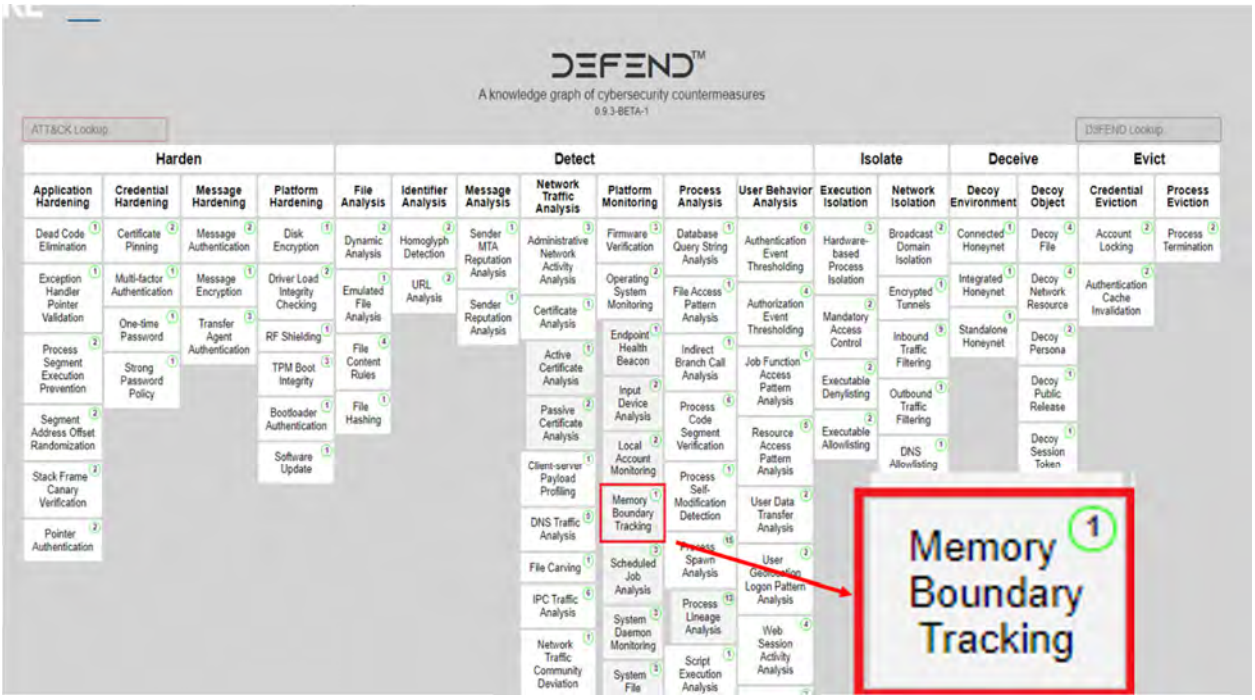
the Accused Products monitor "sensitive API functions—like VirtualAlloc and CreateProcess…invoked by the CALL instruction."

> During normal control-flow, sensitive API functions – like VirtualAlloc and CreateProcess – are invoked by the CALL instruction. Upon invoking a sensitive API, typical ROP defenses stop code execution to determine the API invoking address, using the 'return' address which is located on top of the stack. If the instruction of the API invoking address is not a CALL, the process is terminated.

(*See* https://www.sophos.com/en-us/medialibrary/Gated-Assets/white-papers/Sophos-Comprehensive-Exploit-Prevention-wpna.pdf.)

253. On information and belief, the Accused Products perform a method that includes *executing stack walk processing upon the invocation of one of the plurality of functions in the monitored memory space*. For example, as shown above, the Accused Products evaluate and trace a stack "[u]pon invoking a sensitive API…to determine the API invoking address, using the 'return' address which is located on top of the stack." In another example, the Accused Products include the "CallerCheck" anti-exploit module for "[p]revent[ing] API invocation from stack memory." (*See* https://news.sophos.com/en-us/2021/03/04/covert-code-faces-a-heap-of-trouble-in-memory/.)

254. In another example, the Accused Products "[l]ist detected IoCs mapped to the MITRE ATT&CK [Adversarial Tactics, Techniques and Common Knowledge] framework." Furthermore, the MITRE ATT&CK framework includes companion project D3FEND for defensive cybersecurity techniques, which includes "Memory Boundary Tracking" defined as "[a]nalyzing a call stack for return addresses which point to unexpected memory locations." On information and belief, the Accused Products incorporate the MITRE D3FEND defensive cybersecurity techniques including "Memory Boundary Tracking."



Memory Boundary Tracking

ID: D3-MBT (Memory Boundary Tracking)

Definition

Analyzing a call stack for return addresses which point to unexpected memory locations.

How it works

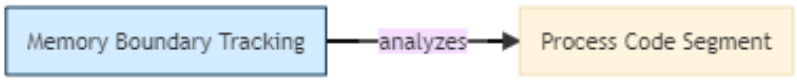
This technique monitors for indicators of whether a return address is outside memory previously allocated for an object (i.e. function, module, process, or thread). If so, code that the return address points to is treated as malicious code.

Considerations

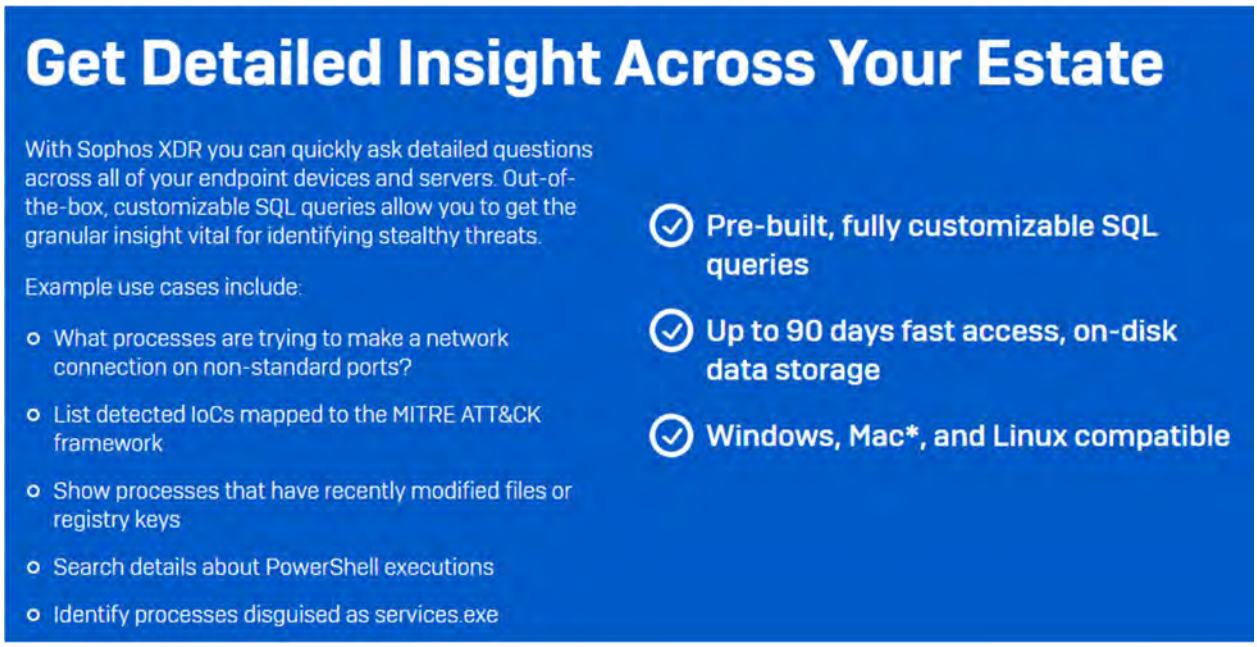
Kernel malware can manipulate memory contents, for example modifying pointers to hide processes, and thereby impact the accuracy of memory allocation information used to perform the analysis.

Digital Artifact Relationships:

This countermeasure technique is related to specific digital artifacts. Click the artifact node for more information.

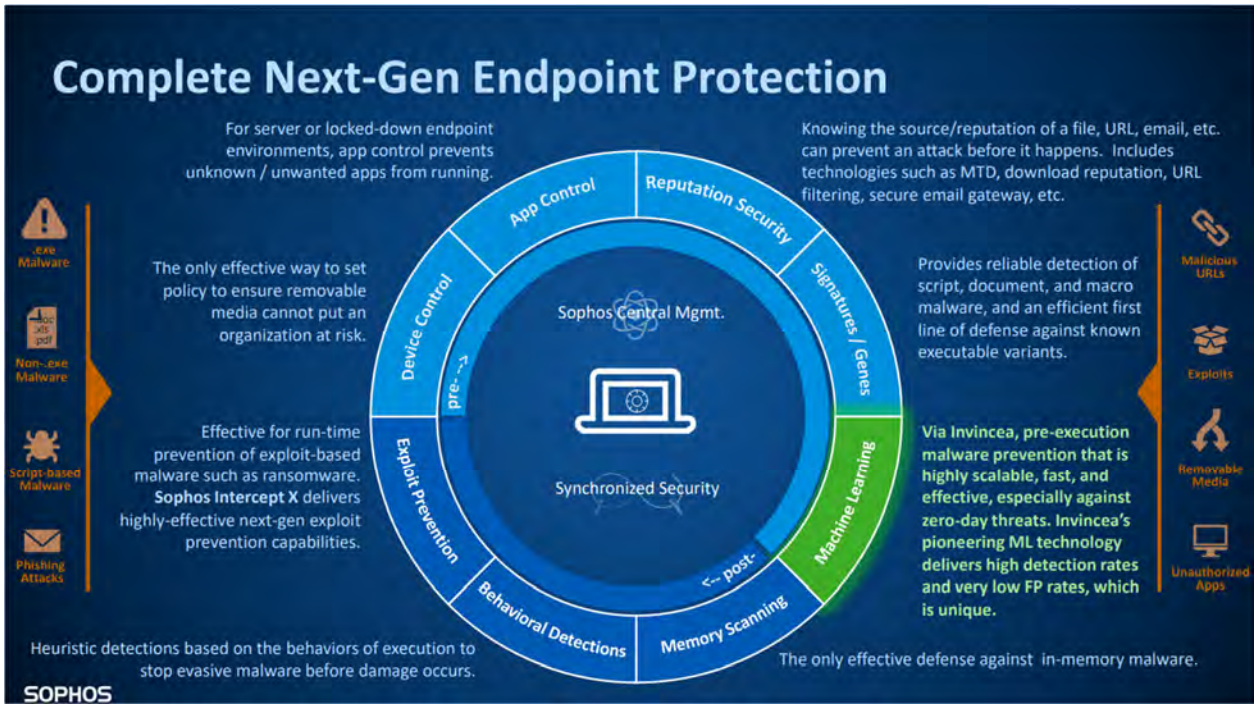


(*See* https://d3fend.mitre.org/technique/d3f:MemoryBoundaryTracking; *see also* https://www.csoonline.com/article/3625470/mitre-d3fend-explained-a-new-knowledge-graph-for-cybersecurity-defenders.html; https://d3fend.mitre.org/resources/D3FEND.pdf.)



(*See* https://www.sophos.com/en-us/content/threat-hunting.aspx.)

255. On information and belief, the Accused Products perform a method that includes *performing, during the executing of the stack walk processing before an address of an originating caller function is reached, a memory check for a plurality of stack entries identified during the stack walk processing to detect suspicious behavior*. For example, the Accused Products include "Memory Scanning," Behavioral Detections," and "Exploit Prevention…[e]ffective for run-time prevention of exploit-based malware."



(*See* https://secure2.sophos.com/it-it/medialibrary/PDFs/other/end-of-ransomware/MarkLomanSophosInterceptX.ashx.)

256. As shown above, the Accused Products include the "CallerCheck" anti-exploit module for "[p]revent[ing] API invocation from stack memory." In addition, as shown above, the Accused Products utilize the threat-based MITRE ATT&CK framework, and on information and belief, utilize companion project D3FEND for defensive cybersecurity techniques including "Memory Boundary Tracking" defined as "[a]nalyzing a call stack for return addresses which point to unexpected memory locations." (*See* https://news.sophos.com/en-us/2021/03/04/covert-code-faces-a-heap-of-trouble-in-memory/.)

Memory Boundary Tracking

ID: D3-MBT (Memory Boundary Tracking)

Definition

Analyzing a call stack for return addresses which point to unexpected memory locations.

How it works

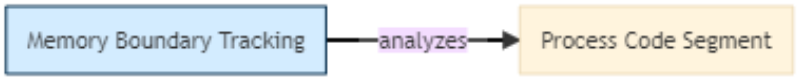
This technique monitors for indicators of whether a return address is outside memory previously allocated for an object (i.e. function, module, process, or thread). If so, code that the return address points to is treated as malicious code.

Considerations

Kernel malware can manipulate memory contents, for example modifying pointers to hide processes, and thereby impact the accuracy of memory allocation information used to perform the analysis.

Digital Artifact Relationships:

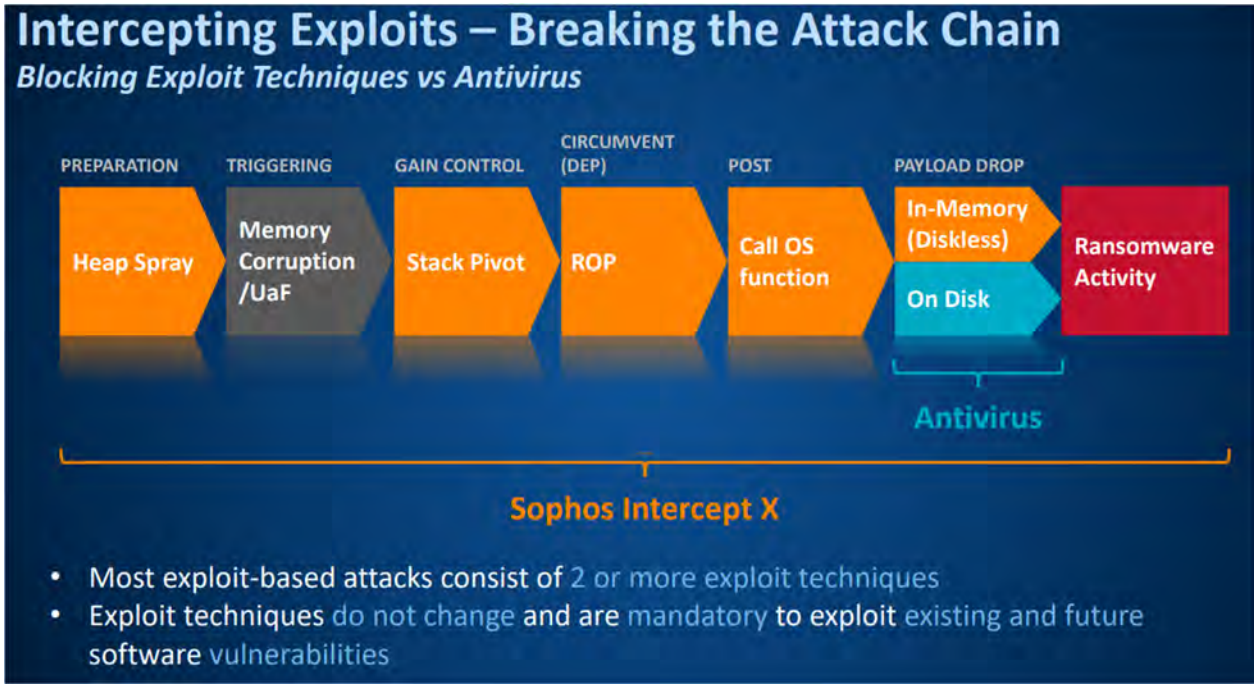
This countermeasure technique is related to specific digital artifacts. Click the artifact node for more information.



(*See* https://d3fend.mitre.org/technique/d3f:MemoryBoundaryTracking; *see also* https://www.csoonline.com/article/3625470/mitre-d3fend-explained-a-new-knowledge-graph-for-cybersecurity-defenders.html; https://d3fend.mitre.org/resources/D3FEND.pdf); https://www.sophos.com/en-us/medialibrary/Gated-Assets/white-papers/Sophos-Comprehensive-Exploit-Prevention-wpna.pdf.)

257. On information and belief, the Accused Products perform a method that includes *wherein an alert of suspicious behavior is triggered when the performing of the memory check detects at least one of the following: code execution is attempted from non-executable memory, a base pointer is identified as being invalid, an invalid stack return address is identified, attempted execution of a return-oriented programming technique is detected, the base pointer is detected as*

being outside a current thread stack, and a return address is detected as being inside a virtual memory area. For example, the Accused Products detect and prevent malware memory exploitations including "Stack Pivot," "Stack Exec," "Stack-based ROP [return-oriented programming] Mitigations," and "Shellcode."



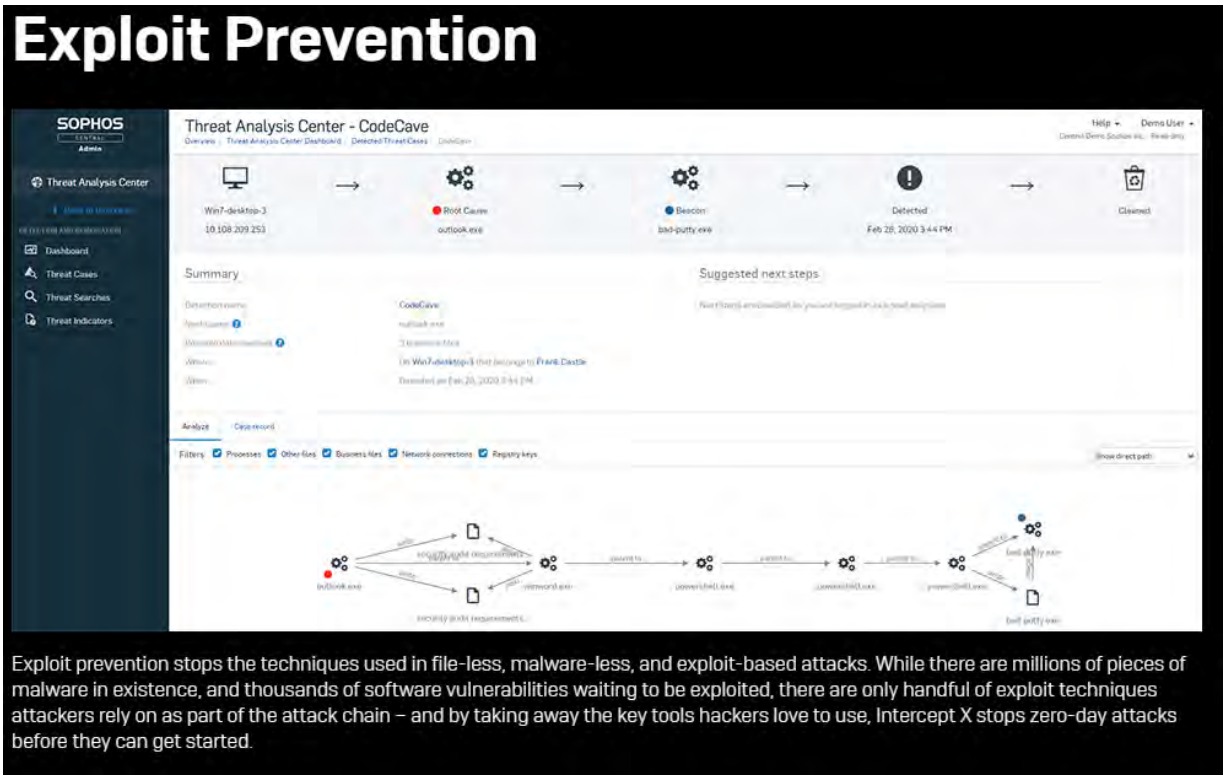
(*See* https://secure2.sophos.com/it-it/medialibrary/PDFs/other/end-of-ransomware/MarkLomanSophosInterceptX.ashx.)

258. In another example, the Accused Products prevent malware attacks using stack memory including "DEP," "ROP," "CallerCheck," "StackPivot," "StackExec," and "AmsiGuard."

Description	Module	Level	Equivalent in Windows 10
Enforce Data Execution Prevention (DEP)	DEP	Application	Yes
Mandatory ASLR on modules	DEP (ASLR)	Application	Yes
Bottom-up ASLR	DEP (ASLR)	Application	Yes
Validate exception chains	SEHOP	Application	Yes
Validate API invocation	ROP	Application	Optional
Prevent API invocation from stack memory	CallerCheck	Application	–
Prevent process creation from dynamic memory	CallerCheck	Application	–
Import address filtering (IAF)	IAF	Application	Optional
Validate stack integrity	StackPivot	Application	Optional
Validate stack memory protection	StackExec	Application	Optional
Prevent in-memory manipulation of AMSI.DLL	AmsiGuard	System	

(*See* https://news.sophos.com/en-us/2021/03/04/covert-code-faces-a-heap-of-trouble-in-memory/.)

259. The Accused Products perform a method that includes *wherein when an alert of suspicious behavior is triggered, preventing execution of a payload for the invoked function from operating*. For example, the Accused Products "stop[] the techniques used in file-less, malware-less, and exploit-based attacks."



(*See* https://www.sophos.com/en-us/products/endpoint-antivirus.aspx.)

260. In another example, the Accused Products are demonstrated preventing shellcode execution by utilizing non-executable memory and using "Data Execution Prevention…to prevent abuse of a buffer overflow."

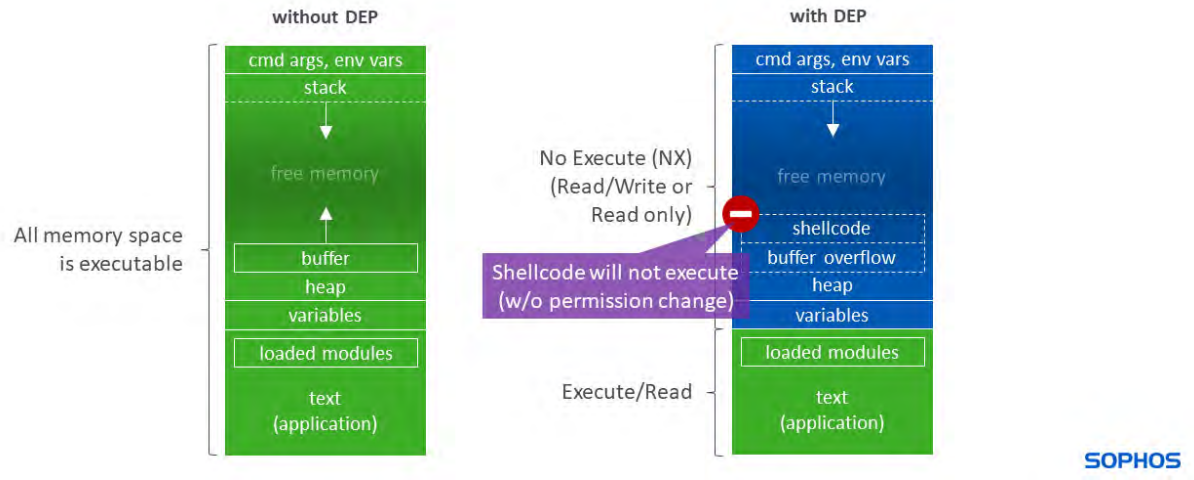


Figure 5 – Data Execution Prevention helps to prevent abuse of a buffer overflow

(*See* https://news.sophos.com/en-us/2021/03/04/covert-code-faces-a-heap-of-trouble-in-memory/.)

261. Each claim in the ’591 Patent recites an independent invention. Neither claim 1, described above, nor any other individual claim is representative of all claims in the ’591 Patent.

262. Defendant has been aware of the ’591 Patent since at least the filing of this Complaint. Further, Plaintiffs have marked its products with the ’591 Patent, including on its web site, since at least July 2020.

263. Defendant directly infringes at least claim 1 of the ’591 Patent, literally or under the doctrine of equivalents, by performing the steps described above. For example, on information and belief, Defendant performs the claimed method as described above by running this software and corresponding systems to protect its own computer and network operations. On information and belief, Defendant also performs the claimed method as described above when testing the operation of the Accused Products and corresponding systems. As another example, Defendant performs the claimed method as described above when providing or administering services to third parties, customers, and partners using the Accused Products.

264. Defendant’s partners, customers, and end users of the Accused Products and corresponding systems and services directly infringe at least claim 1 of the ’591 Patent, literally or under the doctrine of equivalents, at least by using the Accused Products and corresponding systems and services, as described above.

265. Defendant has actively induced and is actively inducing infringement of at least claim 1 of the ’591 Patent with specific intent to induce infringement, and/or willful blindness to the possibility that its acts induce infringement, in violation of 35 U.S.C. § 271(b). For example, Defendant encourages and induces customers with specific intent to use Sophos’s security

software in a manner that infringes claim 1 of the ’591 Patent at least by offering and providing software that performs a method that infringes claim 1 when installed and operated by the customer, and by engaging in activities relating to selling, marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including the activities described below.

266. Defendant encourages, instructs, directs, and/or requires third parties—including its certified partners and/or customers to perform the claimed method using the software, services, and systems in infringing ways, as described above.

267. Defendant further encourages and induces its customers to infringe claim 1 of the ’591 Patent: 1) by making its security services available on its website, providing applications that allow users to access those services, widely advertising those services, and providing technical support and instructions to users, and 2) through activities relating to marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including its Sophos security software, and services in the United States. (*See* https://www.sophos.com/en-us/products/endpoint-antivirus/how-to-buy.aspx; https://partners.sophos.com/english/directory/search?lat=30.267153&lng=-97.7430608&dMI=100&p=1; https://secure2.sophos.com/en-us/security-news-trends/whitepapers/gated-wp/cybersecurity-system-buyers-uide.aspx%23formFrame).

268. For example, on information and belief, Defendant shares instructions, guides, and manuals, which advertise and instruct third parties on how to use the software as described above, including with at least customers and partners. (*Id.*) On further information and belief, Defendant also provides customer service and technical support to purchasers of the Accused Products and corresponding system and services, which directs and encourages customers to perform certain

actions that use the Accused Products in an infringing manner. (*Id.*)

269. Defendant and/or its partners recommend and sell the Accused Products and provide technical support for the installation, implementation, integration, and ongoing operation of the Accused Products for each individual customer. On information and belief, each customer enters into a contractual relationship with Defendant and/or one of its partners, which obligates each customer to perform certain actions in order to use the Accused Products. Further, in order to receive the benefit of Defendant's and/or its partner's continued technical support and their specialized knowledge and guidance of the operability of the Accused Products, each customer must continue to use the Accused Products in a way that performs the claimed method and infringes the '591 Patent.

270. Further, as the entity that provides installation, implementation, and integration of the Accused Products in addition to ensuring the Accused Product remains operational for each customer through ongoing technical support, on information and belief, Defendant and/or its partners affirmatively aid and abet each customer's use of the Accused Products in a manner that performs the claimed method of, and infringes, the '591 Patent.

271. Defendant also contributes to the infringement of its partners, customers, and end-users of the Accused Products by providing within the United States or importing the Accused Products into the United States, which are for use in practicing, and under normal operation practice, methods claimed in the Asserted Patents, constituting a material part of the claimed methods, and not a staple article or commodity of commerce suitable for substantial non-infringing uses. Indeed, as shown above, the Accused Products and the example functionality described below have no substantial non-infringing uses but are specifically designed to practice the '591 Patent.

272. Indeed, as shown above, the Accused Products have no substantial non-infringing uses because the accused functionality, including the anti-exploit , memory analyzing techniques, and related functionality described above, is an integral part of the Accused Products and must be performed for the Accused Products to perform their intended purpose. For example, without scanning the memory within which a malicious object executes and ensuring that it does not exceed its permissible memory boundary, the Accused Products could not detect when that object had attempted to engage in an illicit behavior, and thus could not prevent that behavior. These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer suitably function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice the ’591 patent, that functionality could not be performed.

273. Additionally, the accused functionality, including the anti-exploit, memory analyzing techniques, and related functionality described above, itself has no substantial non-infringing uses because the components, modules and methods identified above are a necessary part of that functionality. For example, without scanning the memory in which an object executes and preventing the object from exceeding its memory boundary, the Accused Products could not assess when the object was about to engage in prohibited behavior and prevent it from doing so. These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice the ’591 patent, that functionality could not be performed.

274. In addition, as shown in the detailed analysis above, the products, systems, modules, and methods provided by Defendant constitute a material part of the invention—indeed, they provide all the components, modules, and features that perform the claimed methods and systems. For example, the Accused Products and accused functionalities (including the anti-exploit, memory analyzing techniques) constitute a material part of the inventions claimed at least because such functionality is integral to the processes identified above (such as "monitoring a memory space of a process for execution of at least one monitored function . . ."; "invoking one of the plurality of functions . . ."; "executing stack walk processing . . ."; "performing, during the executing of the stack walk processing before an address of an originating caller function is reached, a memory check . . .")), as recited in the claims of the '591 patent. None of these products are staple goods--they are sophisticated and customized cyber security and malware detection products, methods, and systems.

275. Defendant's infringing actions have continued after the original Complaint was filed. Defendant had knowledge of the '591 patent and of the specific conduct that constituted infringement of the '591 patent at least based on the original Complaint, but has continued to engage in its infringing activity, including by selling, making, configuring, and installing the Accused Products and performing the accused functionality, and by engaging in activities that constitute inducing infringement and contributory infringement as described above.

276. On information and belief, the infringing actions of each partner, customer, and/or end-user of the Accused Products are attributable to Defendant. For example, on information and belief, Defendant directs and controls the activities or actions of its partners or others in connection with the Accused Products by contractual agreement or otherwise requiring partners or others to provide information and instructions to customers who acquire the Accused Products which, when

followed, results in infringement. Defendant further directs and controls the operation of devices executing the Accused Products by programming the software which, when executed by a customer or end user, perform the method steps of at least claim 1 of the ’591 Patent.

277. Plaintiffs have suffered and continue to suffer damages, including lost profits, as a result of Defendant’s infringement of the ’591 Patent. Defendant is therefore liable to Plaintiffs under 35 U.S.C. § 284 for damages in an amount that adequately compensates Plaintiffs for Defendant’s infringement, but no less than a reasonable royalty.

278. Plaintiffs will continue to suffer irreparable harm unless this Court preliminarily and permanently enjoins Defendant, its agents, employees, representatives, and all others acting in concert with Defendant from infringing the ’591 Patent.

279. Defendant’s infringement of the ’591 Patent is knowing and willful. Defendant had actual knowledge of the ’591 Patent at least by the time Plaintiffs filed this lawsuit and had constructive knowledge of the ’591 Patent from at least the date Plaintiffs marked their products with the ’591 Patent and/or provided notice of the ’591 Patent on their website.

280. On information and belief, despite Defendant’s knowledge of the Asserted Patents and Plaintiffs’ patented technology, Defendant made the deliberate decision to sell products and services that they knew infringe the ’591 Patent. Defendant’s continued infringement of the ’591 Patent with knowledge of the ’591 Patent constitutes willful infringement.

281. Plaintiffs’ allegations of infringement, indirect infringement, and willful infringement with respect to this patent are further set forth in Plaintiffs’ Disclosure of Asserted Claims and Preliminary Infringement Contentions Pursuant to the Court’s Standing Order Governing Patent Proceedings served July 12, 2022.

SIXTH CAUSE OF ACTION
(INFRINGEMENT OF THE '844 PATENT)

282. Plaintiffs reallege and incorporate by reference the allegations of the preceding paragraphs of this Complaint.

283. Defendant has infringed and continues to infringe one or more claims of the '844 Patent in violation of 35 U.S.C. § 271 in this District and elsewhere in the United States and will continue to do so unless enjoined by this Court. The Accused Products, including features such as Intercept X Advanced with EDR ("Intercept X"), at least when used for their ordinary and customary purposes, practice each element of at least claim 1 of the '844 Patent, as described below.

284. Claim 1 of the '844 Patent recites:

> 1. A computer-implemented method comprising:
>
> extracting a plurality of static data points from an executable file without decrypting or unpacking the executable file, wherein the plurality of static data points represent predefined character strings in the executable file;
>
> generating a feature vector from the plurality of static data points using a classifier trained to classify the plurality of static data points based on a collection of data comprising known malicious executable files, known benign executable files, and known unwanted executable files, wherein the collection of data comprises at least a portion of the plurality of static data points, and
>
> wherein one or more features of the feature vector are selectively turned on or off 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; and
>
> evaluating the feature vector using support vector processing to determine whether the executable file is harmful.

285. The Accused Products perform each element of the method of claim 1 of the '844 Patent. To the extent the preamble is construed to be limiting, the Accused Products perform *a computer-implemented method*, as further explained below. For example, Intercept X employs a

"deep neural network…trained on hundreds of millions of samples to detect when a file is malicious, potentially unwanted, or legitimate."

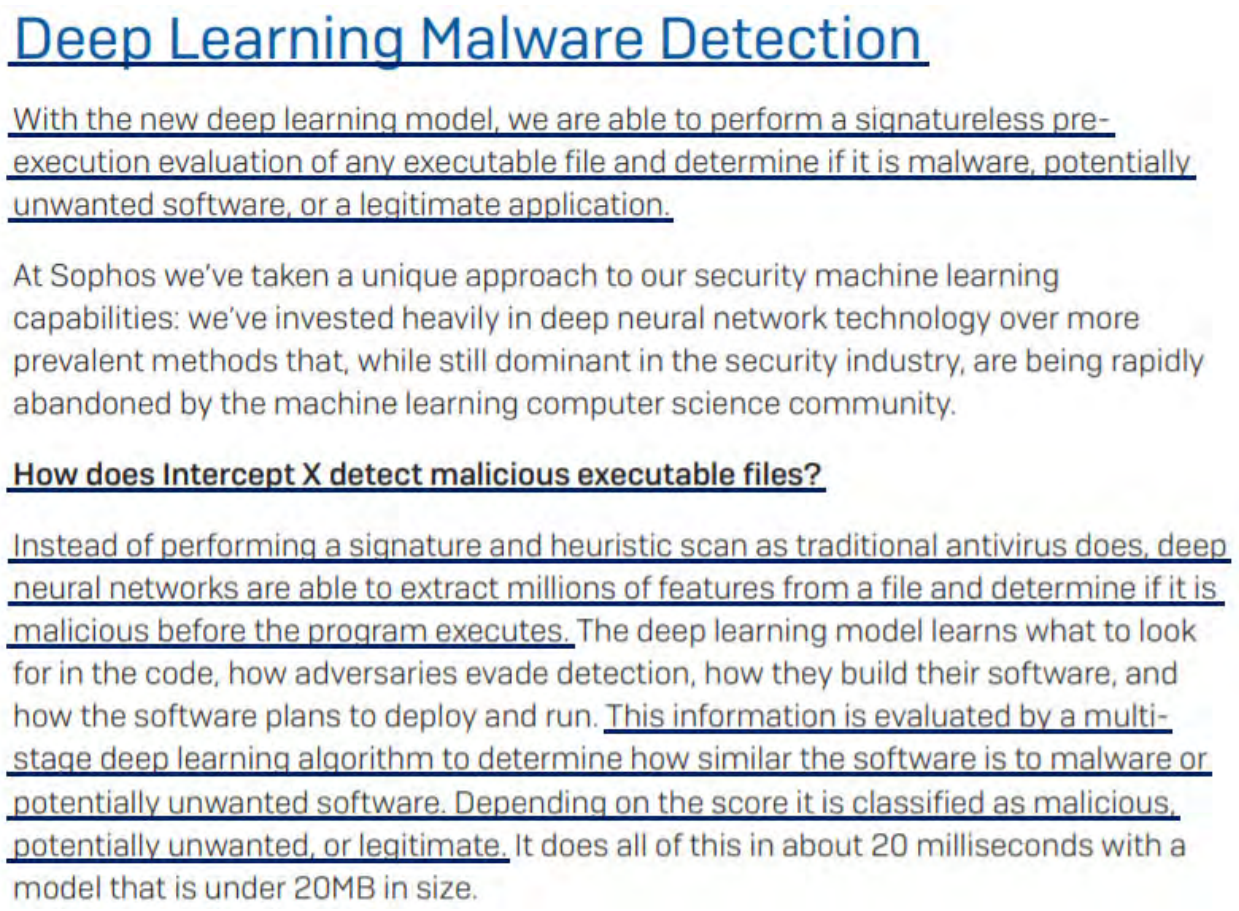
OXFORD, U.K. – Nov. 2, 2017 Sophos (LSE:SOPH), a global leader in network and endpoint security today announced that deep learning driven malware detection is now available through its Intercept X early access program. This deep learning capability has been developed using technology from Invincea, acquired by Sophos in February 2017.

First released in September 2016, Sophos Intercept X is a next-generation endpoint security product that stops zero-day malware, blocks all exploit techniques known today and includes an advanced anti-ransomware feature that can stop both known and unknown ransomware variants within seconds. Deployed through the cloud-based management platform Sophos Central, Intercept X can be installed alongside existing endpoint security software from any vendor, immediately boosting endpoint protection by stopping malicious code before it can do harm.

Deep learning is a branch of machine learning and artificial intelligence that leverages an artificial neural network to build a model used to make predictions with speed, scale, and judgement that exceed human capabilities. The deep neural network of Intercept X is trained on hundreds of millions of samples to detect when a file is malicious, potentially unwanted, or legitimate. Deep learning is more effective than traditional machine learning approaches because of its larger scale training set, smaller model, and more effective detections.

(*See* https://www.sophos.com/en-us/press-office/press-releases/2017/11/sophos-adds-deep-learning-capabilities-to-intercept-x-early-access-program.aspx.)

286. The Accused Products perform a method of *extracting a plurality of static data points from an executable file without decrypting or unpacking the executable file, wherein the plurality of static data points represent predefined character strings in the executable file.* For example, Intercept X's "Deep Learning" technology, "extract[s] millions of features from a file and determine if it is malicious before the program executes." These extracted features include the most "significant strings found in the file," the attribute "Findcrypt," which "shows any suspicious cryptographic constants" found within the file, and the attribute "Resources," which "specifies a resource" within the file "that appears to be compressed or encrypted."



Deep Learning Malware Detection

With the new deep learning model, we are able to perform a signatureless pre-execution evaluation of any executable file and determine if it is malware, potentially unwanted software, or a legitimate application.

At Sophos we've taken a unique approach to our security machine learning capabilities: we've invested heavily in deep neural network technology over more prevalent methods that, while still dominant in the security industry, are being rapidly abandoned by the machine learning computer science community.

How does Intercept X detect malicious executable files?

Instead of performing a signature and heuristic scan as traditional antivirus does, deep neural networks are able to extract millions of features from a file and determine if it is malicious before the program executes. The deep learning model learns what to look for in the code, how adversaries evade detection, how they build their software, and how the software plans to deploy and run. This information is evaluated by a multi-stage deep learning algorithm to determine how similar the software is to malware or potentially unwanted software. Depending on the score it is classified as malicious, potentially unwanted, or legitimate. It does all of this in about 20 milliseconds with a model that is under 20MB in size.

(*See* https://secure2.sophos.com/en-us/medialibrary/Gated-Assets/white-papers/Sophos-Intercept-X-Solution-Brief.pdf.)

Machine learning analysis

Under **Machine learning analysis**, you can see full results of our analysis.

Attributes shows a comparison of the file's attributes with those in millions of known bad and known good files. This enables you to determine how suspicious each attribute is and therefore whether the file is likely to be good or bad. You may see the following attributes:

- **Imports** describes the functionality that the file uses from external DLLs.
- **Strings** describes the most significant strings in the file.
- **Compilers** specifies what was used to compile the source code, for example C++, Delphi, Visual Basic, .NET.
- **Mitigation** describes techniques that the file uses to avoid being exploited.
- **Resources** specifies a resource that seems to be compressed or encrypted.
- **Summary** often relates to build or compilation dates, for example.

- **Packer** often specifies something of note about a specific section of the file, for example a suspicious section name or the fact that a section is both writable and executable.
- **Peid** refers to the output of PEiD, a third-party tool that scans a PE file against various malware signatures.
- **Btcaddress** shows any valid Bitcoin address that is found in the file.
- **Findcrypt** shows any suspicious cryptographic constants.

Code similarity shows a comparison of the file with millions of known bad and known good files, and lists the closest matches. Other matches count toward the result and may affect the rating for the file. The more bad files the file matches and the more closely it matches them, the more suspicious the file is.

File/path shows a comparison of the file's path with that of millions of known bad and known good files. If the file's path more closely matches the path of known bad files, the file is more likely to be suspicious. The path and file name used for comparison are either yours (if you requested the latest intelligence) or those from the last customer who sent us a file. We hide sensitive information in other customers' paths.

(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html.)

287. The Accused Products perform a method of *generating a feature vector from the plurality of static data points using a classifier trained to classify the plurality of static data points based on a collection of data comprising known malicious executable files, known benign executable files, and known unwanted executable files, wherein the collection of data comprises at least a portion of the plurality of static data points*. For example, Intercept X employs "Deep Learning models" (*i.e.*, advanced machine learning models) that have been trained on "hundreds of millions of samples" to classify the features extracted from a file to "determine if a file is benign or malicious…before the file executes." Such samples include "Potentially Unwanted Applications" such as adware. Before using a deep learning model to classify the features extracted from a file, Intercept X creates a "vector of information" that translates those features into data that the model can "intake" and "process."

Performance: Sophos' Deep Learning technology is incredibly fast. In less than 20 milliseconds the model is able to extract millions of features from a file, conduct a deep analysis, and determine if a file is benign or malicious. This entire process happens before the file executes.

SophosLabs: One of the most important aspects to any model is the data that used for training. Our team of data scientists are part of the SophosLabs group, granting them access to hundreds of millions of samples. This allows them to create the best possible predictions in our models. The integration between the two groups also leads to better data labeling (and therefore better modeling). The bi-directional sharing of threat intelligence and real-world feedback between the team of data scientists and threat researchers continuously improves the accuracy of our models.

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/factsheets/sophos-intercept-x-deep-learning-dsna.pdf.)

Feature Engineering in Machine Learning

Before creating a machine learning model, it's important to prepare our data. Preparing the data requires translating it into a language our model can understand. This is referred to as feature engineering.

Artificial neural network models intake data as a vector of information, so simply feeding the model a URL – which is not in the language of a vector – means that the model can't process it without some manipulation. There are countless ways that samples can be translated into features, though it takes some domain knowledge to do so. Using the URL example again, one way to translate a URL into a usable language is through a combination of ngramming and hashing. Ngrams are a popular method in DNA sequencing research. For example, the results of a three-gram ngram for the URL "https://sophos.com/company/careers.aspx" would be:

['htt', 'ttp', 'tps', 'ps:', 's:/', '://', '//s', '/so', 'sop', 'oph', 'pho', 'hos', 'os:', 's.c', '.co', 'com', 'om/', 'm/c', '/co', 'com', 'omp', 'mpa', 'pan', 'any', 'ny/', 'y/c', '/ca', 'car', 'are', 'ree', 'eer', 'ers', 'rs.', 's.a', '.as', 'asp', 'spx']

Once the ngrams are calculated, we need to translate them into a numerical representation. This can be done through a hashing mechanism. We will create an n-length long vector – say 1000 – and hash each ngram using a hashing algorithm. The resulting number from the hash of a particular ngram will be the index of which we will add 1. For example, if the first ngram 'htt' results in a hash of three and our vector is five units long, the result would be [0, 0, 1, 0, 0]. We continue this process for every ngram and for every URL until we have the list of URLs completely transformed into individual n-length vectors. When using this method for our toy model, these vectors will be 1,000 units long.

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/machine-learning-how-to-build-a-better-threat-detection-model.pdf?cmp=70130000001xKqzAAE.)

2.1 MLP Model

An MLP is a class of feedforward ANN with an input layer, an output layer, and one or more hidden layers between them. Each node in one layer fully connects to every node in the following layer. Except for nodes in the input layer, each node is a neuron (or processing element) with a nonlinear activation function associated with a scalar weight which is adjusted during training. An MLP is a supervised learning algorithm that learns to identify data patterns for classification or regression. In order to carry out the learning process, one needs to extract a digital representation $\boldsymbol{x}$ of a given object or event that needs to be fed into the MLP. The learning task becomes to find a multidimensional function $\Phi(\cdot)$ between input $\boldsymbol{x}$ and target $\boldsymbol{y}$

$$\boldsymbol{y} \cong \Phi(\boldsymbol{x}) \quad [1]$$

where $\boldsymbol{x} \in \mathbb{R}^N$, a real-valued input feature vector $\boldsymbol{x} = [x_1, \dots, x_N]^T$ in an N dimensional feature space, with $(\cdot)^T$ denoting the transpose operation. Similarly, $\boldsymbol{y} \in \mathbb{R}^M$, a real-valued target classification vector $\boldsymbol{y} = [y_1, \dots, y_M]^T$ in an M dimensional classification space. In other words, an MLP learning process is to find $\Phi(\cdot)$

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophos-black-hat-2018-technicalpaper.pdf.)

Generic ML PUA detections

Potentially Unwanted Application (PUA) is a term used to describe applications that, while not malicious, are generally considered unsuitable for business networks.

A Generic ML PUA detection is generated by Sophos Intercept X's Machine Learning (ML) engine, also referred to by the specific Sophos approach Deep Learning and is designed to detect PUAs in PE (Portable Executable) files such as:

- .exe
- .sys
- .dll
- .pif
- .scr
- and many more

If a detection of this type has been received, it is because Sophos has detected a file on the computer that our Deep Learning threat model has decided is a PUA. This is a pre-execution detection meaning the file was detected before it was able to be run.

Major PUA classifications

The major PUA classifications are:

- adware
- dialer
- non-malicious spyware
- remote administration tools
- hacking tools

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophos-black-hat-2018-technicalpaper.pdf.)

288. The feature vector is "evaluated by a multi-stage deep learning algorithm to determine how similar the software is to malware or potentially unwanted software. Depending on the score [the file] is classified as malicious, potentially unwanted, or legitimate."

Deep Learning Malware Detection

With the new deep learning model, we are able to perform a signatureless pre-execution evaluation of any executable file and determine if it is malware, potentially unwanted software, or a legitimate application.

At Sophos we've taken a unique approach to our security machine learning capabilities: we've invested heavily in deep neural network technology over more prevalent methods that, while still dominant in the security industry, are being rapidly abandoned by the machine learning computer science community.

How does Intercept X detect malicious executable files?

Instead of performing a signature and heuristic scan as traditional antivirus does, deep neural networks are able to extract millions of features from a file and determine if it is malicious before the program executes. The deep learning model learns what to look for in the code, how adversaries evade detection, how they build their software, and how the software plans to deploy and run. This information is evaluated by a multi-stage deep learning algorithm to determine how similar the software is to malware or potentially unwanted software. Depending on the score it is classified as malicious, potentially unwanted, or legitimate. It does all of this in about 20 milliseconds with a model that is under 20MB in size.

(*See* https://secure2.sophos.com/en-us/medialibrary/Gated-Assets/white-papers/Sophos-Intercept-X-Solution-Brief.pdf.)

289. The Accused Products perform a method *wherein one or more features of the feature vector are selectively turned on or off 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*. For example, Intercept X employs "feature selection" to only "keep relevant features before feeding them into" its deep learning models, "identifying and removing as much noisy and redundant information as possible from extracted features."

> A crucial step in an ML workflow is feature extraction, which can be hand-crafted-based on human expertise, or automatically learned by training modern deep learning models such as convolutional neural networks (CNNs). It is natural to believe that more extracted features can provide better characterization of a learning task and more discriminating power. However, increasing the dimension of the feature vector could result in feature redundancy and noise. Redundant and irrelevant features can cause performance deterioration of an ML model with overfitting and generalization problems. Additionally, excessively increased number of features could significantly slow down a learning process. Therefore, it is of fundamental importance to only keep relevant features before feeding them into an ML model, which leads to requiring feature selection (or feature dimensionality reduction). Feature selection can be seen as the process of identifying and removing as much noisy and redundant information as possible from extracted features.

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophoslabs-machine-learning-tp.pdf.)

290. The Accused Products perform a method of *evaluating the feature vector using support vector processing to determine whether the executable file is harmful*. For example, and as explained above, Intercept X employs "Deep Learning models" (*i.e.*, advanced machine learning models) that have been trained on "hundreds of millions of samples" to classify the features extracted from a file to "determine if a file is benign or malicious…before the file executes." Before using a deep learning model to classify the features extracted from a file, Intercept X creates a "vector of information" that translates those features into data that the model can "intake" and "process."

Performance: Sophos' Deep Learning technology is incredibly fast. In less than 20 milliseconds the model is able to extract millions of features from a file, conduct a deep analysis, and determine if a file is benign or malicious. This entire process happens before the file executes.

SophosLabs: One of the most important aspects to any model is the data that used for training. Our team of data scientists are part of the SophosLabs group, granting them access to hundreds of millions of samples. This allows them to create the best possible predictions in our models. The integration between the two groups also leads to better data labeling (and therefore better modeling). The bi-directional sharing of threat intelligence and real-world feedback between the team of data scientists and threat researchers continuously improves the accuracy of our models.

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/factsheets/sophos-intercept-x-deep-learning-dsna.pdf.)

Feature Engineering in Machine Learning

Before creating a machine learning model, it's important to prepare our data. Preparing the data requires translating it into a language our model can understand. This is referred to as feature engineering.

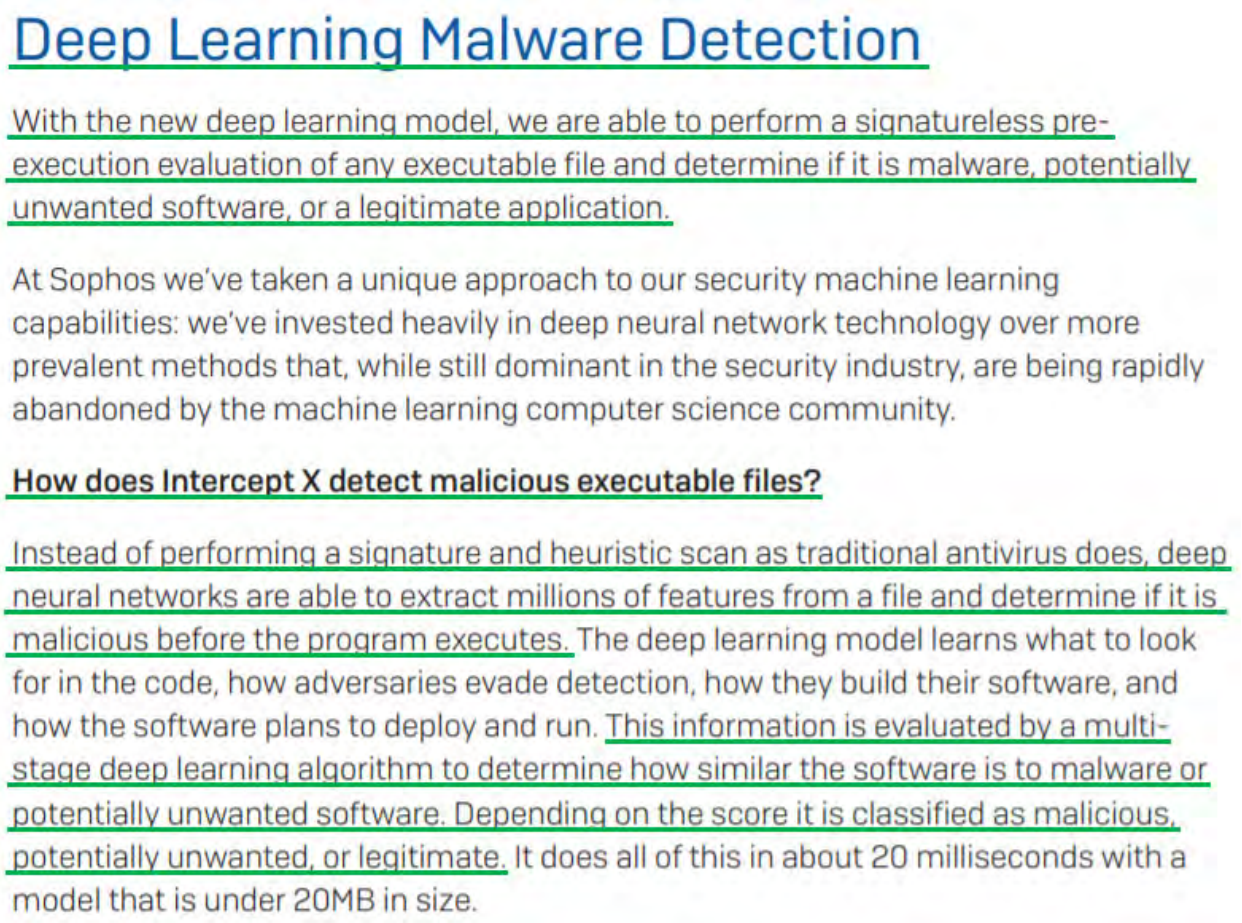
Artificial neural network models intake data as a vector of information, so simply feeding the model a URL – which is not in the language of a vector – means that the model can't process it without some manipulation. There are countless ways that samples can be translated into features, though it takes some domain knowledge to do so. Using the URL example again, one way to translate a URL into a usable language is through a combination of ngramming and hashing. Ngrams are a popular method in DNA sequencing research. For example, the results of a three-gram ngram for the URL "https://sophos.com/company/careers.aspx" would be:

['htt', 'ttp', 'tps', 'ps:', 's:/', '://', '//s', '/so', 'sop', 'oph', 'pho', 'hos', 'os:', 's.c', '.co', 'com', 'om/', 'm/c', '/co', 'com', 'omp', 'mpa', 'pan', 'any', 'ny/', 'y/c', '/ca', 'car', 'are', 'ree', 'eer', 'ers', 'rs.', 's.a', '.as', 'asp', 'spx']

Once the ngrams are calculated, we need to translate them into a numerical representation. This can be done through a hashing mechanism. We will create an n-length long vector – say 1000 – and hash each ngram using a hashing algorithm. The resulting number from the hash of a particular ngram will be the index of which we will add 1. For example, if the first ngram 'htt' results in a hash of three and our vector is five units long, the result would be [0, 0, 1, 0, 0]. We continue this process for every ngram and for every URL until we have the list of URLs completely transformed into individual n-length vectors. When using this method for our toy model, these vectors will be 1,000 units long.

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/machine-learning-how-to-build-a-better-threat-detection-model.pdf?cmp=70130000001xKqzAAE.)

291. The feature vector is "evaluated by a multi-stage deep learning algorithm to determine how similar the software is to malware or potentially unwanted software. Depending on the score [the file] is classified as malicious, potentially unwanted, or legitimate." That evaluation is processed using support vector processing.



Deep Learning Malware Detection

With the new deep learning model, we are able to perform a signatureless pre-execution evaluation of any executable file and determine if it is malware, potentially unwanted software, or a legitimate application.

At Sophos we've taken a unique approach to our security machine learning capabilities: we've invested heavily in deep neural network technology over more prevalent methods that, while still dominant in the security industry, are being rapidly abandoned by the machine learning computer science community.

How does Intercept X detect malicious executable files?

Instead of performing a signature and heuristic scan as traditional antivirus does, deep neural networks are able to extract millions of features from a file and determine if it is malicious before the program executes. The deep learning model learns what to look for in the code, how adversaries evade detection, how they build their software, and how the software plans to deploy and run. This information is evaluated by a multi-stage deep learning algorithm to determine how similar the software is to malware or potentially unwanted software. Depending on the score it is classified as malicious, potentially unwanted, or legitimate. It does all of this in about 20 milliseconds with a model that is under 20MB in size.

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophoslabs-machine-learning-tp.pdf.)

292. Each claim in the ’844 Patent recites an independent invention. Neither claim 1, described above, nor any other individual claim is representative of all claims in the ‘844 Patent.

293. Defendant has been aware of the ’844 Patent since at least the filing of this Complaint. Further, Plaintiffs have marked its products with the ’844 Patent, including on its web site, since at least July 2020.

294. Defendant directly infringes at least claim 1 of the ’844 Patent, either literally or under the doctrine of equivalents, by performing the steps described above. For example, on information and belief, Defendant performs the claimed method as described above by running the

Sophos security software and corresponding systems to protect its own computer and network operations. On information and belief, Defendant also performs the claimed method as described above when testing the operation of the Accused Products and corresponding systems. As another example, Defendant performs the claimed method as described above when providing or administering services to third parties, customers, and partners using the Accused Products.

295. Defendant's partners, customers, and end users of the Accused Products and corresponding systems and services directly infringe at least claim 1 of the '844 Patent, literally or under the doctrine of equivalents, at least by using the Accused Products and corresponding systems and services, as described above.

296. Defendant has actively induced and is actively inducing infringement of at least claim 1 of the '844 Patent with specific intent to induce infringement, and/or willful blindness to the possibility that its acts induce infringement, in violation of 35 U.S.C. § 271(b). For example, Sophos encourages and induces customers with specific intent to use Sophos's security software in a manner that infringes claim 1 of the '844 Patent at least by offering and providing software that performs a method that infringes claim 1 when installed and operated by the customer, and by engaging in activities relating to selling, marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including the activities described below.

297. Defendant encourages, instructs, directs, and/or requires third parties—including its certified partners and/or customers to perform the claimed method using the software, services, and systems in infringing ways, as described above.

298. Defendant further encourages and induces its customers to infringe claim 1 of the '844 Patent: 1) by making its security services available on its website, providing applications that allow users to access those services, widely advertising those services, and providing technical

support and instructions to users, and 2) through activities relating to marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including its Sophos security software, and services in the United States. (*See* https://www.sophos.com/en-us/products/endpoint-antivirus/how-to-buy.aspx; https://partners.sophos.com/english/directory/search?lat=30.267153&lng=-97.7430608&dMI=100&p=1; https://secure2.sophos.com/en-us/security-news-trends/whitepapers/gated-wp/cybersecurity-system-buyers-guide.aspx%23formFrame; *see also* https://www.sophos.com/en-us/medialibrary/PDFs/factsheets/sophos-synchronized-security-ds.pdf.)

299. For example, on information and belief, Defendant shares instructions, guides, and manuals, which advertise and instruct third parties on how to use the software as described above, including at least customers and partners. (*Id*.) On further information and belief, Defendant provides customer service and technical support to purchasers of the Accused Products and corresponding system and services, which directs and encourages customers to perform certain actions that use the Accused Products in an infringing manner. (*Id*.)

300. Defendant and/or its partners recommend and sell the Accused Products and provide technical support for the installation, implementation, integration, and ongoing operation of the Accused Products for each individual customer. On information and belief, each customer enters into a contractual relationship with Defendant and/or one of its partners, which obligates each customer to perform certain actions in order to use the Accused Products. Further, in order to receive the benefit of Defendant's or its partner's continued technical support and their specialized knowledge and guidance of the operability of the Accused Products, each customer must continue to use the Accused Products in a way that performs the claimed method and infringes the '844 Patent.

301. Further, as the entity that provides installation, implementation, and integration of the Accused Products in addition to ensuring the Accused Product remains operational for each customer through ongoing technical support, on information and belief, Defendant and/or Sophos's partners affirmatively aid and abet each customer's use of the Accused Products in a manner that performs the claimed method of, and infringes, the '844 Patent.

302. Defendant also contributes to the infringement of its partners, customers, and end-users of the Accused Products by providing within the United States or importing the Accused Products into the United States, which are for use in practicing, and under normal operation practice, methods claimed in the Asserted Patents, constituting a material part of the claimed methods, and not a staple article or commodity of commerce suitable for substantial non-infringing uses. Indeed, as shown above, the Accused Products and the example functionality described below have no substantial non-infringing uses but are specifically designed to practice the '844 Patent.

303. Indeed, as shown above, the Accused Products have no substantial non-infringing uses because the accused functionality, including the machine learning static analysis and related functionality described above, is an integral part of the Accused Products and must be performed for the Accused Products to perform their intended purpose. For example, without extracting a set of static data points from an object, the Accused Products could not create feature vector that they could feed to their machine learning model. These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer suitably function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice \the '844 patent, that functionality could not be

performed.

304. Additionally, the accused functionality, including the machine learning static analysis and related functionality described above, itself has no substantial non-infringing uses because the components, modules and methods identified above are a necessary part of that functionality. For example, without extracting and analyzing a set of static features from an object and calibrating a feature vector on the basis of that analysis, the Accused Products could not deploy their machine learning model to determine whether or not that object is malicious or benign. These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice the '844 patent, that functionality could not be performed.

305. In addition, as shown in the detailed analysis above, the products, systems, modules, and methods provided by Defendant constitute a material part of the invention—indeed, they provide all the components, modules, and features that perform the claimed methods and systems. For example, the Accused Products and accused functionality (including the machine learning static anlaysis) constitute a material part of the inventions claimed at least because such an attack chain is integral to the processes identified above (such as "extracting a plurality of static data points from an executable file . . ."; "generating a feature vector from the plurality of static data points . . ."; "wherein one or more features of the feature vector are selectively turned on or off . . ."; "evaluating the feature vector . . .") as recited in the claims of the '844 patent. None of these products are staple goods—they are sophisticated and customized cyber security and malware detection products, methods, and systems.

306. Defendant's infringing actions have continued after the original Complaint was filed. Defendant had knowledge of the '844 patent and of the specific acts that constitutes infringement of the '844 patent at least based on the original Complaint, but has continued to engage in the infringing activity, including by selling, making, configuring, and installing the Accused Products and performing the accused functionality, and by engaging in activities that constituted infringement and contributory infringement as described above.

307. On information and belief, the infringing actions of each partner, customer, and/or end-user of the Accused Products are attributable to Defendant. For example, on information and belief, Defendant directs and controls the activities or actions of its partners or others in connection with the Accused Products by contractual agreement or otherwise requiring partners or others to provide information and instructions to customers who acquire the Accused Products which, when followed, results in infringement. Defendant further directs and controls the operation of devices executing the Accused Products by programming the software which, when executed by a customer or end user, perform the method steps of at least claim 1 of the '844 Patent.

308. Plaintiffs have suffered and continue to suffer damages, including lost profits, as a result of Defendant's infringement of the '844 Patent. Defendant is therefore liable to Plaintiffs under 35 U.S.C. § 284 for damages in an amount that adequately compensates Plaintiffs for Defendant's infringement, but no less than a reasonable royalty.

309. Plaintiffs will continue to suffer irreparable harm unless this Court preliminarily and permanently enjoins Defendant, its agents, employees, representatives, and all others acting in concert with Defendant from infringing the '844 Patent.

310. Defendant's infringement of the '844 Patent is knowing and willful. Defendant had actual knowledge of the '844 Patent at least by the time Plaintiffs filed this lawsuit and had

constructive knowledge of the ’844 Patent from at least the date Plaintiffs marked their products with the ’844 Patent and/or provided notice of the ’844 Patent on their website.

311. On information and belief, despite Defendant’s knowledge of the Asserted Patents and Plaintiffs’ patented technology, Defendant made the deliberate decision to sell products and services that they knew infringe the ’844 Patent. Defendant’s continued infringement of the ’844 Patent with knowledge of the ’844 Patent constitutes willful infringement.

312. Plaintiffs’ allegations of infringement, indirect infringement, and willful infringement with respect to this patent are further set forth in Plaintiffs’ Disclosure of Asserted Claims and Preliminary Infringement Contentions Pursuant to the Court’s Standing Order Governing Patent Proceedings served July 12, 2022.

SEVENTH CAUSE OF ACTION
(INFRINGEMENT OF THE ’721 PATENT)

313. Plaintiffs reallege and incorporate by reference the allegations of the preceding paragraphs of this Complaint.

314. Sophos has infringed and continues to infringe one or more claims of the ’721 Patent in violation of 35 U.S.C. § 271 in this District and elsewhere in the United States and will continue to do so unless enjoined by this Court. The Accused Products, including features such as Sophos Firewall, at least when used for their ordinary and customary purposes, practice each element of at least claim 1 of the ’721 Patent as described below.

315. For example, claim 1 of the ’721 Patent recites:

> 1. A method of classifying a computer object as malware, the method comprising:
>
> receiving, at a first threat server, details of a first computer object from a first remote computer, wherein the details of the first computer object include data uniquely identifying the first computer object;

determining, by the first threat server, whether the first computer object has been previously seen by comparing the data uniquely identifying the first computer object to a plurality of data uniquely identifying plural computer objects in a first database associated with the first threat server;

receiving additional information about the first computer object from the first remote computer when the first computer object has not been previously seen;

storing the details of the first computer object and the received additional information about the first computer object in a second database associated with the first threat server when the first computer object has not been previously seen;

providing contents of the second database to at least one database associated with a central server, wherein the contents comprise a signature of the first computer object, behavior information about the first computer object, and information about the first remote computer;

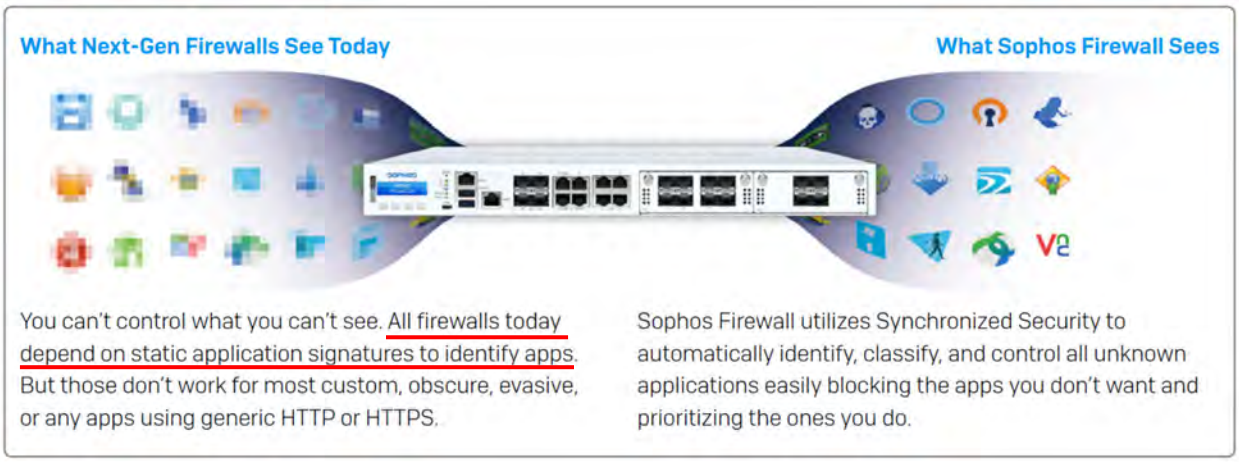
increasing a count associated with a number of times that the first computer object has been seen, and providing the increased count associated with the number of times that the first computer object has been seen to the central server; and

receiving, at a second threat server, at least a portion of the contents of the at least one database associated with the central server, wherein the at least a portion of the contents of the at least one database associated with the central server include a subset of the details of the first computer object stored in the second database.

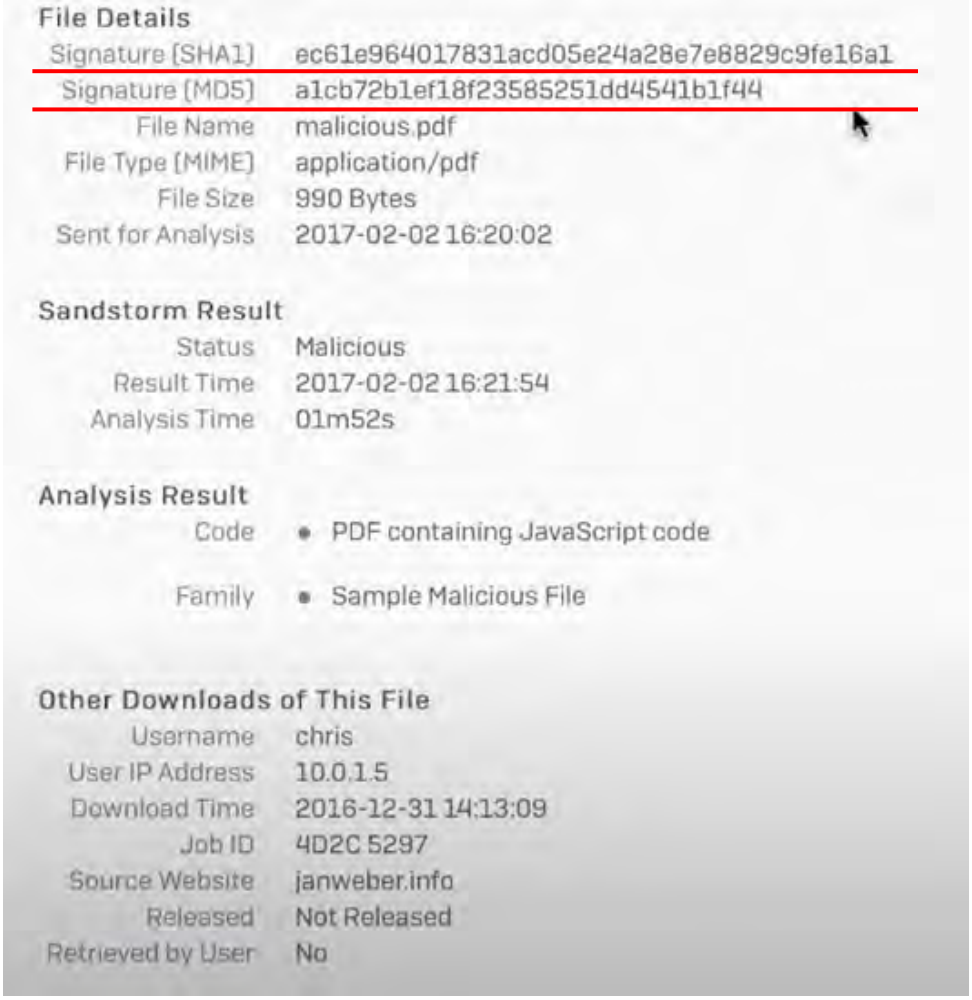
316. To the extent the preamble is construed as limiting, the Accused Products include a method for *classifying malware* as explained below.

317. The Accused Products perform a method that includes *receiving, at a first threat server details of a first computer object from a first remote computer wherein the details of the first computer object include data uniquely identifying the first computer object*. For example, Sophos' Firewall connects to the Sophos endpoint protection service installed at a remote computer through Sophos Heartbeat. Through that connection to the remote computer, Sophos Firewall receives data about computer objects and, on information and belief, like "all firewalls," uses a static application signature received from the endpoints to identify computer objects.

Cryptographically strong hashes such as MD5 and SHA1, as shown below, uniquely identify an object (*e.g.*, file named "malicious.pdf").



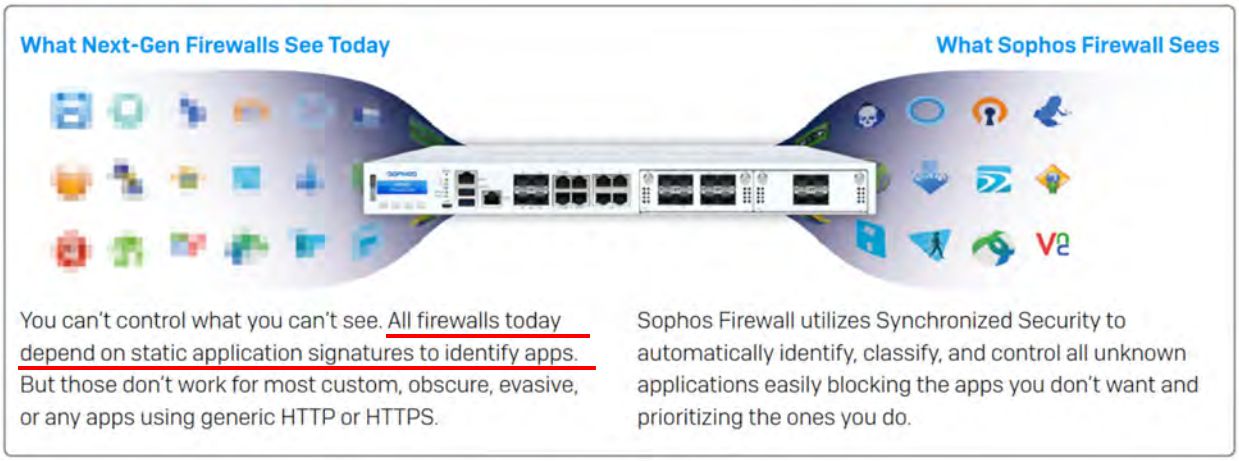
(*See* https://www.sophos.com/en-us/medialibrary/pdfs/factsheets/sophos-firewall-br.pdf (annotations added); *see also* https://www.sophos.com/en-us/medialibrary/pdfs/factsheets/sophos-firewall-br.pdf.)



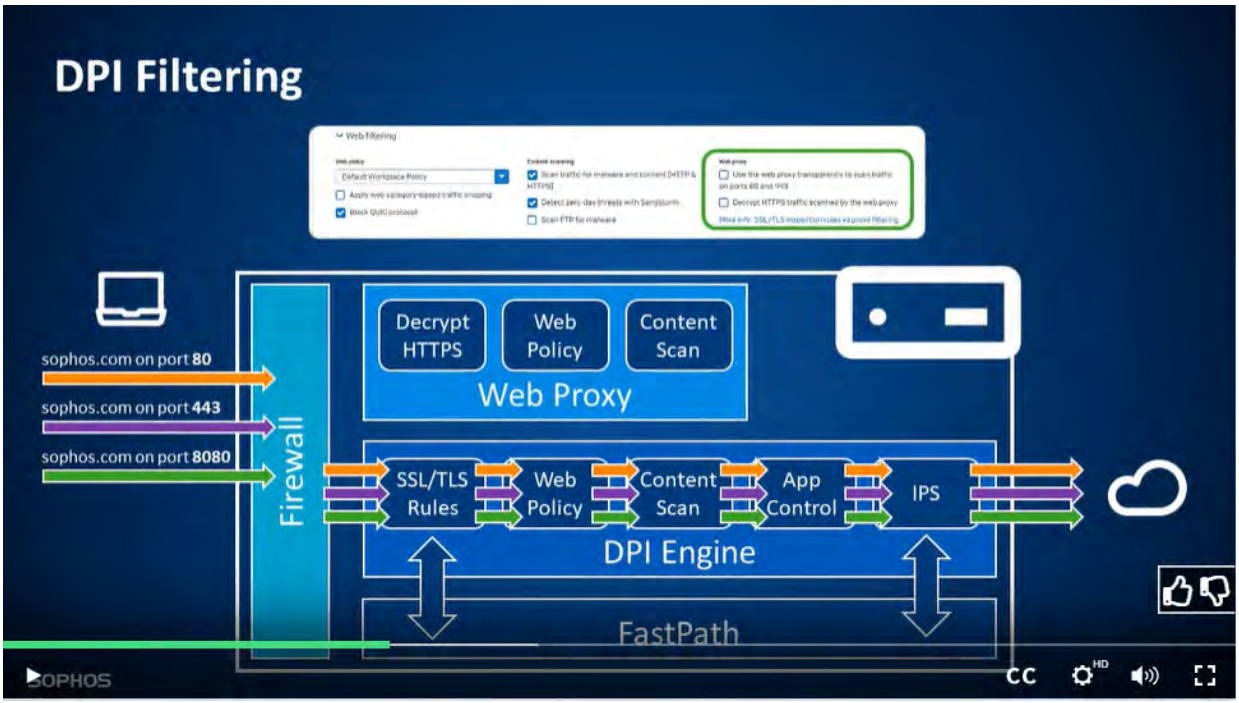
(*See* https://www.youtube.com/watch?v=YR4CR4Sht3A.)

318. The Accused Products perform a method that includes *determining, by the first threat server whether the first computer object has been previously seen by comparing the data uniquely identifying the first computer object to a plurality of data uniquely identifying plural computer objects in a first database associated with the first threat server*. As explained above, Sophos Firewall receives data about computer objects and, on information and belief, like "all firewalls," uses a static application signature received from the endpoints to identify computer objects. For example, if the Firewall does not have the data in its CPU to enact Fastpath —a mechanism by which trusted data or objects bypass security measures— automatically, Sophos

Firewall looks at static signature data from the unique computer object and compares that information with information provided by Sophos' Talos IPS Signature database. The object then travels along two paths: Sophos Fastpath or a standard protocol. Fastpath is reserved for known permissible objects, and the standard track is used for unknown or malicious objects.



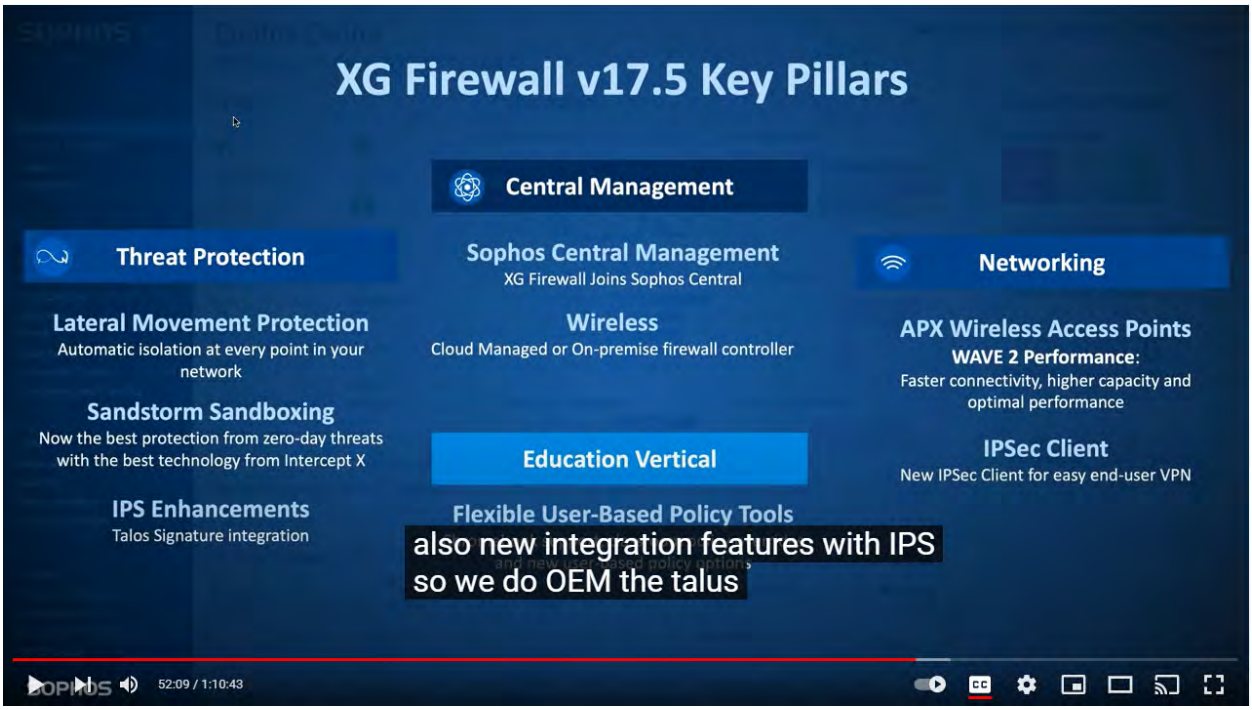
(*See* https://www.sophos.com/en-us/medialibrary/pdfs/factsheets/sophos-firewall-br.pdf.)



(*See* https://techvids.sophos.com/watch/uCC7QqkYcTJtLiBMNhZV32.)

FastPath network flow offloads (bypasses processing of) trusted traffic. Offloading eliminates the need to apply full firewall processing to every packet in a connection, minimizing the use of processing cycles.

(*See* https://docs.sophos.com/nsg/sophos-firewall/18.0/Help/en-us/webhelp/onlinehelp/nsg/sfos/concepts/Architecture.html; *see also* https://www.youtube.com/watch?v=YR4CR4Sht3A.)



(*See* https://www.youtube.com/watch?v=aXNo4V2A1Gw.)

(*See* https://www.sophos.com/en-us/medialibrary/pdfs/factsheets/sophos-firewall-br.pdf.)

319. The Accused Products perform a method that includes *receiving additional information about the first computer object from the first remote computer when the first computer object has not been previously seen*. For example, if Sophos Firewall cannot immediately identify the object based on its Fastpath data or there is no match in Talos database, then Sophos Firewall receives more information about the unknown object through Sophos Heartbeat. As explained below, that additional data is provided to an Advanced Threat Protection ("ATP") database, which "provides" information on "JavaScript emulation, behavioral analysis, and origin reputation" based on the information received from the endpoints.

Synchronized Security

Security Heartbeat™: Your firewall and your endpoints are finally talking

Sophos Firewall is the only network security solution that is able to fully identify the user and source of an infection on your network, and automatically limit access to other network resources in response. This is made possible with our unique Sophos Security Heartbeat that shares telemetry and health status between Sophos endpoints and your firewall and integrates endpoint health into firewall rules to control access and isolate compromised systems.

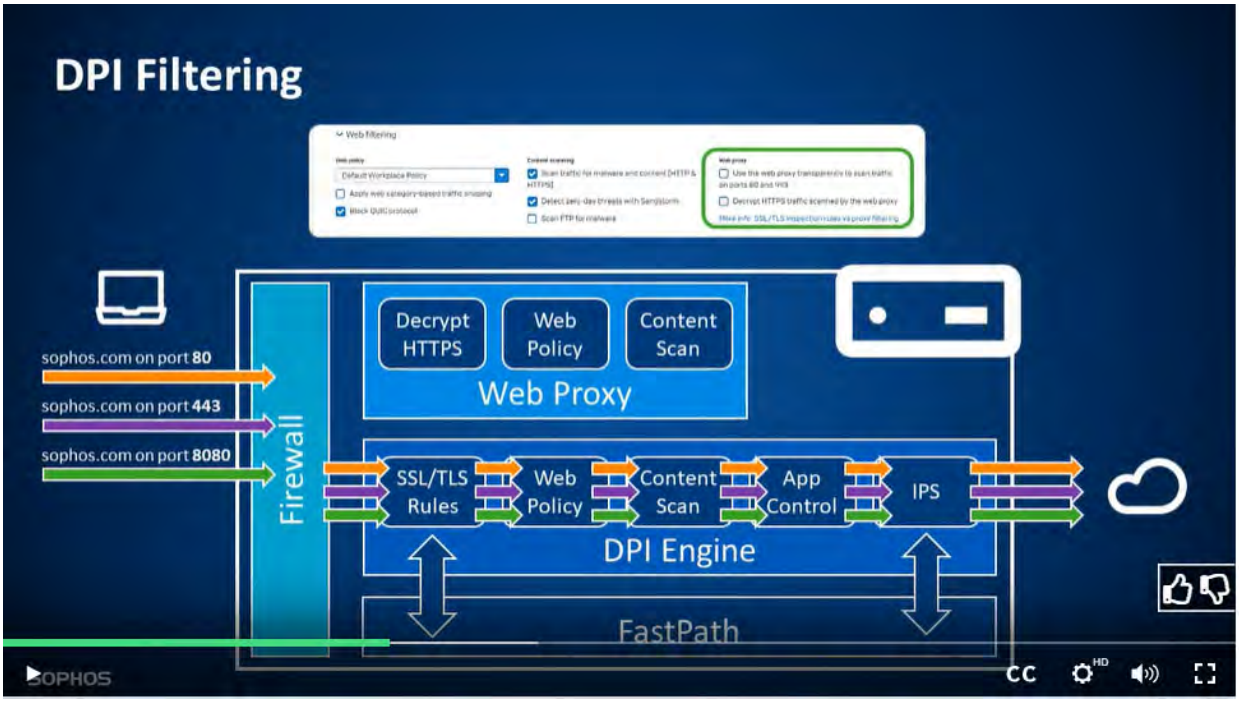
The good news is, this all happens automatically, and is successfully helping numerous businesses and organizations to save time and money in protecting their environments today.

Synchronized Application Control

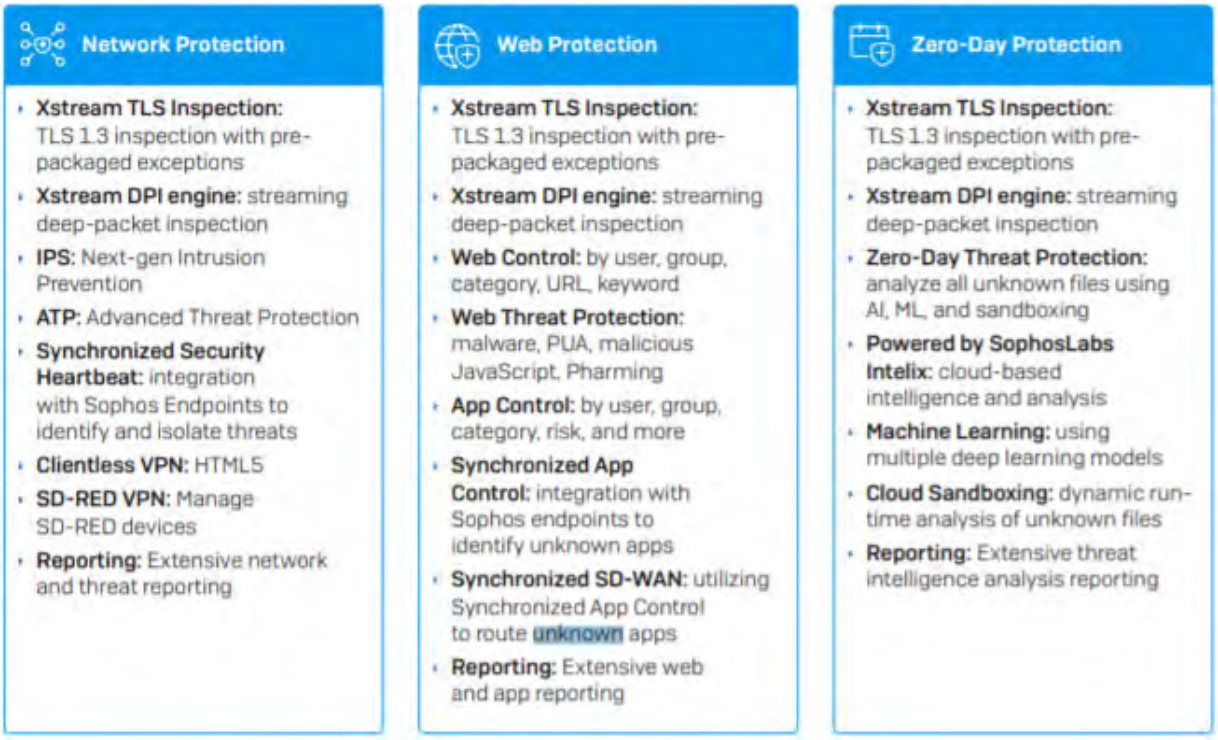
Using Security Heartbeat, we can do much more than just see the health status of an endpoint. We also have a solution to one of the biggest problems most network administrators face today - lack of visibility into network traffic.

Synchronized Application Control utilizes the Heartbeat connections with Sophos endpoints to automatically identify, classify, and control application traffic. All encrypted, custom, evasive, and generic HTTP or HTTPS applications which are currently going unidentified will be revealed.

(*See* https://www.sophos.com/en-us/medialibrary/pdfs/factsheets/sophos-firewall-br.pdf.)



(*See* https://techvids.sophos.com/watch/uCC7QqkYcTJtLiBMNhZV32.)



Network Protection

- **Xstream TLS Inspection:** TLS 1.3 inspection with pre-packaged exceptions
- **Xstream DPI engine:** streaming deep-packet inspection
- **IPS:** Next-gen Intrusion Prevention
- **ATP:** Advanced Threat Protection
- **Synchronized Security Heartbeat:** integration with Sophos Endpoints to identify and isolate threats
- **Clientless VPN:** HTML5
- **SD-RED VPN:** Manage SD-RED devices
- **Reporting:** Extensive network and threat reporting

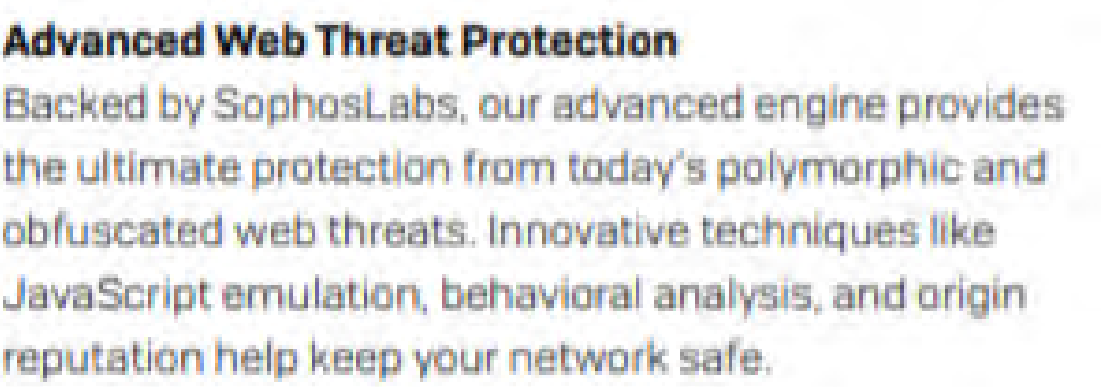
Web Protection

- **Xstream TLS Inspection:** TLS 1.3 inspection with pre-packaged exceptions
- **Xstream DPI engine:** streaming deep-packet inspection
- **Web Control:** by user, group, category, URL, keyword
- **Web Threat Protection:** malware, PUA, malicious JavaScript, Pharming
- **App Control:** by user, group, category, risk, and more
- **Synchronized App Control:** integration with Sophos endpoints to identify unknown apps
- **Synchronized SD-WAN:** utilizing Synchronized App Control to route unknown apps
- **Reporting:** Extensive web and app reporting

Zero-Day Protection

- **Xstream TLS Inspection:** TLS 1.3 inspection with pre-packaged exceptions
- **Xstream DPI engine:** streaming deep-packet inspection
- **Zero-Day Threat Protection:** analyze all unknown files using AI, ML, and sandboxing
- **Powered by SophosLabs Intelix:** cloud-based intelligence and analysis
- **Machine Learning:** using multiple deep learning models
- **Cloud Sandboxing:** dynamic run-time analysis of unknown files
- **Reporting:** Extensive threat intelligence analysis reporting

(*See* https://www.sophos.com/en-us/medialibrary/pdfs/factsheets/sophos-firewall-br.pdf.)



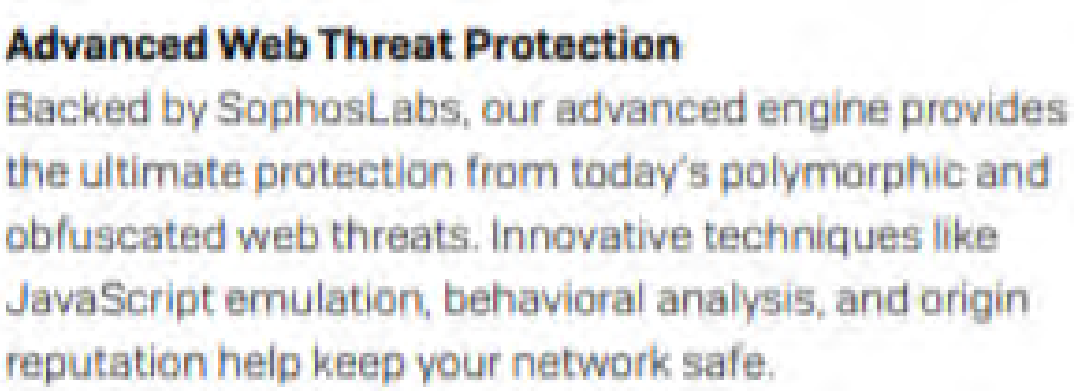
Advanced Web Threat Protection

Backed by SophosLabs, our advanced engine provides the ultimate protection from today's polymorphic and obfuscated web threats. Innovative techniques like JavaScript emulation, behavioral analysis, and origin reputation help keep your network safe.

(*See* https://www.sophos.com/en-us/medialibrary/pdfs/factsheets/sophos-firewall-br.pdf.)

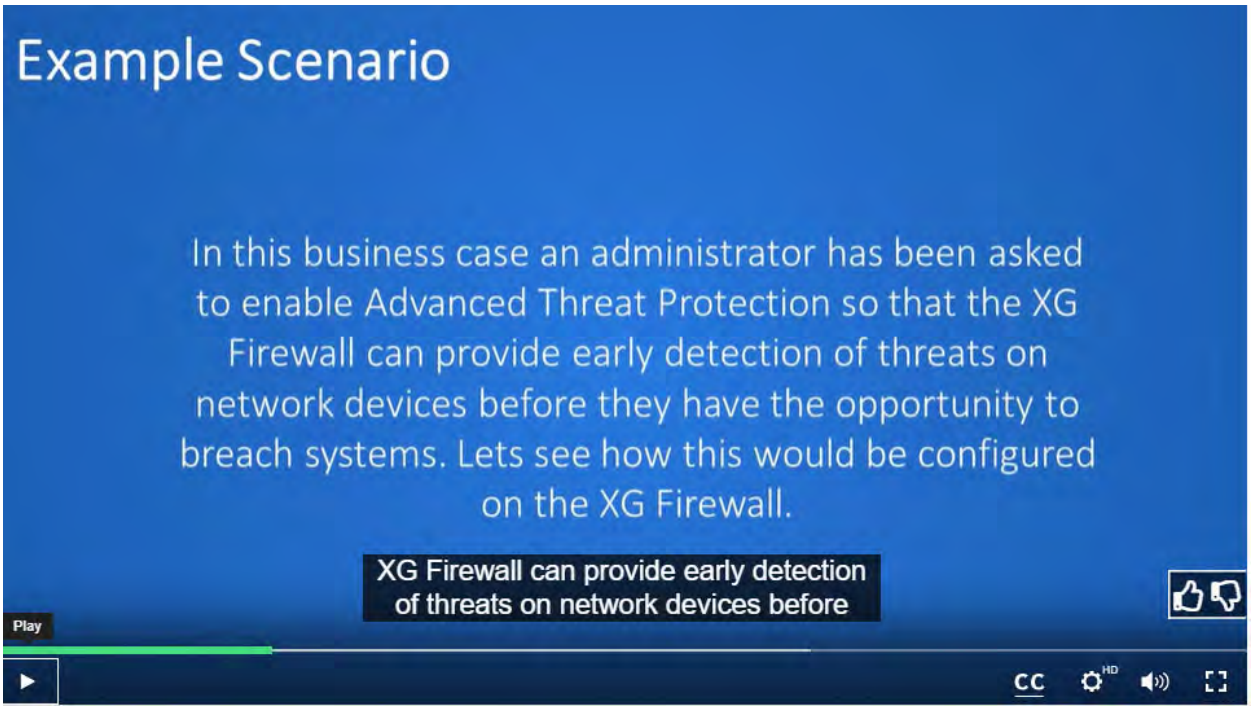
320. The Accused Products perform a method that includes *storing the details of the first computer object and the received additional information about the first computer object in a second database associated with the first threat server when the first computer object has not been previously seen.* For example, after Sophos Firewall determines that it has not previously seen the

computer object, it then sends the information it has received about the object to the Advanced Threat Protection ("ATP") database associated with that Sophos Firewall. The ATP database collects and "provides" information on "JavaScript emulation, behavioral analysis, and origin reputation." On information and belief, each Sophos Firewall is associated with an ATP database. For example, after a first firewall encounters a new object, the system administrator must update other firewalls on the network with the new information from the first firewall, thereby indicating there are multiple ATP databases.

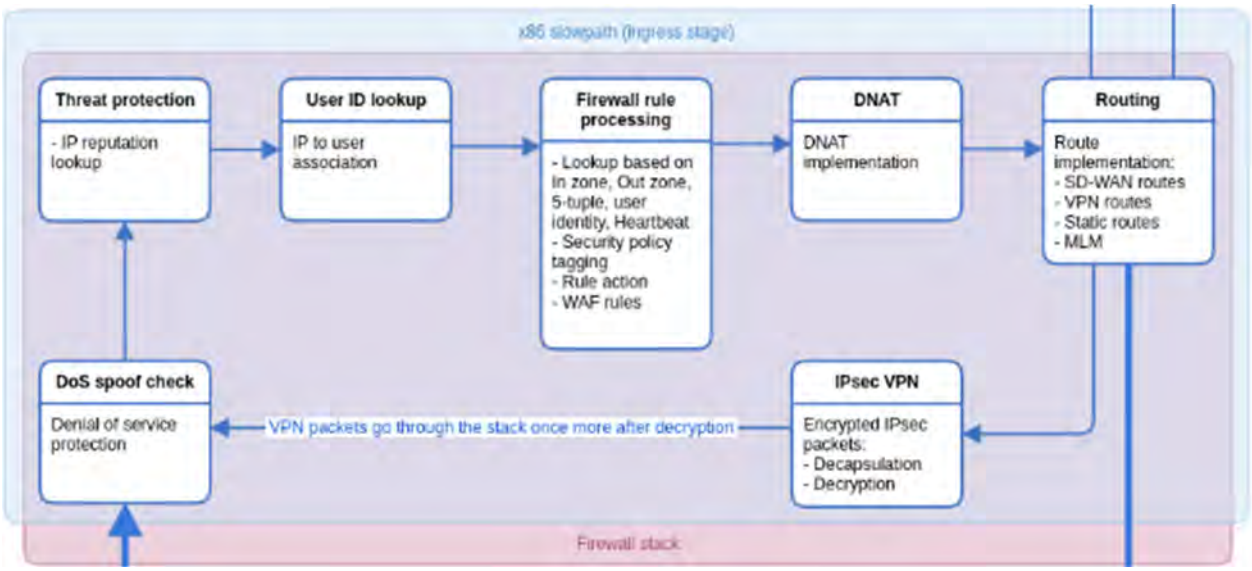


(*See* https://www.sophos.com/en-us/medialibrary/pdfs/factsheets/sophos-firewall-br.pdf.)

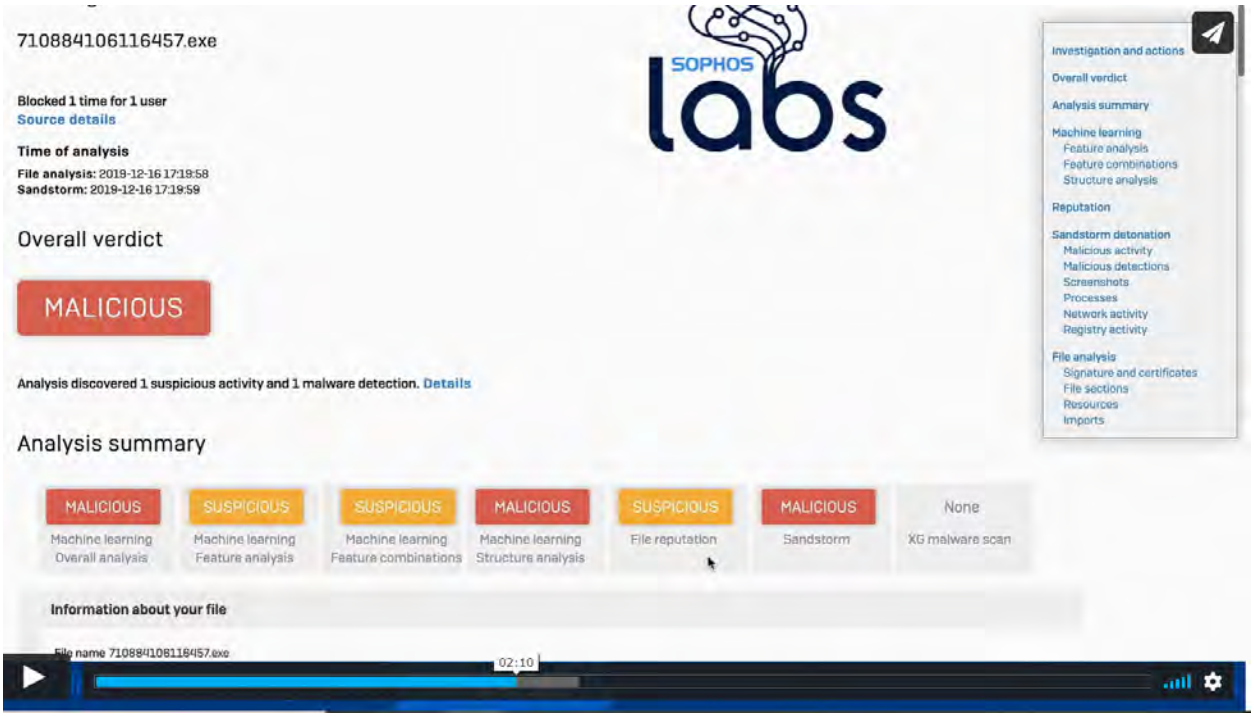
(*See* https://www.sophos.com/en-us/medialibrary/pdfs/factsheets/sophos-firewall-br.pdf.)



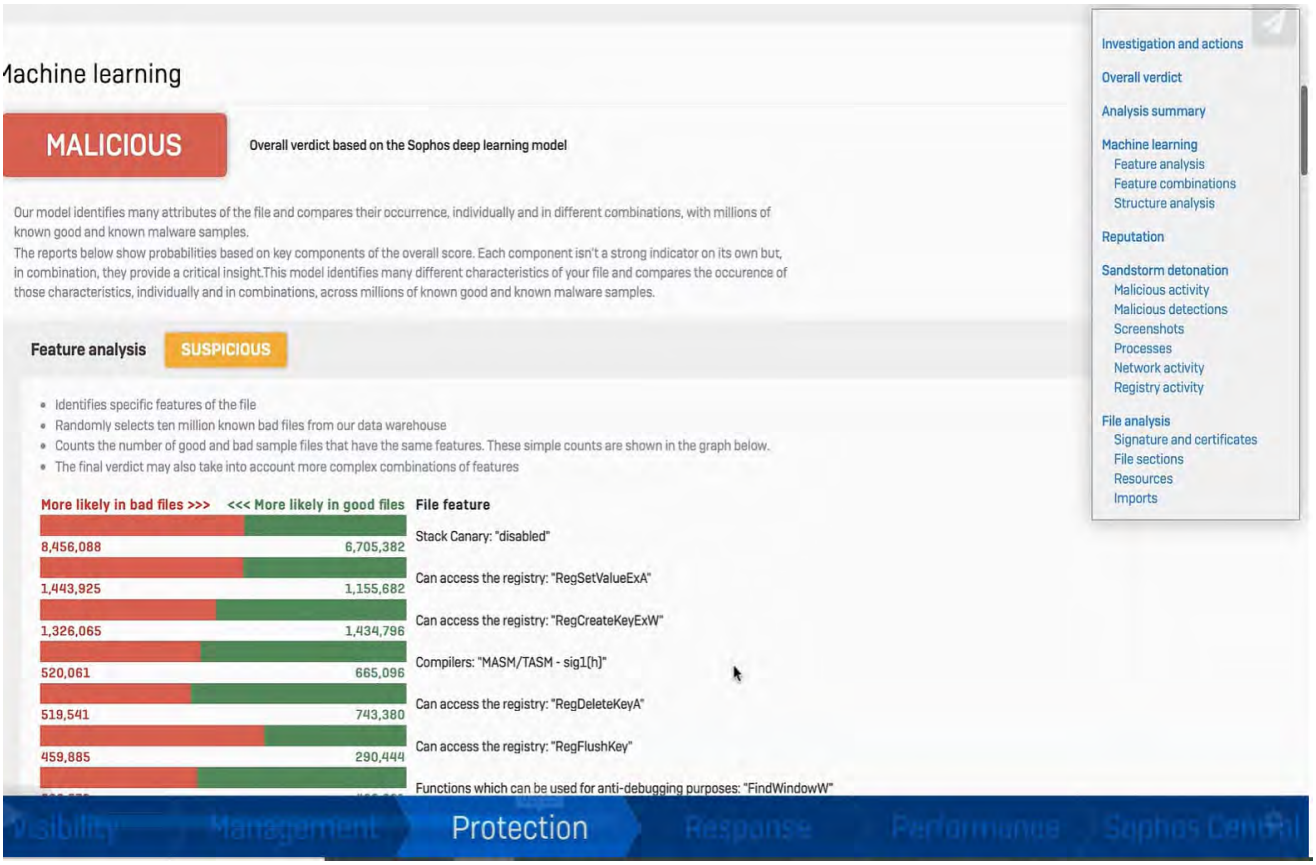
(*See* https://techvids.sophos.com/watch/wm84yg3wcZtYB1sZjwYyKt.)



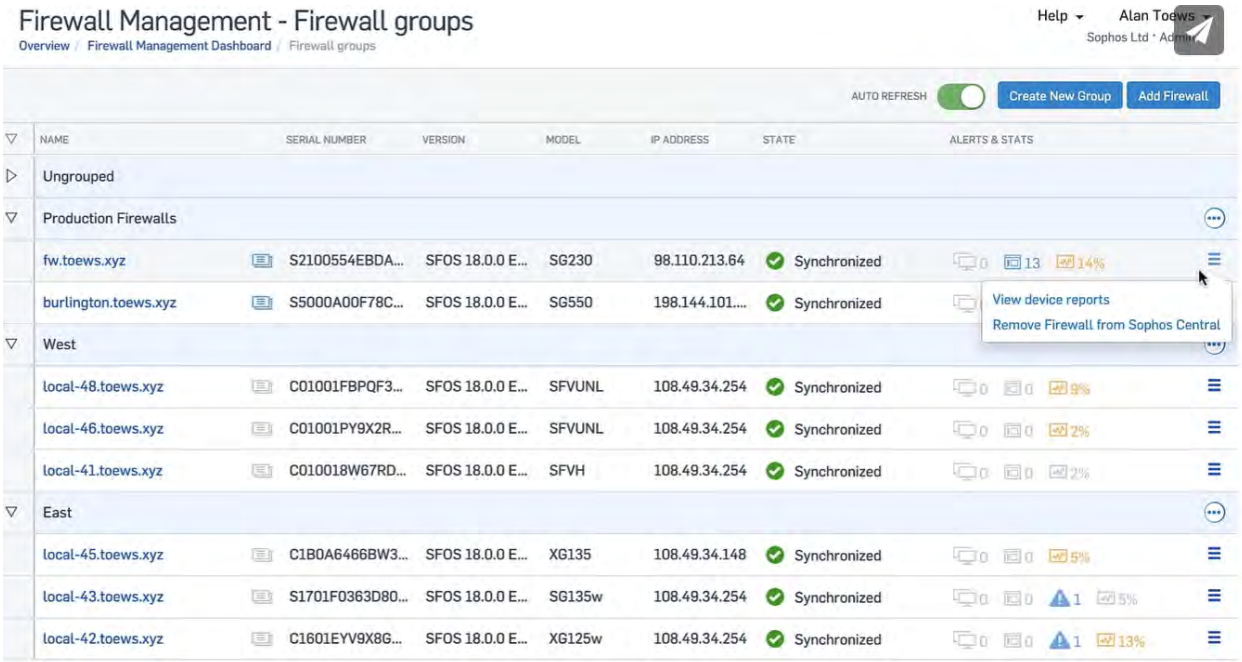
(*See* https://community.sophos.com/sophos-xg-firewall/f/recommended-reads/122357/life-of-a-packet-sophos-firewall#mcetoc_1fc8lebu84.)



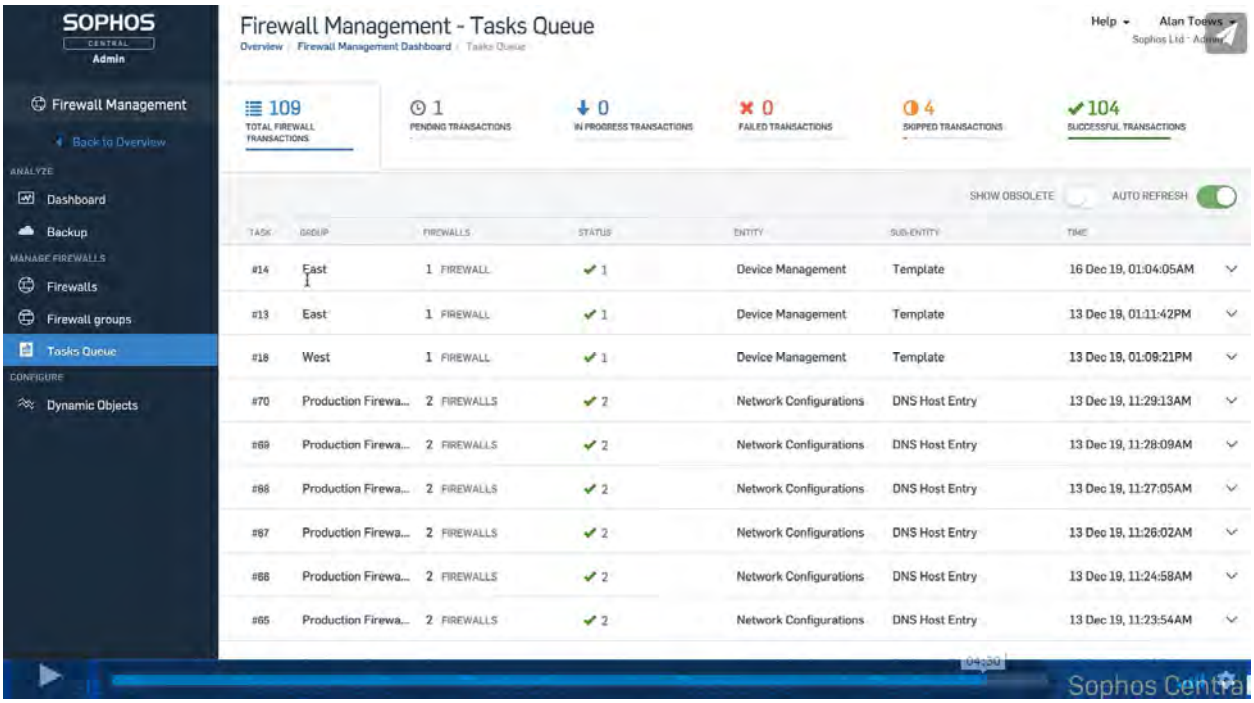
(*See* https://player.vimeo.com/video/144094496?width=800&height=450&iframe=true&portrait=0.)



(*See* https://player.vimeo.com/video/144094496?width=800&height=450&iframe=true&portrait=0.)

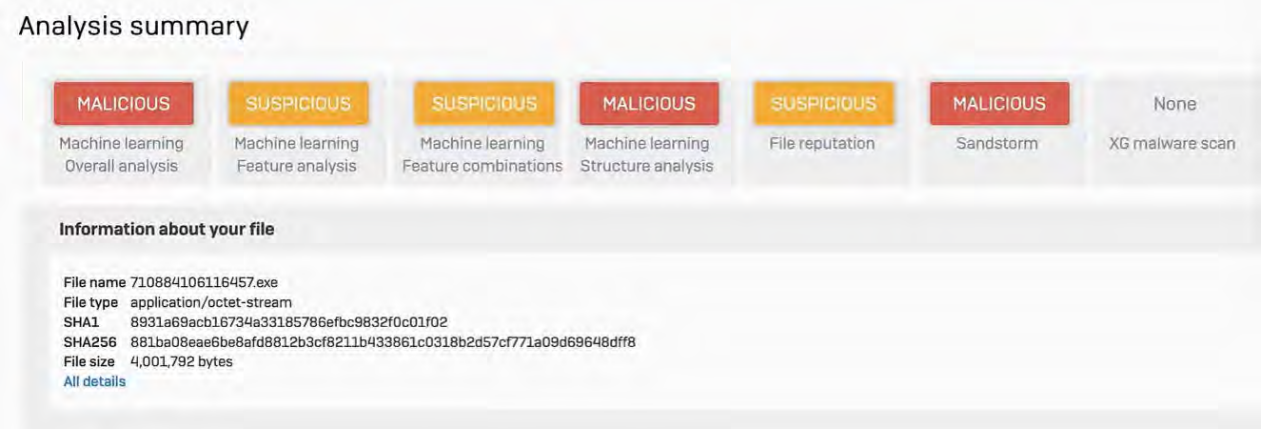


(*See* https://player.vimeo.com/video/144094496?width=800&height=450&iframe=true&portrait=0.)

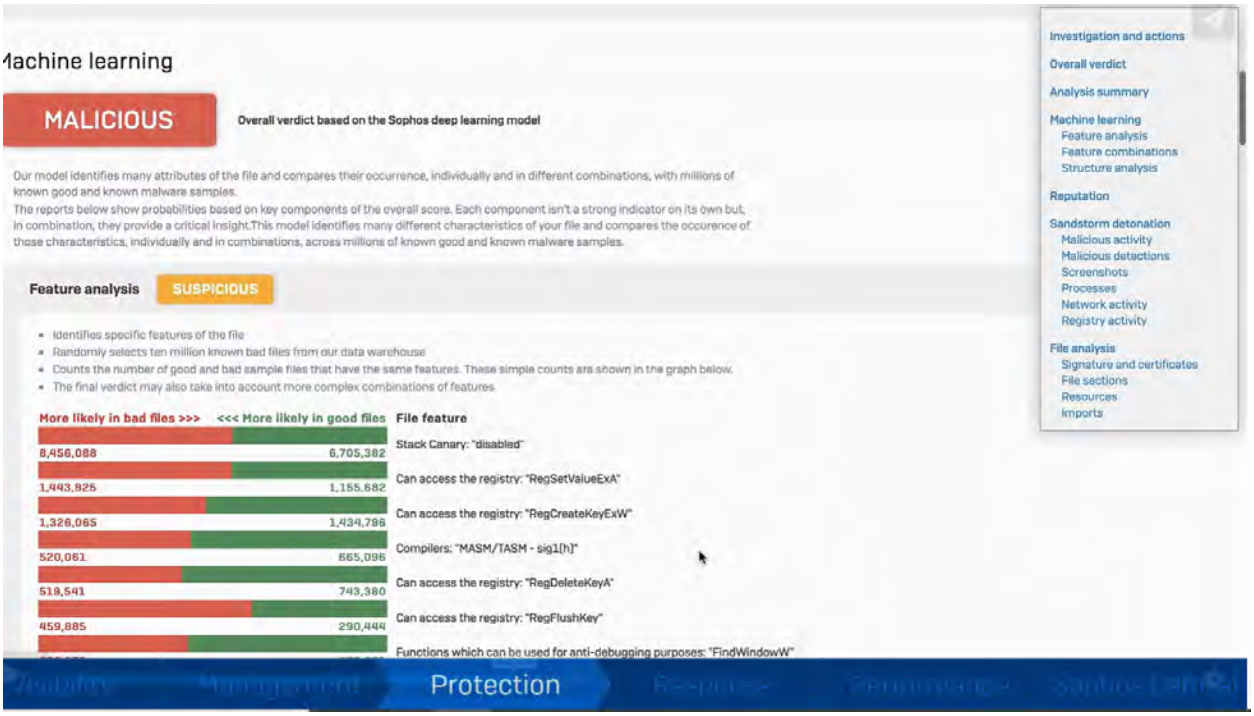


(*See* https://player.vimeo.com/video/144094496?width=800&height=450&iframe=true&portrait=0.)

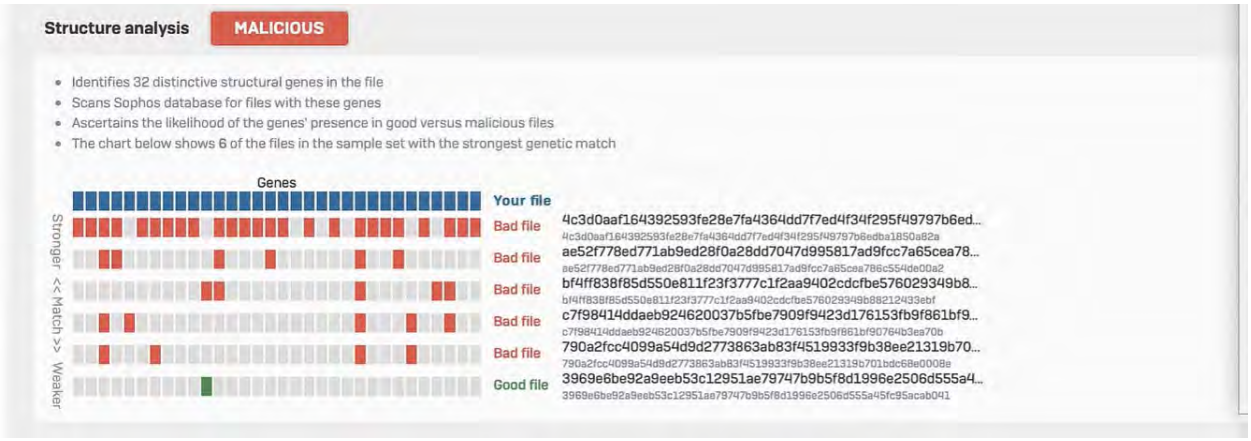
321. The Accused Products perform a method that includes *providing contents of the second database to at least one database associated with a central server, wherein the contents comprise a signature of the first computer object, behavior information about the first computer object, and information about the first remote computer*. For example, on information and belief, the ATP database associated with the Firewall that receives the data from the first remote computer communicates with a reporter database associated with Sophos' Control Center, Sophos Central. Based on the information collected and reported, Sophos Central displays information about the object's signature, behavior, and the computer on which it was observed.



(*See* https://player.vimeo.com/video/144094496?width=800&height=450&iframe=true&portrait=0.)



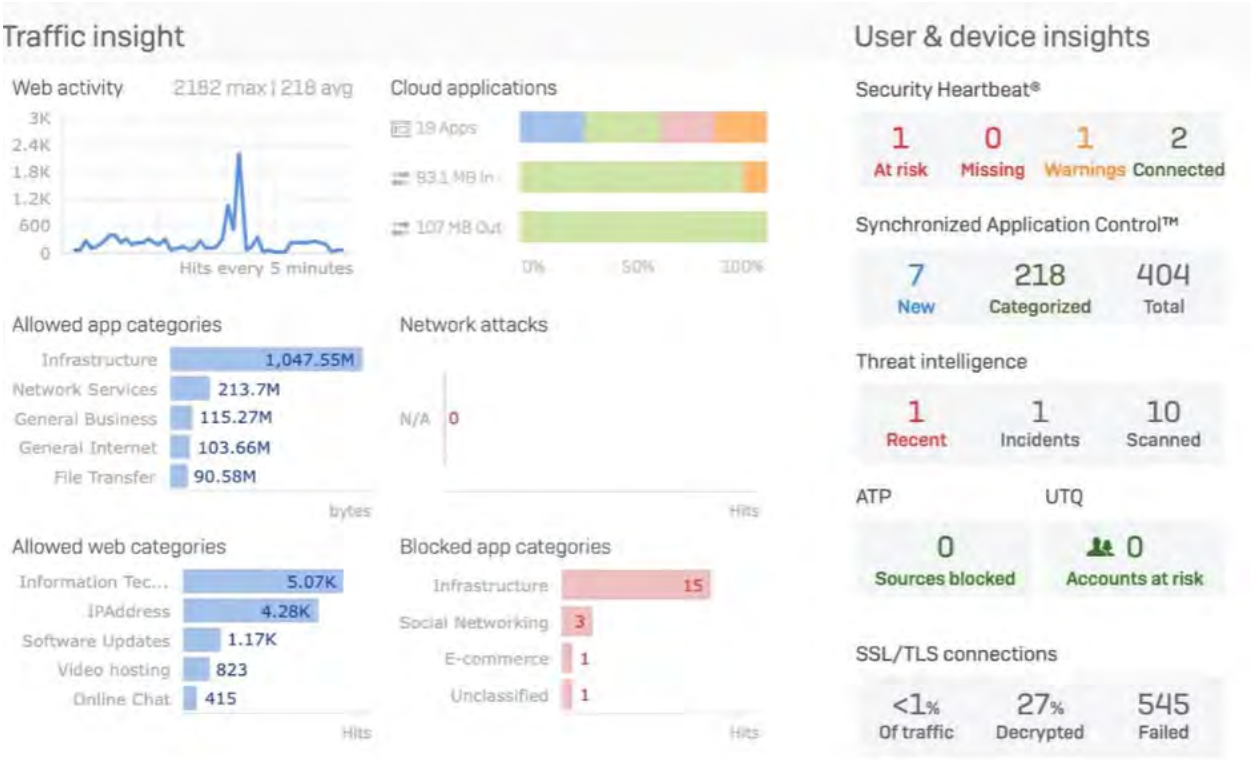
(*See* https://player.vimeo.com/video/144094496?width=800&height=450&iframe=true&portrait=0.)



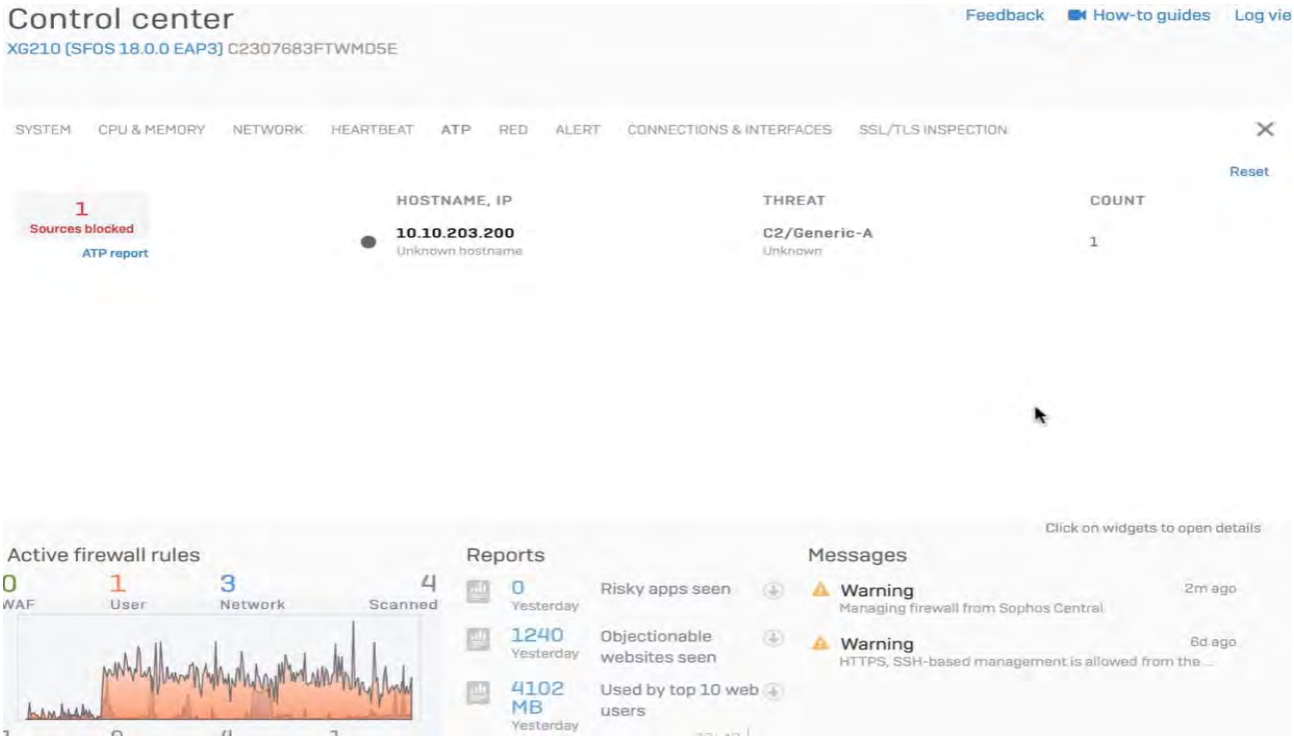
(*See* https://player.vimeo.com/video/144094496?width=800&height=450&iframe=true&portrait=0.)

322. The Accused Products perform a method that includes *increasing a count associated with a number of times that the first computer object has been seen and providing the increased count associated with the number of times that the first computer object has been seen*

to the central server. For example, Sophos Central displays a count for the number of times the object under investigation has been seen. It then updates the count based on the current observation.



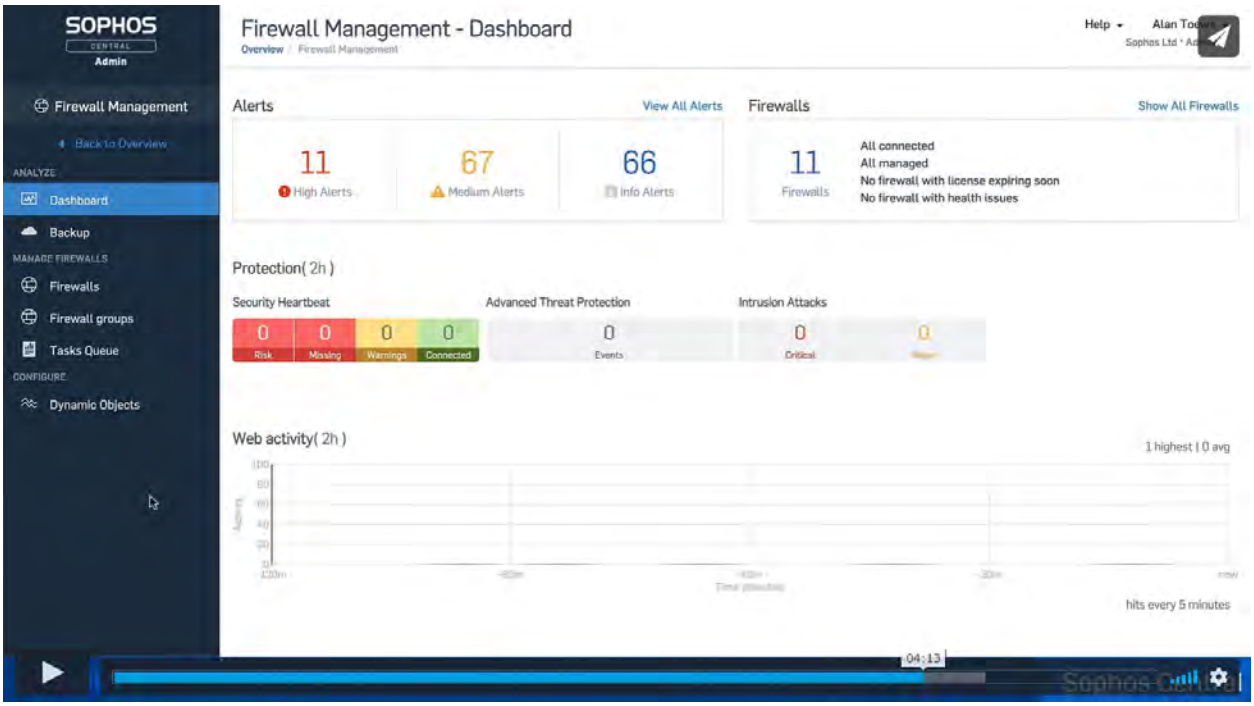
(*See* https://player.vimeo.com/video/144094496?width=800&height=450&iframe=true&portrait=0.)



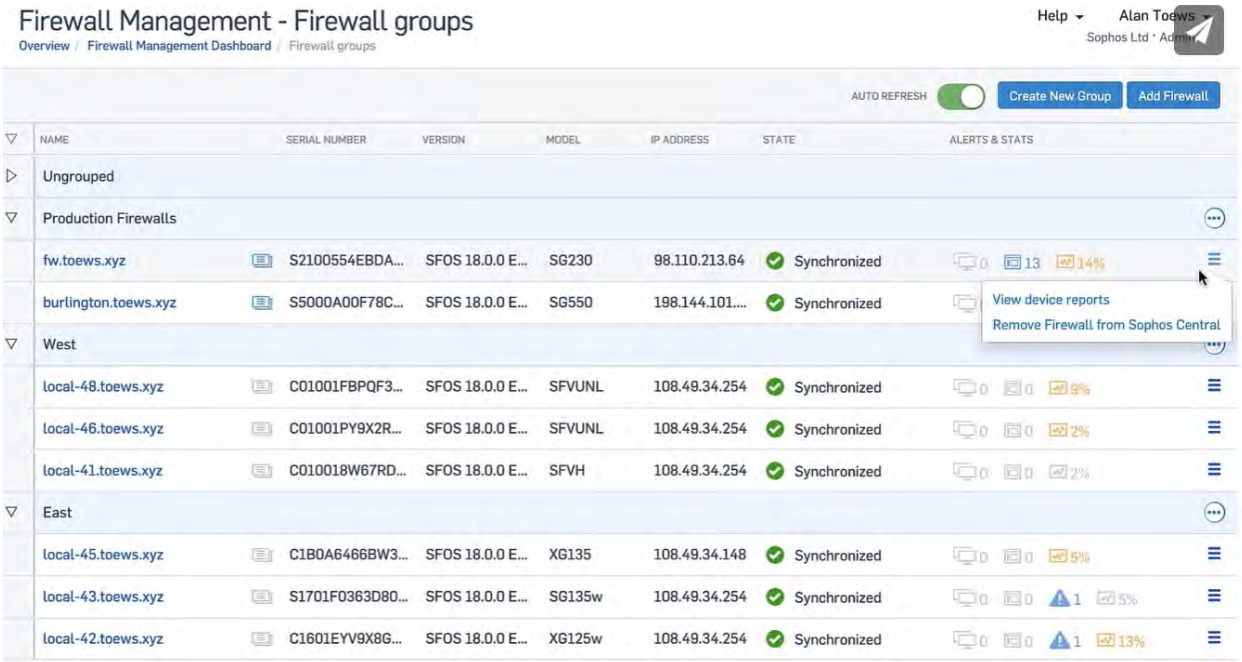
(*See* https://player.vimeo.com/video/144094496?width=800&height=450&iframe=true&portrait=0.)

323. The Accused Products perform a method that includes *receiving, at a second threat server at least a portion of the contents of the at least one database associated with the central server wherein the at least a portion of the contents of the at least one database associated with the central server include a subset of the details of the first computer object stored in the second database*. For example, based on the information received at Sophos Central from the first Sophos Firewall, Sophos Central sends at least some of the information about the computer object to each of the other ATP databases associated with each of the Sophos Firewalls (*i.e.*, other Sophos Firewalls connected to the network)—such as information that enables the object to be identified (*e.g.*, on information and belief, information that enables each Sophos Firewall to use a static

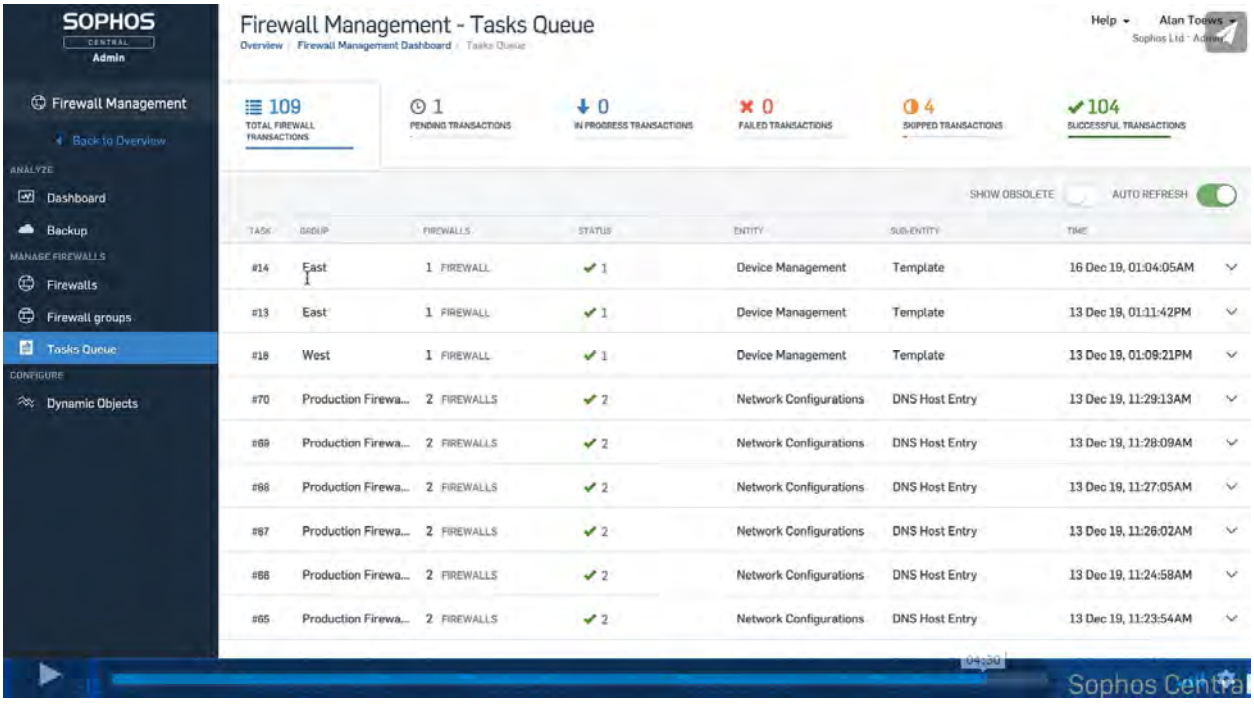
application signature to identify computer objects). The updating of the ATP databases is demonstrated by the update to the Firewall rules to look for the new object.



(*See* https://player.vimeo.com/video/144094496?width=800&height=450&iframe=true&portrait=0.)



(*See* https://player.vimeo.com/video/144094496?width=800&height=450&iframe=true&portrait=0.)



(*See* https://player.vimeo.com/video/144094496?width=800&height=450&iframe=true&portrait=0.)

324. Each claim in the '721 Patent recites an independent invention. Neither claim 1, described above, nor any other individual claim is representative of all claims in the '721 Patent.

325. Defendant has been aware of the '721 Patent since at least the filing of this Complaint. Further, Plaintiffs have marked its products with the '721 Patent, including on its web site, since at least July 2020.

326. Defendant directly infringes at least claim 1 of the '721 Patent, literally or under the doctrine of equivalents, by performing the steps described above. For example, on information and belief, Defendant performs the claimed method in an infringing manner as described above by running this software and corresponding systems to protect its own computer and network operations. On information and belief, Defendant also performs the claimed method as described above when testing the operation of the Accused Products and corresponding systems. As another example, Defendant perform each of the method steps as described above when providing or administering services to third parties, customers, and partners using the Accused Products.

327. Defendant's partners, customers, and end users of the Accused Products and corresponding systems and services directly infringe at least claim 1 of the '721 Patent, literally or under the doctrine of equivalents, at least by using the Accused Products and corresponding systems and services, as described above.

328. Defendant has actively induced and is actively inducing infringement of at least claim 1 of the '721 Patent with specific intent to induce infringement, and/or willful blindness to the possibility that its acts induce infringement, in violation of 35 U.S.C. § 271(b). For example, Defendant encourages and induces customers with specific intent to use Sophos security software

in a manner that infringes claim 1 of the ’721 Patent at least by offering and providing software that performs a method that infringes claim 1 when installed and operated by the customer, and by engaging in activities relating to selling, marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including the activities described below.

329. Defendant encourages, instructs, directs, and/or requires third parties—including its certified partners and/or customers—to perform the claimed method using the software, services, and systems in infringing ways, as described above.

330. Defendant further encourages and induces its customers to infringe claim 1 of the ’721 Patent: 1) by making its security services available on its website, providing applications that allow users to access those services, widely advertising those services, and providing technical support and instructions to users, and 2) through activities relating to marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including Sophos security software, and services in the United States. (*See* https://www.sophos.com/en-us/products/endpoint-antivirus/how-to-buy.aspx; https://partners.sophos.com/english/directory/search?lat=30.267153&lng=-97.7430608&dMI=100&p=1; https://secure2.sophos.com/en-us/security-news-trends/whitepapers/gated-wp/cybersecurity-system-buyers-guide.aspx%23formFrame; *see also* https://www.sophos.com/en-us/medialibrary/PDFs/factsheets/sophos-synchronized-security-ds.pdf.)

331. For example, on information and belief, Defendant shares instructions, guides, and manuals, which advertise and instruct third parties on how to use the software as described above, including to at least customers and partners. (*Id.*) On further information and belief, Defendant also provides customer service and technical support to purchasers of the Accused Products and corresponding system and services, which directs and encourages customers to perform certain

actions that use the Accused Products in an infringing manner. (*Id.*)

332. Defendant and/or its partners recommend and sell the Accused Products and provide technical support for the installation, implementation, integration, and ongoing operation of the Accused Products for each individual customer. On information and belief, each customer enters into a contractual relationship with Defendant and/or one of its partners, which obligates each customer to perform certain actions in order to use the Accused Products. Further, in order to receive the benefit of Defendant's or its partner's continued technical support and their specialized knowledge and guidance of the operability of the Accused Products, each customer must continue to use the Accused Products in a way that infringes the '721 Patent.

333. Further, as the entity that provides installation, implementation, and integration of the Accused Products in addition to ensuring the Accused Product remains operational for each customer through ongoing technical support, on information and belief, Defendant and/or its partners affirmatively aid and abet each customer's use of the Accused Products in a manner that performs the claimed method of, and infringes, the '721 Patent.

334. Defendant also contributes to the infringement of its partners, customers, and end-users of the Accused Products by providing within the United States or importing the Accused Products into the United States, which are for use in practicing, and under normal operation practice, methods claimed in the Asserted Patents, constituting a material part of the claimed methods, and not a staple article or commodity of commerce suitable for substantial non-infringing uses. Indeed, as shown above, the Accused Products and the example functionality described below have no substantial non-infringing uses but are specifically designed to practice the '721 Patent.

335. Indeed, as shown above, the Accused Products have no substantial non-infringing

uses because the accused functionality, including the Sophos database and server architecture and related functionality described above , is an integral part of the Accused Products and must be performed for the Accused Products to perform their intended purpose. For example, without compiling and correlating threat information about an assessed object from dynamically updated threat databases, the Accused Products could not assess whether to allow those objects to pass through firewalls or subject them to further analysis, triage and final designation as malicious or benign. These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer suitably function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice the ’721 patent, that functionality could not be performed.

336. Additionally, the accused functionality, including the Sophos database and server architecture and related functionality described above, itself has no substantial non-infringing uses because the components, modules and methods identified above are a necessary part of that functionality. For example, without leveraging information from multiple dynamically updated threat databases, the Accused Products could not propagate the latest intelligence about objects to their firewalls, and therefore could not enable their firewalls to make accurate decisions about the malicious or benign nature of assessed objects. These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice the ’721 patent, that functionality could not be performed.

337. In addition, as shown in the detailed analysis above, the products, systems, modules, and methods provided by Defendant constitute a material part of the invention—indeed, they provide all the components, modules, and features that perform the claimed methods and systems. For example, the Accused Products and accused functionalities (including Sophos database and server architecture) constitute a material part of the inventions claimed at least because such functionality is integral to the processes identified above (such as "receiving, at a first threat server, details of a first computer object . . ."; "determining, by the first threat server, whether the first computer object has been previously seen . . ."; "receiving additional information about the first computer object . . ."; "storing the details of the first computer object . . . "; "providing contents of the second database . . ."; "increasing a count associated with a number of times that the first computer object has been seen . . ."; "receiving, at a second threat server, at least a portion of the contents of the at least one database associated with the central server . . . ."), as recited in the claims of the '721 patent. None of these products are staple goods—they are sophisticated and customized cyber security and malware detection products, methods, and systems.

338. Defendant's infringing actions have continued after the original Complaint was filed. Defendant had knowledge of the '721 patent and of the specific acts that constitute infringement of the '721 patent at least based on the original Complaint, but has continued to engage in its infringing activity, including by selling, making, configuring, and installing the Accused Products and performing the accused functionality, and by engaging in activities that constitute inducing infringement and contributory infringement as described above.

339. On information and belief, the infringing actions of each partner, customer, and/or end-user of the Accused Products are attributable to Defendant. For example, on information and

belief, Defendant directs and controls the activities or actions of its partners or others in connection with the Accused Products by contractual agreement or otherwise requiring partners or others to provide information and instructions to customers who acquire the Accused Products which, when followed, results in infringement. Defendant further directs and controls the operation of devices executing the Accused Products by programming the software which, when executed by a customer or end user, perform the method steps of at least claim 1 of the '721 Patent.

340. Plaintiffs have suffered and continue to suffer damages, including lost profits, as a result of Defendant's infringement of the '721 Patent. Defendant is therefore liable to Plaintiffs under 35 U.S.C. § 284 for damages in an amount that adequately compensates Plaintiffs for Defendant's infringement, but no less than a reasonable royalty.

341. Plaintiffs will continue to suffer irreparable harm unless this Court preliminarily and permanently enjoins Defendant, its agents, employees, representatives, and all others acting in concert with Defendant from infringing the '721 Patent.

342. Defendant's infringement of the '721 Patent is knowing and willful. Defendant had actual knowledge of the '721 Patent at least by the time Plaintiffs filed this lawsuit and had constructive knowledge of the '721 Patent from at least the date Plaintiffs marked their products with the '721 Patent and/or provided notice of the '721 Patent on their website.

343. On information and belief, despite Defendant's knowledge of the Asserted Patents and Plaintiffs' patented technology, Defendant made the deliberate decision to sell products and services that they knew infringe the patents. Defendant's continued infringement of the '721 Patent with knowledge of the '721 Patent constitutes willful infringement.

344. Plaintiffs' allegations of infringement, indirect infringement, and willful infringement with respect to this patent are further set forth in Plaintiffs' Disclosure of Asserted

Claims and Preliminary Infringement Contentions Pursuant to the Court's Standing Order Governing Patent Proceedings served July 12, 2022.

EIGHTH CAUSE OF ACTION
(INFRINGEMENT OF THE '869 PATENT)

345. Plaintiffs reallege and incorporate by reference the allegations of the preceding paragraphs of this Complaint.

346. Defendant has infringed and continues to infringe one or more claims of the '869 Patent in violation of 35 U.S.C. § 271 in this District and elsewhere in the United States and will continue to do so unless enjoined by this Court. The Accused Products, including features such as Intercept X Advanced with EDR ("Intercept X"), at least when used for their ordinary and customary purposes, practice each element of at least claim 46 of the '869 Patent, as described below.

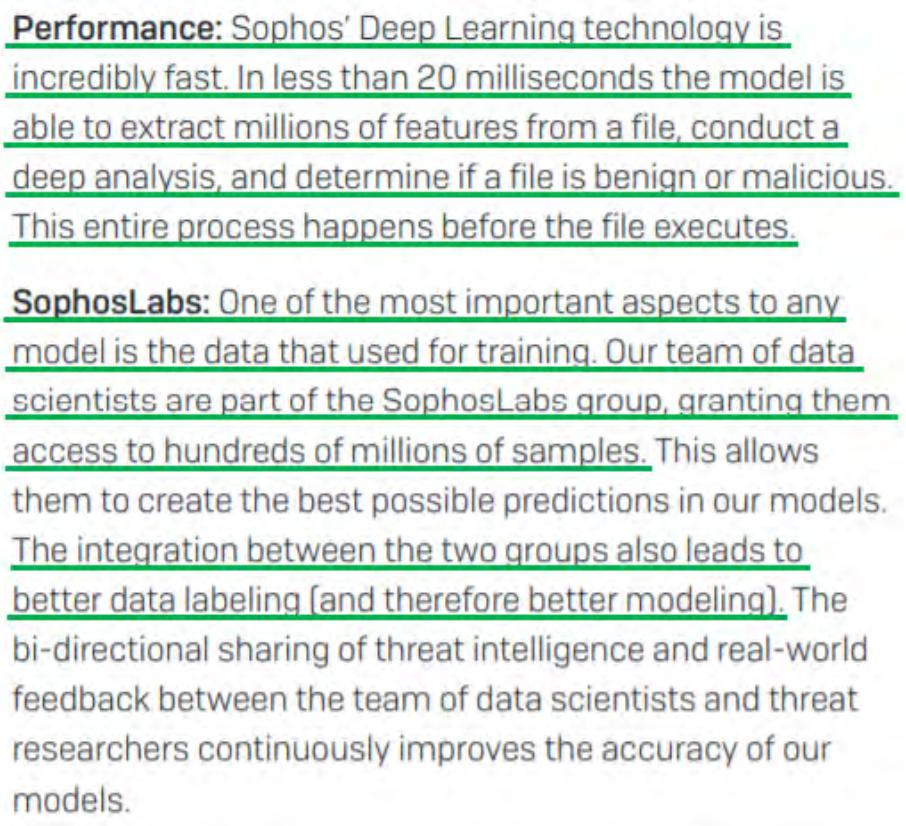
347. Claim 46 of the '869 Patent recites:

> 46. A system comprising:
>
> at least one memory;
>
> at least one processor configured to perform operation of:
>
> identifying static data points that may be indicative of either a harmful or benign executable file; and
>
> associating the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files; and
>
> at least one processor configured to perform operation of:
>
> identifying an executable file to be evaluated;
>
> extracting a plurality of static data points from the executable file;
>
> generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files,

> wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range; and
>
> evaluating the feature vector using a machine learning model to determine whether the executable file is harmful.

348. The Accused Products constitute a system comprising each element of claim 46 of the '869 Patent. To the extent the preamble is construed to be limiting, the Accused Products constitute *a system* of malware detection using machine learning, as further explained below.

349. The Accused Products constitute a system comprising *at least one memory and at least one processor configured to perform operation of identifying static data points that may be indicative of either a harmful or benign executable file and associating the identified static data points with one of a plurality of categories of files, the plurality of categories of files including harmful files and benign files*. For example, Intercept X employs "Deep Learning models" (i.e., advanced machine learning models) that have been trained on "hundreds of millions of samples" to classify features extracted from a file to "determine if a file is benign or malicious…before the file executes." Such samples include "Potentially Unwanted Applications" such as adware. Before using a deep learning model to classify the features extracted from a file, Intercept X creates a "vector of information" that translates those features into data that the model can "intake" and "process."



Performance: Sophos' Deep Learning technology is incredibly fast. In less than 20 milliseconds the model is able to extract millions of features from a file, conduct a deep analysis, and determine if a file is benign or malicious. This entire process happens before the file executes.

SophosLabs: One of the most important aspects to any model is the data that used for training. Our team of data scientists are part of the SophosLabs group, granting them access to hundreds of millions of samples. This allows them to create the best possible predictions in our models. The integration between the two groups also leads to better data labeling (and therefore better modeling). The bi-directional sharing of threat intelligence and real-world feedback between the team of data scientists and threat researchers continuously improves the accuracy of our models.

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/factsheets/sophos-intercept-x-deep-learning-dsna.pdf.)

Feature Engineering in Machine Learning

Before creating a machine learning model, it's important to prepare our data. Preparing the data requires translating it into a language our model can understand. This is referred to as feature engineering.

Artificial neural network models intake data as a vector of information, so simply feeding the model a URL – which is not in the language of a vector – means that the model can't process it without some manipulation. There are countless ways that samples can be translated into features, though it takes some domain knowledge to do so. Using the URL example again, one way to translate a URL into a usable language is through a combination of ngramming and hashing. Ngrams are a popular method in DNA sequencing research. For example, the results of a three-gram ngram for the URL "https://sophos.com/company/careers.aspx" would be:

['htt', 'ttp', 'tps', 'ps:', 's:/', '://', '//s', '/so', 'sop', 'oph', 'pho', 'hos', 'os:', 's.c', '.co', 'com', 'om/', 'm/c', '/co', 'com', 'omp', 'mpa', 'pan', 'any', 'ny/', 'y/c', '/ca', 'car', 'are', 'ree', 'eer', 'ers', 'rs.', 's.a', '.as', 'asp', 'spx']

Once the ngrams are calculated, we need to translate them into a numerical representation. This can be done through a hashing mechanism. We will create an n-length long vector – say 1000 – and hash each ngram using a hashing algorithm. The resulting number from the hash of a particular ngram will be the index of which we will add 1. For example, if the first ngram 'htt' results in a hash of three and our vector is five units long, the result would be [0, 0, 1, 0, 0]. We continue this process for every ngram and for every URL until we have the list of URLs completely transformed into individual n-length vectors. When using this method for our toy model, these vectors will be 1,000 units long.

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/machine-learning-how-to-build-a-better-threat-detection-model.pdf?cmp=70130000001xKqzAAE.)

2.1 MLP Model

An MLP is a class of feedforward ANN with an input layer, an output layer, and one or more hidden layers between them. Each node in one layer fully connects to every node in the following layer. Except for nodes in the input layer, each node is a neuron (or processing element) with a nonlinear activation function associated with a scalar weight which is adjusted during training. An MLP is a supervised learning algorithm that learns to identify data patterns for classification or regression. In order to carry out the learning process, one needs to extract a digital representation $\boldsymbol{x}$ of a given object or event that needs to be fed into the MLP. The learning task becomes to find a multidimensional function $\Phi(\cdot)$ between input $\boldsymbol{x}$ and target $\boldsymbol{y}$

$$\boldsymbol{y} \cong \Phi(\boldsymbol{x}) \quad [1]$$

where $\boldsymbol{x} \in \mathbb{R}^N$, a real-valued input feature vector $\boldsymbol{x} = [x_1, \dots, x_N]^T$ in an N dimensional feature space, with $(\cdot)^T$ denoting the transpose operation. Similarly, $\boldsymbol{y} \in \mathbb{R}^M$, a real-valued target classification vector $\boldsymbol{y} = [y_1, \dots, y_M]^T$ in an M dimensional classification space. In other words, an MLP learning process is to find $\Phi(\cdot)$

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophos-black-hat-2018-technicalpaper.pdf.)

Generic ML PUA detections

Potentially Unwanted Application (PUA) is a term used to describe applications that, while not malicious, are generally considered unsuitable for business networks.

A Generic ML PUA detection is generated by Sophos Intercept X's Machine Learning (ML) engine, also referred to by the specific Sophos approach Deep Learning and is designed to detect PUAs in PE (Portable Executable) files such as:

- .exe
- .sys
- .dll
- .pif
- .scr
- and many more

If a detection of this type has been received, it is because Sophos has detected a file on the computer that our Deep Learning threat model has decided is a PUA. This is a pre-execution detection meaning the file was detected before it was able to be run.

Major PUA classifications

The major PUA classifications are:

- adware
- dialer
- non-malicious spyware
- remote administration tools
- hacking tools

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophos-black-hat-2018-technicalpaper.pdf.)

350. These features include the most "significant strings found in the file," the attribute "Findcrypt," which "shows any suspicious cryptographic constants" found within the file, and the attribute "Resources," which "specifies a resource" within the file "that appears to be compressed or encrypted."

Deep Learning Malware Detection

With the new deep learning model, we are able to perform a signatureless pre-execution evaluation of any executable file and determine if it is malware, potentially unwanted software, or a legitimate application.

At Sophos we've taken a unique approach to our security machine learning capabilities: we've invested heavily in deep neural network technology over more prevalent methods that, while still dominant in the security industry, are being rapidly abandoned by the machine learning computer science community.

How does Intercept X detect malicious executable files?

Instead of performing a signature and heuristic scan as traditional antivirus does, deep neural networks are able to extract millions of features from a file and determine if it is malicious before the program executes. The deep learning model learns what to look for in the code, how adversaries evade detection, how they build their software, and how the software plans to deploy and run. This information is evaluated by a multi-stage deep learning algorithm to determine how similar the software is to malware or potentially unwanted software. Depending on the score it is classified as malicious, potentially unwanted, or legitimate. It does all of this in about 20 milliseconds with a model that is under 20MB in size.

(*See* https://secure2.sophos.com/en-us/medialibrary/Gated-Assets/white-papers/Sophos-Intercept-X-Solution-Brief.pdf.)

Machine learning analysis

Under **Machine learning analysis**, you can see full results of our analysis.

Attributes shows a comparison of the file's attributes with those in millions of known bad and known good files. This enables you to determine how suspicious each attribute is and therefore whether the file is likely to be good or bad. You may see the following attributes:

- **Imports** describes the functionality that the file uses from external DLLs.

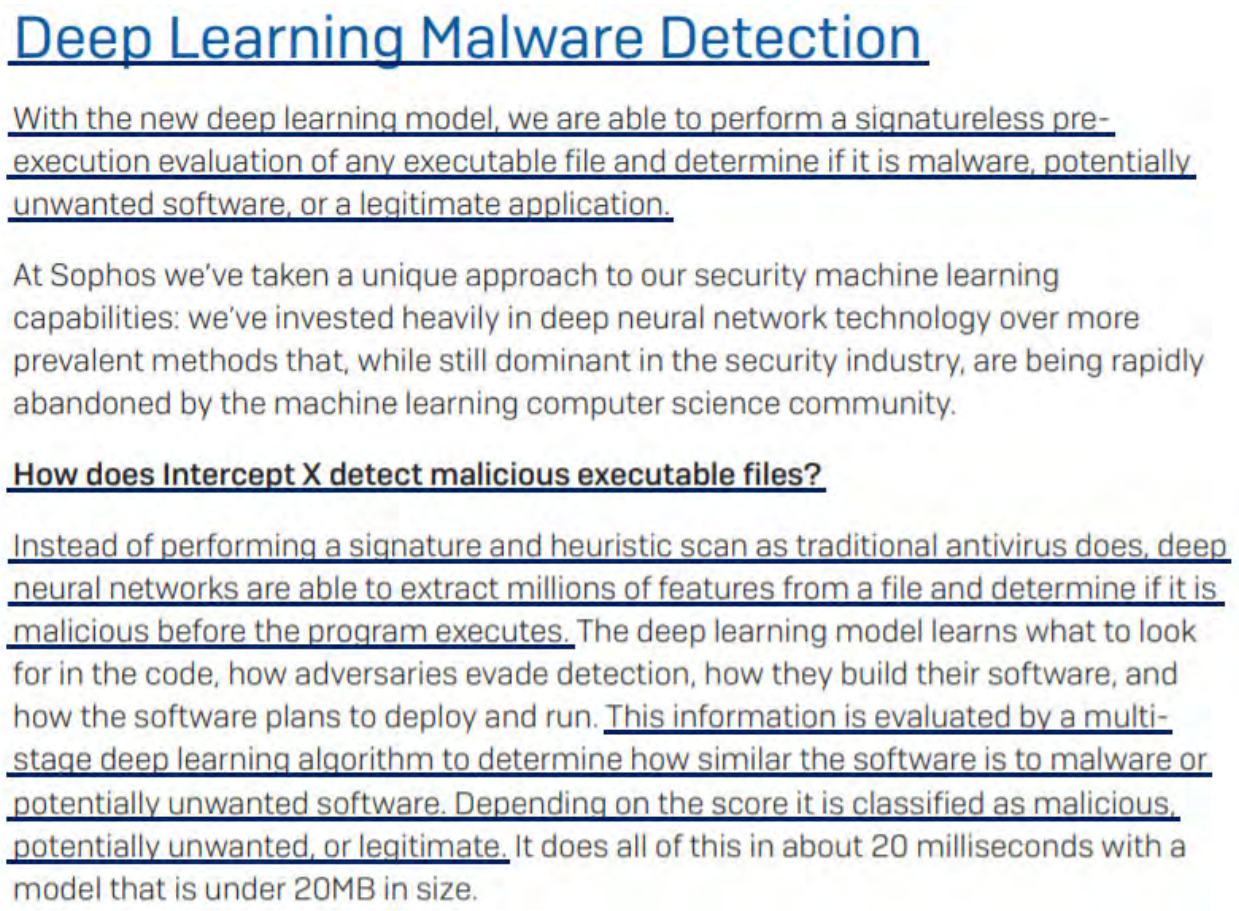
- **Strings** describes the most significant strings in the file.
- **Compilers** specifies what was used to compile the source code, for example C++, Delphi, Visual Basic, .NET.
- **Mitigation** describes techniques that the file uses to avoid being exploited.
- **Resources** specifies a resource that seems to be compressed or encrypted.
- **Summary** often relates to build or compilation dates, for example.
- **Packer** often specifies something of note about a specific section of the file, for example a suspicious section name or the fact that a section is both writable and executable.
- **Peid** refers to the output of PEiD, a third-party tool that scans a PE file against various malware signatures.
- **Btcaddress** shows any valid Bitcoin address that is found in the file.
- **Findcrypt** shows any suspicious cryptographic constants.

Code similarity shows a comparison of the file with millions of known bad and known good files, and lists the closest matches. Other matches count toward the result and may affect the rating for the file. The more bad files the file matches and the more closely it matches them, the more suspicious the file is.

File/path shows a comparison of the file's path with that of millions of known bad and known good files. If the file's path more closely matches the path of known bad files, the file is more likely to be suspicious. The path and file name used for comparison are either yours (if you requested the latest intelligence) or those from the last customer who sent us a file. We hide sensitive information in other customers' paths.

(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html.)

351. The Accused Products also comprise *at least one processor configured to perform operation of identifying an executable file to be evaluated* and *extracting a plurality of static data points from the executable file*. For example, Intercept X's "Deep Learning" technology, "extract[s] millions of features from a file and determine if it is malicious before the program executes." These extracted features include the most "significant strings found in the file," the attribute "Findcrypt," which "shows any suspicious cryptographic constants" found within the file, and the attribute "Resources," which "specifies a resource" within the file "that appears to be compressed or encrypted."



Deep Learning Malware Detection

With the new deep learning model, we are able to perform a signatureless pre-execution evaluation of any executable file and determine if it is malware, potentially unwanted software, or a legitimate application.

At Sophos we've taken a unique approach to our security machine learning capabilities: we've invested heavily in deep neural network technology over more prevalent methods that, while still dominant in the security industry, are being rapidly abandoned by the machine learning computer science community.

How does Intercept X detect malicious executable files?

Instead of performing a signature and heuristic scan as traditional antivirus does, deep neural networks are able to extract millions of features from a file and determine if it is malicious before the program executes. The deep learning model learns what to look for in the code, how adversaries evade detection, how they build their software, and how the software plans to deploy and run. This information is evaluated by a multi-stage deep learning algorithm to determine how similar the software is to malware or potentially unwanted software. Depending on the score it is classified as malicious, potentially unwanted, or legitimate. It does all of this in about 20 milliseconds with a model that is under 20MB in size.

(*See* https://secure2.sophos.com/en-us/medialibrary/Gated-Assets/white-papers/Sophos-Intercept-X-Solution-Brief.pdf.)

Machine learning analysis

Under **Machine learning analysis**, you can see full results of our analysis.

Attributes shows a comparison of the file's attributes with those in millions of known bad and known good files. This enables you to determine how suspicious each attribute is and therefore whether the file is likely to be good or bad. You may see the following attributes:

- **Imports** describes the functionality that the file uses from external DLLs.
- **Strings** describes the most significant strings in the file.
- **Compilers** specifies what was used to compile the source code, for example C++, Delphi, Visual Basic, .NET.
- **Mitigation** describes techniques that the file uses to avoid being exploited.
- **Resources** specifies a resource that seems to be compressed or encrypted.
- **Summary** often relates to build or compilation dates, for example.

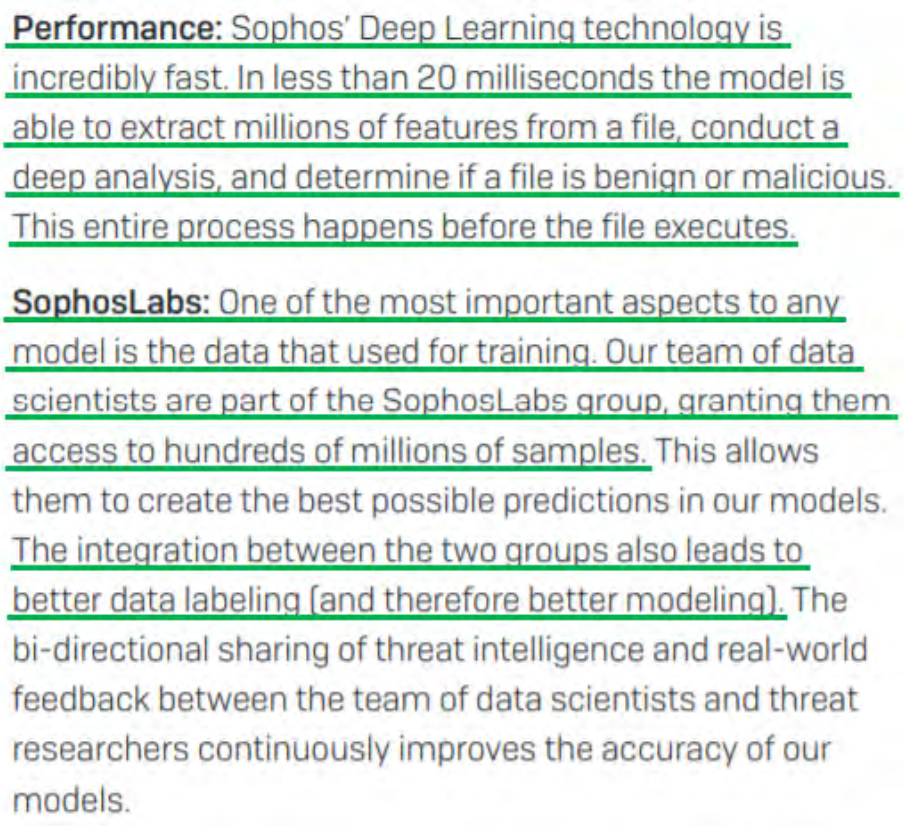
- **Packer** often specifies something of note about a specific section of the file, for example a suspicious section name or the fact that a section is both writable and executable.
- **Peid** refers to the output of PEiD, a third-party tool that scans a PE file against various malware signatures.
- **Btcaddress** shows any valid Bitcoin address that is found in the file.
- **Findcrypt** shows any suspicious cryptographic constants.

Code similarity shows a comparison of the file with millions of known bad and known good files, and lists the closest matches. Other matches count toward the result and may affect the rating for the file. The more bad files the file matches and the more closely it matches them, the more suspicious the file is.

File/path shows a comparison of the file's path with that of millions of known bad and known good files. If the file's path more closely matches the path of known bad files, the file is more likely to be suspicious. The path and file name used for comparison are either yours (if you requested the latest intelligence) or those from the last customer who sent us a file. We hide sensitive information in other customers' paths.

(*See* https://docs.sophos.com/central/Customer/help/en-us/central/Customer/concepts/ProcessDetails.html.)

352. The Accused Products also comprise a processor configured to perform operation of *generating a feature vector from the plurality of static data points using a classifier trained to classify the static data points based on training data, the training data comprising files known to fit into one of the plurality of categories of files*. For example, Intercept X employs "Deep Learning models" (i.e., advanced machine learning models) that have been trained on "hundreds of millions of samples" to classify the features extracted from a file to "determine if a file is benign or malicious…before the file executes." Such samples include "Potentially Unwanted Applications" such as adware. Before using a deep learning model to classify the features extracted from a file, Intercept X creates a "vector of information" that translates those features into data that the model can "intake" and "process."



Performance: Sophos' Deep Learning technology is incredibly fast. In less than 20 milliseconds the model is able to extract millions of features from a file, conduct a deep analysis, and determine if a file is benign or malicious. This entire process happens before the file executes.

SophosLabs: One of the most important aspects to any model is the data that used for training. Our team of data scientists are part of the SophosLabs group, granting them access to hundreds of millions of samples. This allows them to create the best possible predictions in our models. The integration between the two groups also leads to better data labeling (and therefore better modeling). The bi-directional sharing of threat intelligence and real-world feedback between the team of data scientists and threat researchers continuously improves the accuracy of our models.

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/factsheets/sophos-intercept-x-deep-learning-dsna.pdf.)

Feature Engineering in Machine Learning

Before creating a machine learning model, it's important to prepare our data. Preparing the data requires translating it into a language our model can understand. This is referred to as feature engineering.

Artificial neural network models intake data as a vector of information, so simply feeding the model a URL – which is not in the language of a vector – means that the model can't process it without some manipulation. There are countless ways that samples can be translated into features, though it takes some domain knowledge to do so. Using the URL example again, one way to translate a URL into a usable language is through a combination of ngramming and hashing. Ngrams are a popular method in DNA sequencing research. For example, the results of a three-gram ngram for the URL "https://sophos.com/company/careers.aspx" would be:

['htt', 'ttp', 'tps', 'ps:', 's:/', '://', '//s', '/so', 'sop', 'oph', 'pho', 'hos', 'os:', 's.c', '.co', 'com', 'om/', 'm/c', '/co', 'com', 'omp', 'mpa', 'pan', 'any', 'ny/', 'y/c', '/ca', 'car', 'are', 'ree', 'eer', 'ers', 'rs.', 's.a', '.as', 'asp', 'spx']

Once the ngrams are calculated, we need to translate them into a numerical representation. This can be done through a hashing mechanism. We will create an n-length long vector – say 1000 – and hash each ngram using a hashing algorithm. The resulting number from the hash of a particular ngram will be the index of which we will add 1. For example, if the first ngram 'htt' results in a hash of three and our vector is five units long, the result would be [0, 0, 1, 0, 0]. We continue this process for every ngram and for every URL until we have the list of URLs completely transformed into individual n-length vectors. When using this method for our toy model, these vectors will be 1,000 units long.

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/machine-learning-how-to-build-a-better-threat-detection-model.pdf?cmp=70130000001xKqzAAE.)

2.1 MLP Model

An MLP is a class of feedforward ANN with an input layer, an output layer, and one or more hidden layers between them. Each node in one layer fully connects to every node in the following layer. Except for nodes in the input layer, each node is a neuron (or processing element) with a nonlinear activation function associated with a scalar weight which is adjusted during training. An MLP is a supervised learning algorithm that learns to identify data patterns for classification or regression. In order to carry out the learning process, one needs to extract a digital representation $\boldsymbol{x}$ of a given object or event that needs to be fed into the MLP. The learning task becomes to find a multidimensional function $\Phi(\cdot)$ between input $\boldsymbol{x}$ and target $\boldsymbol{y}$

$$\boldsymbol{y} \cong \Phi(\boldsymbol{x}) \quad [1]$$

where $\boldsymbol{x} \in \mathbb{R}^N$, a real-valued input feature vector $\boldsymbol{x} = [x_1, \dots, x_N]^T$ in an N dimensional feature space, with $(\cdot)^T$ denoting the transpose operation. Similarly, $\boldsymbol{y} \in \mathbb{R}^M$, a real-valued target classification vector $\boldsymbol{y} = [y_1, \dots, y_M]^T$ in an M dimensional classification space. In other words, an MLP learning process is to find $\Phi(\cdot)$

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophos-black-hat-2018-technicalpaper.pdf.)

Generic ML PUA detections

Potentially Unwanted Application (PUA) is a term used to describe applications that, while not malicious, are generally considered unsuitable for business networks.

A Generic ML PUA detection is generated by Sophos Intercept X's Machine Learning (ML) engine, also referred to by the specific Sophos approach Deep Learning and is designed to detect PUAs in PE (Portable Executable) files such as:

- .exe
- .sys
- .dll
- .pif
- .scr
- and many more

If a detection of this type has been received, it is because Sophos has detected a file on the computer that our Deep Learning threat model has decided is a PUA. This is a pre-execution detection meaning the file was detected before it was able to be run.

Major PUA classifications

The major PUA classifications are:

- adware
- dialer
- non-malicious spyware
- remote administration tools
- hacking tools

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophos-black-hat-2018-technicalpaper.pdf.)

353. The feature vector is "evaluated by a multi-stage deep learning algorithm to determine how similar the software is to malware or potentially unwanted software. Depending on the score [the file] is classified as malicious, potentially unwanted, or legitimate."

Deep Learning Malware Detection

With the new deep learning model, we are able to perform a signatureless pre-execution evaluation of any executable file and determine if it is malware, potentially unwanted software, or a legitimate application.

At Sophos we've taken a unique approach to our security machine learning capabilities: we've invested heavily in deep neural network technology over more prevalent methods that, while still dominant in the security industry, are being rapidly abandoned by the machine learning computer science community.

How does Intercept X detect malicious executable files?

Instead of performing a signature and heuristic scan as traditional antivirus does, deep neural networks are able to extract millions of features from a file and determine if it is malicious before the program executes. The deep learning model learns what to look for in the code, how adversaries evade detection, how they build their software, and how the software plans to deploy and run. This information is evaluated by a multi-stage deep learning algorithm to determine how similar the software is to malware or potentially unwanted software. Depending on the score it is classified as malicious, potentially unwanted, or legitimate. It does all of this in about 20 milliseconds with a model that is under 20MB in size.

(*See* https://secure2.sophos.com/en-us/medialibrary/Gated-Assets/white-papers/Sophos-Intercept-X-Solution-Brief.pdf.)

354. The Accused Products generate a feature vector wherein one or more features of the feature vector are selectively turned on or off based at least in part on evaluation of whether a value of one of the plurality of static data points is within a predetermined range. For example, Intercept X employs "feature selection" to only "keep relevant features before feeding them into" its deep learning models, "identifying and removing as much noisy and redundant information as possible from extracted features."

> A crucial step in an ML workflow is feature extraction, which can be hand-crafted-based on human expertise, or automatically learned by training modern deep learning models such as convolutional neural networks (CNNs). It is natural to believe that more extracted features can provide better characterization of a learning task and more discriminating power. However, increasing the dimension of the feature vector could result in feature redundancy and noise. Redundant and irrelevant features can cause performance deterioration of an ML model with overfitting and generalization problems. Additionally, excessively increased number of features could significantly slow down a learning process. Therefore, it is of fundamental importance to only keep relevant features before feeding them into an ML model, which leads to requiring feature selection (or feature dimensionality reduction). Feature selection can be seen as the process of identifying and removing as much noisy and redundant information as possible from extracted features.

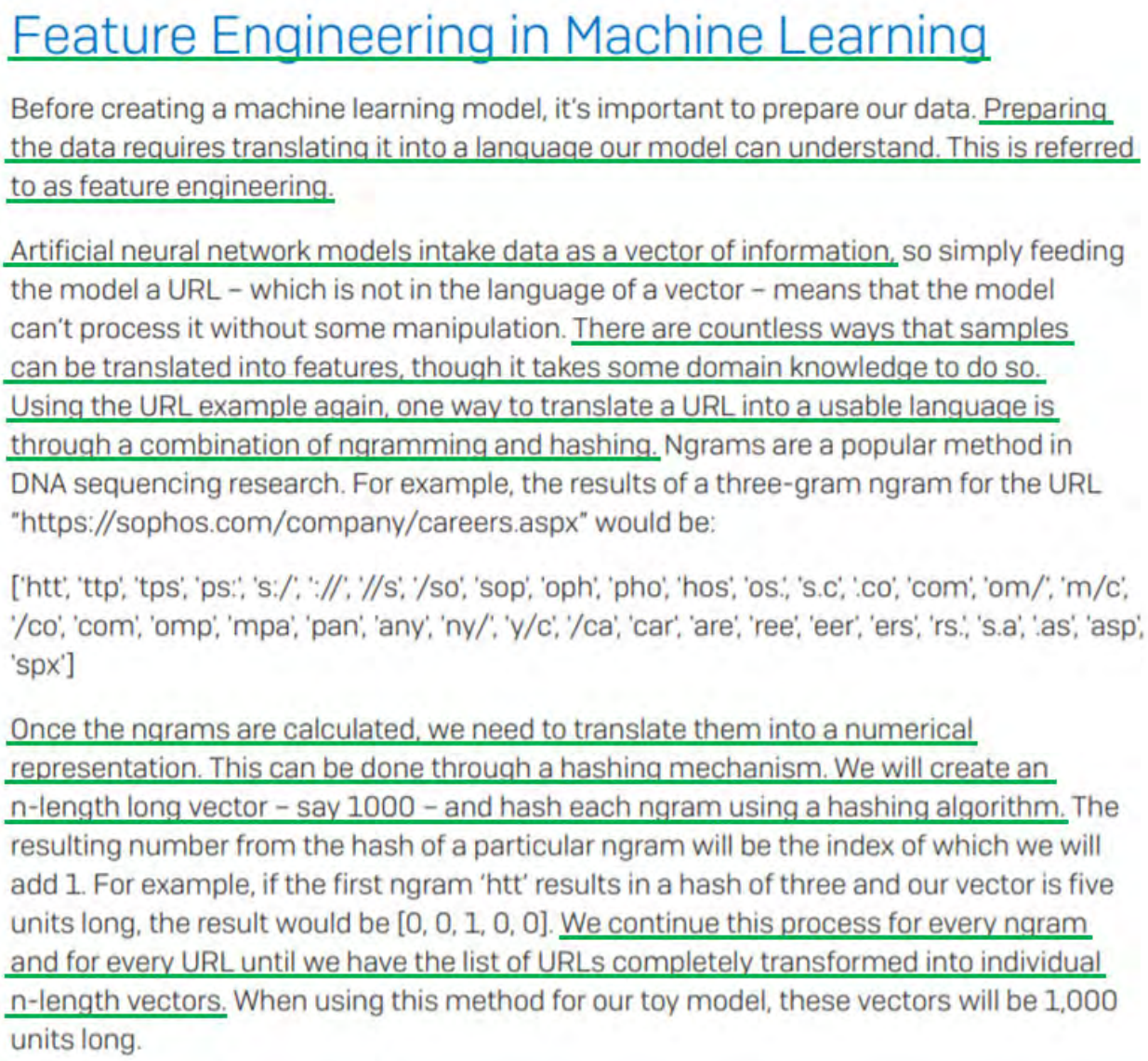
(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophoslabs-machine-learning-tp.pdf.)

355. The Accused Products constitute at least one processor configured to perform the operation of *evaluating the feature vector using a machine learning model to determine whether the executable file is harmful*. For example, and as explained above, Intercept X employs "Deep Learning models" (i.e., advanced machine learning models) that have been trained on "hundreds of millions of samples" to classify the features extracted from a file to "determine if a file is benign or malicious…before the file executes." Before using a deep learning model to classify the features extracted from a file, Intercept X creates a "vector of information" that translates those features into data that the model can "intake" and "process."

Performance: Sophos' Deep Learning technology is incredibly fast. In less than 20 milliseconds the model is able to extract millions of features from a file, conduct a deep analysis, and determine if a file is benign or malicious. This entire process happens before the file executes.

SophosLabs: One of the most important aspects to any model is the data that used for training. Our team of data scientists are part of the SophosLabs group, granting them access to hundreds of millions of samples. This allows them to create the best possible predictions in our models. The integration between the two groups also leads to better data labeling (and therefore better modeling). The bi-directional sharing of threat intelligence and real-world feedback between the team of data scientists and threat researchers continuously improves the accuracy of our models.

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/factsheets/sophos-intercept-x-deep-learning-dsna.pdf.)



Feature Engineering in Machine Learning

Before creating a machine learning model, it's important to prepare our data. Preparing the data requires translating it into a language our model can understand. This is referred to as feature engineering.

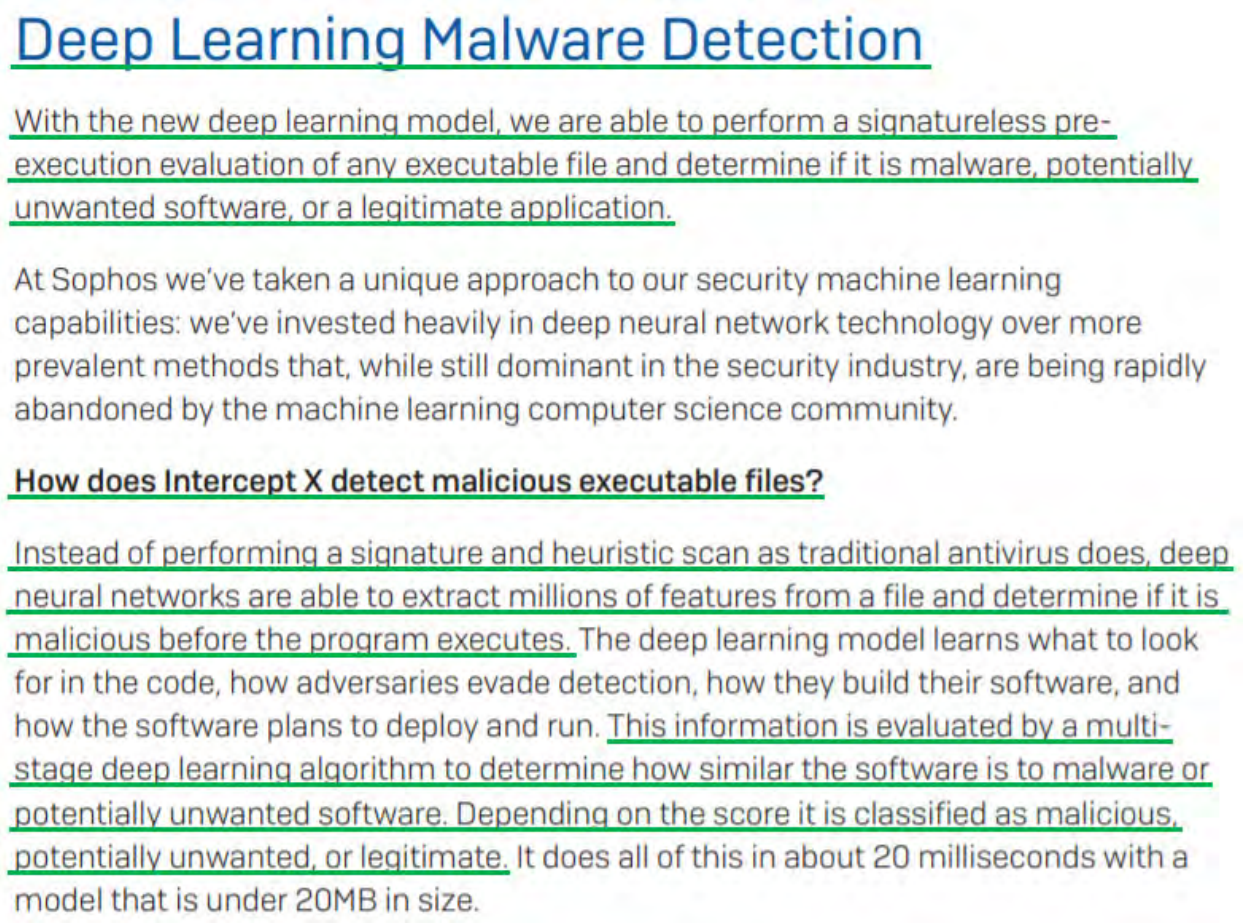
Artificial neural network models intake data as a vector of information, so simply feeding the model a URL – which is not in the language of a vector – means that the model can't process it without some manipulation. There are countless ways that samples can be translated into features, though it takes some domain knowledge to do so. Using the URL example again, one way to translate a URL into a usable language is through a combination of ngramming and hashing. Ngrams are a popular method in DNA sequencing research. For example, the results of a three-gram ngram for the URL "https://sophos.com/company/careers.aspx" would be:

['htt', 'ttp', 'tps', 'ps:', 's:/', '://', '//s', '/so', 'sop', 'oph', 'pho', 'hos', 'os:', 's.c', '.co', 'com', 'om/', 'm/c', '/co', 'com', 'omp', 'mpa', 'pan', 'any', 'ny/', 'y/c', '/ca', 'car', 'are', 'ree', 'eer', 'ers', 'rs.', 's.a', '.as', 'asp', 'spx']

Once the ngrams are calculated, we need to translate them into a numerical representation. This can be done through a hashing mechanism. We will create an n-length long vector – say 1000 – and hash each ngram using a hashing algorithm. The resulting number from the hash of a particular ngram will be the index of which we will add 1. For example, if the first ngram 'htt' results in a hash of three and our vector is five units long, the result would be [0, 0, 1, 0, 0]. We continue this process for every ngram and for every URL until we have the list of URLs completely transformed into individual n-length vectors. When using this method for our toy model, these vectors will be 1,000 units long.

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/machine-learning-how-to-build-a-better-threat-detection-model.pdf?cmp=70130000001xKqzAAE.)

356. The feature vector is "evaluated by a multi-stage deep learning algorithm to determine how similar the software is to malware or potentially unwanted software. Depending on the score [the file] is classified as malicious, potentially unwanted, or legitimate."



Deep Learning Malware Detection

With the new deep learning model, we are able to perform a signatureless pre-execution evaluation of any executable file and determine if it is malware, potentially unwanted software, or a legitimate application.

At Sophos we've taken a unique approach to our security machine learning capabilities: we've invested heavily in deep neural network technology over more prevalent methods that, while still dominant in the security industry, are being rapidly abandoned by the machine learning computer science community.

How does Intercept X detect malicious executable files?

Instead of performing a signature and heuristic scan as traditional antivirus does, deep neural networks are able to extract millions of features from a file and determine if it is malicious before the program executes. The deep learning model learns what to look for in the code, how adversaries evade detection, how they build their software, and how the software plans to deploy and run. This information is evaluated by a multi-stage deep learning algorithm to determine how similar the software is to malware or potentially unwanted software. Depending on the score it is classified as malicious, potentially unwanted, or legitimate. It does all of this in about 20 milliseconds with a model that is under 20MB in size.

(*See* https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophoslabs-machine-learning-tp.pdf.)

357. Each claim in the '869 Patent recites an independent invention. Neither claim 46, described above, nor any other individual claim is representative of all claims in the '869 Patent.

358. Defendant has been aware of the '869 Patent since at least the filing of this Second Amended Complaint. Further, Plaintiffs have marked its products with the '869 Patent, including on its web site, since at least July 2020.

359. Defendant directly infringes at least claim 46 of the '869 Patent, either literally or under the doctrine of equivalents, by performing the steps described above. For example, on information and belief, Defendant makes and uses the claimed system in an infringing manner as

described above by running the Sophos security software and corresponding systems to protect its own computer and network operations. On information and belief, Defendant also performs the claimed method as described above when testing the operation of the Accused Products and corresponding systems. As another example, Defendant performs the claimed method as described above when providing or administering services to third parties, customers, and partners using the Accused Products.

360. Defendant's partners, customers, and end users of the Accused Products and corresponding systems and services directly infringe at least claim 46 of the '869 Patent, literally or under the doctrine of equivalents, at least by making and using the Accused Products and corresponding systems and services, as described above.

361. Defendant has actively induced and is actively inducing infringement of at least claim 46 of the '869 Patent with specific intent to induce infringement, and/or willful blindness to the possibility that its acts induce infringement, in violation of 35 U.S.C. § 271(b). For example, Sophos encourages and induces customers with specific intent to use Sophos's security software in a manner that infringes claim 46 of the '869 Patent at least by offering and providing software that software that and/or hardware that embodies the system that infringes claim 46 when installed and operated by the customer, and by engaging in activities relating to selling, marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including the activities described below.

362. Defendant encourages, instructs, directs, and/or requires third parties—including its certified partners and/or customers to perform the claimed method using the software, services, and systems in infringing ways, as described above.

363. Defendant further encourages and induces its customers to infringe claim 46 of the

‘869 Patent: 1) by making its security services available on its website, providing applications that allow users to access those services, widely advertising those services, and providing technical support and instructions to users, and 2) through activities relating to marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including its Sophos security software, and services in the United States. (*See* https://www.sophos.com/en-us/products/endpoint-antivirus/how-to-buy.aspx; https://partners.sophos.com/english/directory/search?lat=30.267153&lng=-97.7430608&dMI=100&p=1; https://secure2.sophos.com/en-us/security-news-trends/whitepapers/gated-wp/cybersecurity-system-buyers-guide.aspx%23formFrame; *see also* https://www.sophos.com/en-us/medialibrary/PDFs/factsheets/sophos-synchronized-security-ds.pdf.)

364. For example, on information and belief, Defendant shares instructions, guides, and manuals, which advertise and instruct third parties on how to use the software as described above, including at least customers and partners. (*Id*.) On further information and belief, Defendant provides customer service and technical support to purchasers of the Accused Products and corresponding system and services, which directs and encourages customers to perform certain actions that use the Accused Products in an infringing manner. (*Id*.)

365. Defendant and/or its partners recommend and sell the Accused Products and provide technical support for the installation, implementation, integration, and ongoing operation of the Accused Products for each individual customer. On information and belief, each customer enters into a contractual relationship with Defendant and/or one of its partners, which obligates each customer to perform certain actions in order to use the Accused Products. Further, in order to receive the benefit of Defendant’s or its partner’s continued technical support and their specialized knowledge and guidance of the operability of the Accused Products, each customer must continue

to use the Accused Products in a way that infringes the ‘869 Patent.

366. Further, as the entity that provides installation, implementation, and integration of the Accused Products in addition to ensuring the Accused Product remains operational for each customer through ongoing technical support, on information and belief, Defendant and/or Sophos’s partners affirmatively aid and abet each customer’s use of the Accused Products in a manner that infringes the ‘869 Patent.

367. Defendant also contributes to the infringement of its partners, customers, and end-users of the Accused Products by providing within the United States or importing the Accused Products into the United States, which are for use in practicing, and under normal operation practice, methods claimed in the Asserted Patents, constituting a material part of the claimed methods, and not a staple article or commodity of commerce suitable for substantial non-infringing uses. Indeed, as shown above, the Accused Products and the example functionality described below have no substantial non-infringing uses but are specifically designed to practice the ‘869 Patent.

368. Indeed, as shown above, the Accused Products have no substantial non-infringing uses because the accused functionality, including the machine learning static anlaysis and related functionality described above, is an integral part of the Accused Products and must be performed for the Accused Products to perform their intended purpose. For example, without extracting a set of static data points from an object, the Accused Products could not create feature vector that they could feed to their machine learning model. These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer suitably function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system

and components identified above that practice the '869 patent, that functionality could not be performed.

369. Additionally, the accused functionality, including the machine learning static anlaysis and related functionality described above, itself has no substantial non-infringing uses because the components, modules and methods identified above are a necessary part of that functionality. For example, without extracting and analyzing a set of static features from an object and calibrating a feature vector on the basis of that analysis, the Accused Products could not deploy their machine learning model to determine whether or not that object is malicious or benign. These processes are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer function for its intended purpose). Moreover, for the same reasons, without performing each of the steps as described and shown above, or without the system and components identified above that practice the '869 patent, that functionality could not be performed.

370. In addition, as shown in the detailed analysis above, the products, systems, modules, and methods provided by Defendant constitute a material part of the invention—indeed, they provide all the components, modules, and features that perform the claimed methods and systems. For example, the Accused Products and accused functionality (including the machine learning static anlaysis) constitute a material part of the inventions claimed at least because such an attack chain is integral to the processes identified above (such as "extracting a plurality of static data points from an executable file . . ."; "generating a feature vector from the plurality of static data points . . ."; "wherein one or more features of the feature vector are selectively turned on or off . . ."; "evaluating the feature vector . . .") as recited in the claims of the '869 patent. None of these products are staple goods—they are sophisticated and customized cyber security and

malware detection products, methods, and systems.

371. Defendant's infringing actions have continued after the Second Amended Complaint was filed. Defendant had knowledge of the '869 patent and of the specific acts that constitutes infringement of the '869 patent at least based on the filing of the Second Amended Complaint, but has continued to engage in the infringing activity, including by selling, making, configuring, and installing the Accused Products and performing the accused functionality, and by engaging in activities that constituted infringement and contributory infringement as described above.

372. On information and belief, the infringing actions of each partner, customer, and/or end-user of the Accused Products are attributable to Defendant. For example, on information and belief, Defendant directs and controls the activities or actions of its partners or others in connection with the Accused Products by contractual agreement or otherwise requiring partners or others to provide information and instructions to customers who acquire the Accused Products which, when followed, results in infringement. Defendant further directs and controls the operation of devices executing the Accused Products by programming the software which, when executed by a customer or end user, infringes at least claim 46 of the '869 Patent.

373. Plaintiffs have suffered and continue to suffer damages, including lost profits, as a result of Defendant's infringement of the '869 Patent. Defendant is therefore liable to Plaintiffs under 35 U.S.C. § 284 for damages in an amount that adequately compensates Plaintiffs for Defendant's infringement, but no less than a reasonable royalty.

374. Plaintiffs will continue to suffer irreparable harm unless this Court preliminarily and permanently enjoins Defendant, its agents, employees, representatives, and all others acting in concert with Defendant from infringing the '869 Patent.

375. Defendant's infringement of the '869 Patent is knowing and willful. Defendant had actual knowledge of the '869 Patent at least by the time Plaintiffs filed this Second Amended Complaint.

376. On information and belief, despite Defendant's knowledge of the Asserted Patents, Defendant made the deliberate decision to sell products and services that they knew infringe the '869 Patent. Defendant's continued infringement of the '869 Patent with knowledge of the '869 Patent constitutes willful infringement.

377. Plaintiffs' allegations of infringement, indirect infringement, and willful infringement with respect to this patent are further set forth in Plaintiffs' Disclosure of Asserted Claims and Preliminary Infringement Contentions Pursuant to the Court's Standing Order Governing Patent Proceedings served July 12, 2022.

PRAYER FOR RELIEF

WHEREFORE, Plaintiffs respectfully request the following relief:

a) That this Court adjudge and decree that Defendant has been, and are currently, infringing each of the Asserted Patents;

b) That this Court award damages to Plaintiffs to compensate it for Defendant's past infringement of the Asserted Patents, through the date of trial in this action;

c) That this Court award pre- and post-judgment interest on such damages to Plaintiffs;

d) That this Court order an accounting of damages incurred by Plaintiffs from six years prior to the date this lawsuit was filed through the entry of a final, non-appealable judgment;

e) That this Court determine that this patent infringement case is exceptional pursuant

to 35 U.S.C. §§ 284 and 285 and award Plaintiffs its costs and attorneys' fees incurred in this action;

f) That this Court award increased damages under 35 U.S.C. § 284;

g) That this Court preliminarily and permanently enjoin Defendant from infringing any of the Asserted Patents;

h) That this Court order Defendant to:

(i) recall and collect from all persons and entities that have purchased any and all products found to infringe any of the Asserted Patents that were made, offered for sale, sold, or otherwise distributed in the United States by Defendant or anyone acting on its behalf;

(ii) destroy or deliver all such infringing products to Plaintiffs;

(iii) revoke all licenses to all such infringing products;

(iv) disable all web pages offering or advertising all such infringing products;

(v) destroy all other marketing materials relating to all such infringing products;

(vi) disable all applications providing access to all such infringing software; and

(vii) destroy all infringing software that exists on hosted systems,

i) That this Court, if it declines to enjoin Defendant from infringing any of the Asserted Patents, award damages for future infringement in lieu of an injunction; and

j) That this Court award such other relief as the Court deems just and proper.

DEMAND FOR JURY TRIAL

OpenText respectfully requests a trial by jury on all issues triable thereby.

DATED: August 31, 2022

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
Telephone: 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

A true and correct copy of the foregoing instrument was served or delivered electronically via the U.S. District Court ECF filing system to all counsel of record on this 31st day of August, 2022.

/s/ Jeffrey D. Mills
Jeffrey D. Mills