

Filed on behalf of: GoSecure, Inc.

Entered: December 12, 2025

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CROWDSTRIKE, INC.,
Petitioner,

v.

GOSECURE, INC.,
Patent Owner.

Case IPR2025-00070
Patent 9,954,872

PATENT OWNER'S RESPONSE

TABLE OF CONTENTS

I.	Introduction.....	1
II.	The '872 patent discloses and claims important endpoint-focused technology.....	2
III.	<i>Capalik</i> discloses a system for analyzing network intrusions based on detection of network activity.	10
IV.	Ground 1: CrowdStrike failed to show that <i>Capalik</i> renders any claim obvious.....	14
A.	Limitation 1[C]: <i>Capalik</i> does not teach or suggest storing in the memory a data structure that identifies the activity sources, the activity targets, and the associations for the plurality of activities.....	14
B.	Limitation 1[D]: <i>Capalik</i> does not teach or suggest transmitting activity information to other computer systems to prevent future attacks on those systems.....	18
1.	The Petition failed to show that <i>Capalik</i> discloses or renders obvious the transmission of 1[D].	19
2.	CrowdStrike's new theory cannot salvage the Petition's deficiencies.....	23
C.	Claim 2: CrowdStrike failed to demonstrate Grounds 1-2 render obvious "monitoring all activity on the first computer system."	43
D.	CrowdStrike failed to meet its burden for claims 3-6, 12, 15, and 20-21.....	54
V.	Ground 2: CrowdStrike failed to show that <i>Capalik-Goel</i> renders any claim obvious.....	54
A.	A POSITA would not have combined <i>Capalik</i> and <i>Goel</i>	55
VI.	Ground 3: CrowdStrike failed to show that claims 7-11, 14, 16, and 18-19 are obvious over a combination of <i>Capalik</i> , <i>Goel</i> , and <i>King</i>	59

A.	Claim 8: CrowdStrike failed to demonstrate that claim 8 is obvious.	59
B.	Claim 16: CrowdStrike failed to demonstrate that claim 16 is obvious.	61
C.	Claim 19: CrowdStrike failed to demonstrate that claim 19 is obvious.	64
VII.	Ground 4: CrowdStrike failed to show that claim 13 is obvious.	67
VIII.	Ground 5: CrowdStrike failed to meet its burden for claim 17.....	68
IX.	CrowdStrike's copying and resultant commercial success favor non-obviousness.....	68
A.	CrowdStrike principals had access to GoSecure's non-public technology.	68
B.	CrowdStrike's Falcon Platform copies the applications leading to the '872 patent.....	71
C.	CrowdStrike's industry praise and commercial success have nexus with the claimed combination of features.....	75
X.	Conclusion	76

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>AK Steel Corp. v. Sollac and Ugine</i> , 344 F.3d 1234 (Fed. Cir. 2003)	62
<i>Baran v. Med. Device Techs. Inc.</i> , 616 F.3d 1309 (Fed. Cir. 2010)	36, 37
<i>Becton, Dickinson & Co. v. Tyco Healthcare Grp., LP</i> , 616 F.3d 1249 (Fed. Cir. 2010)	28
<i>Bicon, Inc. v. Straumann Co.</i> , 441 F.3d 945 (Fed. Cir. 2006)	27, 29
<i>Geneva Pharms., Inc. v. GlaxoSmithKline PLC</i> , 349 F.3d 1373 (Fed. Cir. 2003)	39
<i>KSR Int'l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	24
<i>Laryngeal Mask Co. v. Ambu</i> , 618 F. 3d 1367 (Fed. Cir. 2010)	39
<i>Liqwd, Inc. v. L'Oreal USA, Inc.</i> , 941 F.3d 1133 (Fed. Cir. 2019)	73
<i>On Demand Mach. Corp. v. Ingram Indus., Inc.</i> , 442 F.3d 1331 (Fed. Cir. 2006)	35
<i>Perfect Surgical Techniques, Inc. v. Olympus Am., Inc.</i> , 841 F.3d 1004 (Fed. Cir. 2016)	37
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (en banc)	42
<i>SAS Inst., Inc. v. ComplementSoft, LLC</i> , 825 F.3d 1341 (Fed. Cir. 2016), <i>rev'd on other grounds</i> , 584 U.S. 357 (2018).....	30
<i>SAS Inst. Inc. v. Iancu</i> , 584 U.S. 357 (2018).....	26
<i>Scantibodies Lab'y, Inc. v. Immutopics, Inc.</i> , 374 F. App'x 968 (Fed. Cir. 2010)	40
<i>V-Formation, Inc. v. Benetton Grp. SpA</i> , 401 F.3d 1307 (Fed. Cir. 2005)	42

<i>Wasica Fin. GmbH v. Cont'l Auto. Sys., Inc.</i> , 853 F.3d 1272 (Fed. Cir. 2017)	28
--	----

Statutes

35 U.S.C. § 112(d)	62
--------------------------	----

Other Authorities

37 C.F.R. § 42.104(b)(3)	26
--------------------------------	----

Exhibit List

Exhibit	Description
2001	Declaration of Joseph Greenfield, Ph. D.
2002	<i>Curriculum vitae</i> of Joseph Greenfield, Ph. D.
2003	Cisco ISP 4200 Series Sensors Datasheet
2004	Snort and Snort Rules Explained, available at https://www.crowdstrike.com/en-us/cybersecurity-101/threat-intelligence/snort-rules/
2005	<i>Microsoft Corp. v. ThroughPuter, Inc.</i> , IPR2023-00852, Paper 18 (PTAB Nov. 9. 2023)
2006	Declaration of June Ann Munford, <i>Microsoft Corp. v. ThroughPuter, Inc.</i> , IPR2023-00852, Ex. 1012
2007	<i>SalesForce.com, Inc. v. WSOU Investments, LLC</i> , IPR2022-00357, Paper 12 (PTAB July 13, 2022)
2008	<i>Microsoft Corp. v. ThroughPuter, Inc.</i> , IPR2022-01566, Paper 13 (PTAB May 31, 2023)
2009	Cisco – Upgrading, Downgrading, and Installing System Images, available at https://www.cisco.com/en/US/docs/security/ips/5.0/configuration/guide/cli/cliimage.html#wp1091978
2010	RESERVED
2011	Univ. of Mich. Libr., Catalog-Library Search: <i>computer intrusions</i> , <a computer+intrusions\"&utm_source='lib-use-search-page"' href="https://search.lib.umich.edu/catalog?query=\">https://search.lib.umich.edu/catalog?query=\"computer+intrusions\"&utm_source=lib-use-search-page (last visited Mar. 26, 2025, 9:24 AM)

Exhibit	Description
2012	<i>PR Newswire</i> , “CounterTack Appoints Two New Board Members,” https://www.prnewswire.com/news-releases/countertack-appoints-two-new-board-members-135086178.html (last accessed June 16, 2025)
2013	Wayback Machine archive of <i>Security Battlefield</i> , George Kurtz, “CrowdStrike launches in stealth-mode with \$26 million Series A round led by Warburg Pincus,” Feb. 22, 2012, archived January 26, 2013, https://web.archive.org/web/20130126085516/http://www.georgekurtz.com/2012_02_01_archive.html (last accessed June 16, 2025)
2014	LinkedIn, <i>Jeremy Gould – Experience</i> , https://www.linkedin.com/in/jb0nz/details/experience/ (last accessed June 17, 2025)
2015	Wayback Machine archive of <i>CrowdStrike</i> , “Technology,” archived July 7, 2013, https://web.archive.org/web/20130707085243/http://www.crowdstrike.com/technology/index.html (last accessed June 20, 2025)
2016	Wayback Machine archive of <i>CrowdStrike</i> , “Next-Generation Threat Protection,” archived November 17, 2013, https://web.archive.org/web/20131127114229/http://www.crowdstrike.com/falcon-host/index.html (last accessed June 16, 2025)
2017	<i>YouTube</i> , “CrowdStrike Falcon Use Case: True Endpoint Visibility,” Oct. 13, 2015, https://www.youtube.com/watch?v=pHxb6EyjhPw (last accessed June 20, 2025)
2018	LinkedIn, <i>Dmitri Alperovitch – Experience</i> , https://www.linkedin.com/in/dmitrialperovitch/details/experience/ (last accessed June 18, 2025)

Exhibit	Description
2019	Wayback Machine archive of <i>CrowdStrike</i> , “Our Team,” archived February 26, 2014, https://web.archive.org/web/20140226192409/http://www.crowdstrike.com/our-team/index.html (last accessed June 18, 2025)
2020	Wayback Machine archive of <i>CrowdStrike</i> , “CrowdStrike Recognized on CRN’s Security 100 List,” archived April 13, 2016, https://web.archive.org/web/20160413184238/http://www.crowdstrike.com/crowdstrike-recognized-on-crns-security-100-list/ (last accessed June 20, 2025)
2021	Wayback Machine archive of <i>CrowdStrike</i> , “CrowdStrike Named 2015 Technology Pioneer by World Economic Forum,” archived April 13, 2016, https://web.archive.org/web/20160413185023/http://www.crowdstrike.com/crowdstrike-named-2015-technology-pioneer-by-world-economic-forum/ (last accessed June 20, 2025)
2022	Wayback Machine archive of <i>CrowdStrike</i> , “Resources – Case Studies,” archived March 31, 2016, https://web.archive.org/web/20160331144104/http://www.crowdstrike.com/resources#case-studies (last accessed June 20, 2025)
2023	Wayback Machine archive of <i>CrowdStrike</i> , “CrowdStrike Closes \$100 Million Financing Round Led by Google Capital,” archived March 29, 2016, https://web.archive.org/web/20160329225128/http://www.crowdstrike.com/crowdstrike-closes-100-million-financing-round-led-by-google-capital (last accessed June 20, 2025)

Exhibit	Description
2024	Wayback Machine archive of <i>CrowdStrike</i> , “Google Capital Bets Big on CrowdStrike to Accelerate Hyper-growth,” archived Dec. 20, 2015, https://web.archive.org/web/20151220114826/http://www.crowdstrike.com/blog/google-capital-bets-big-on-crowdstrike (last accessed June 20, 2025)
2025	Plaintiff’s First Set of Requests for Production to CrowdStrike (Nos. 1-59), in <i>GoSecure, Inc. v. CrowdStrike, Inc.</i> , Case No. 1:24-cv-00526-DII (W.D. Tex., Aug. 16, 2024)
2026	(Proposed) Default Protective Order
2027	Wayback Machine archive of <i>CrowdStrike</i> , “CrowdStrike™ Named 2014 SINET 16 Innovator,” archived Feb. 22, 2017, https://web.archive.org/web/20170222022251/https://www.crowdstrike.com/resources/crowdstrike-named-2014-sinet-16-innovator/ (last accessed June 20, 2025)
2028	<i>CrowdStrike, Inc. v. NSS Labs, Inc.</i> , No. 17-cv-146 (D. Del. Jan. 18, 2018), Dkt. 35 (redacted version of Second Amended Complaint)
2029	CrowdStrike Job Listing on Accel, https://jobs.accel.com/companies/crowdstrike/jobs/36416955-sr-analyst-sales-analytics-remote (last accessed October 9, 2025)
2030	CrowdStrike Falcon© Go, https://www.crowdstrike.com/en-us/pricing/falcon-go/ (last accessed October 9, 2025)
2031	Declaration of Angelos Stavrou
2032	<i>Curriculum vitae</i> of Angelos Stavrou
2033	Markus Jakobsson Deposition Transcript taken on November 11, 2025

Exhibit	Description
2034	Anne Arness, “What is EDR ? Endpoint Detection and Response (EDR)?” January 7, 2025, available at: https://www.crowdstrike.com/en-us/cybersecurity-101/endpoint-security/endpoint-detection-and-response-edr/ (last accessed November 26, 2025)
2035	CrowdStrike Holding, Inc. Initial Public Offering SEC Filing
2036	US 8,656,493 (Capalik 493)
2037	US 2008/0320594 (Jiang)
2038	US 2012/0331553 (Aziz)
2039	US 2006/0136720 (Armstrong)
2040	US 7,356,845 (Jakobsson)
2041	Bart Lenaerts-Bergmans, “What is a Backdoor Attack?” July 17, 2023, available at: https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/backdoor-attack/ (last accessed November 26, 2025)
2042	Arfan Sharif, “What is CVE? Common Vulnerabilities & Exposures,” December 7, 2022, available at: https://www.crowdstrike.com/en-us/cybersecurity-101/exposure-management/common-vulnerabilities-and-exposures-cve/ (last accessed November 26, 2025)
2043	Bart Lenaerts-Bergmans, “What is Hacktivism?” July 7, 2022, available at: https://www.crowdstrike.com/en-us/cybersecurity-101/threat-intelligence/hacktivism/ (last accessed November 26, 2025)

Exhibit	Description
2044	Thuy Nguyen, "What are Tailgating and Piggybacking Attacks?" June 25, 2025, available at: https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/tailgating-piggybacking-attack/ (last accessed November 26, 2025)
2045	Cybersecurity 101: The Fundamentals of Cybersecurity, available at: https://www.crowdstrike.com/en-us/cybersecurity-101/ (last accessed November 26, 2025)
2046	David Rojas and Mark Robinson, "Hiding in Plain Sight: Remediating "Hidden" Malware with Real Time Response," December 16, 2020, available at: https://www.crowdstrike.com/en-us/blog/how-to-remediate-hidden-malware-real-time-response/ (last accessed November 26, 2025)
2047	Kurt Baker, "Bootkit: Definition, Prevention, and Removal," January 4, 2023, available at: https://www.crowdstrike.com/en-us/cybersecurity-101/malware/bootkits/ (last accessed November 26, 2025)
2048	M. Churchill, "Parsing Sysmon Events for IR Indicators," February 23, 2015, available at: https://www.crowdstrike.com/en-us/blog/sysmon-2/ (last accessed November 26, 2025)
2049	Microsoft Computer Dictionary, p. 105 (4 th ed. 1999)
2050	Heng Yin, Zhenkai Liang and Dawn Song, "HookFinder: Identifying and Understanding Malware Hooking Behaviors," October 17, 2007, CyLab, CMU
2051	Video: Introduction to CrowdStrike, available at: https://www.youtube.com/watch?v=n2LkiwbAYCI (Jan. 25, 2013) (last accessed December 11, 2025)

Exhibit	Description
2052	Stephen T. Jones, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau, "VMM-based Hidden Process Detection and Identification using Lycosid," <i>Proceedings of the Fourth ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments</i> , 2008.
2053-2099	RESERVED
2100	CounterTack Board of Directors Meeting Slides, Feb. 27, 2012 [PROTECTIVE ORDER MATERIAL]
2101	CounterTack Internal Meeting Invitation, Nov. 23, 2011 [PROTECTIVE ORDER MATERIAL]
2102	NeuralIQ internal email re: Jeremy Gould [PROTECTIVE ORDER MATERIAL]
2103	Alperovitch email re: Jeremy Gould [PROTECTIVE ORDER MATERIAL]
2104	Gould farewell email [PROTECTIVE ORDER MATERIAL]
2105	Alperovitch farewell email [PROTECTIVE ORDER MATERIAL]
2106	Minutes of the Feb. 27, 2012 Meeting of the Board of Directors of CounterTack Inc. [PROTECTIVE ORDER MATERIAL]

I. Introduction

The '872 patent claims pioneering technology that was fundamental to a new area of cybersecurity called Endpoint Detection and Response, or EDR. According to Petitioner CrowdStrike, "EDR security solutions record the activities and events taking place on endpoints and all workloads, providing security teams with the visibility they need to uncover incidents that would otherwise remain invisible." Ex.2034. The '872 patent claims technology that enables EDR by, among other things, monitoring all activity on a computer system and transmitting information about suspicious activity to other computer systems to prevent future attacks on the other computer systems.

CrowdStrike, having been accused of patent infringement and trade secret misappropriation due to copying GoSecure's technology, asserts that the '872 patent is obvious over an earlier GoSecure patent application. But that application (*Capalik*) does not disclose the key features of the '872 patent claims that helped CrowdStrike's EDR product reshape the industry.

Instead, CrowdStrike stretches both the '872 patent and *Capalik* beyond recognition. The Petition never squarely addresses what the claims specifically require. The independent claims set out a method that (i) monitors activity on a "first computer system," (ii) identifies sources, targets, and their associations, (iii) stores a data structure that identifies, for each of a plurality of activities, the activity

sources, the activity targets, and the associations between them, and (iv) transmits information identifying those sources, targets, and associations to “one or more computer systems other than the first computer system,” to prevent future attacks on those other computer systems. Petitioner relies on *Capalik*, which uses unstructured forensic data, without ever storing a data structure that identifies the source, target, and associations of each activity as claimed, and merely transmits updates to an intrusion detection system. That mismatch infects the whole Petition, including multiple dependent claims.

The Board should find all claims patentable.

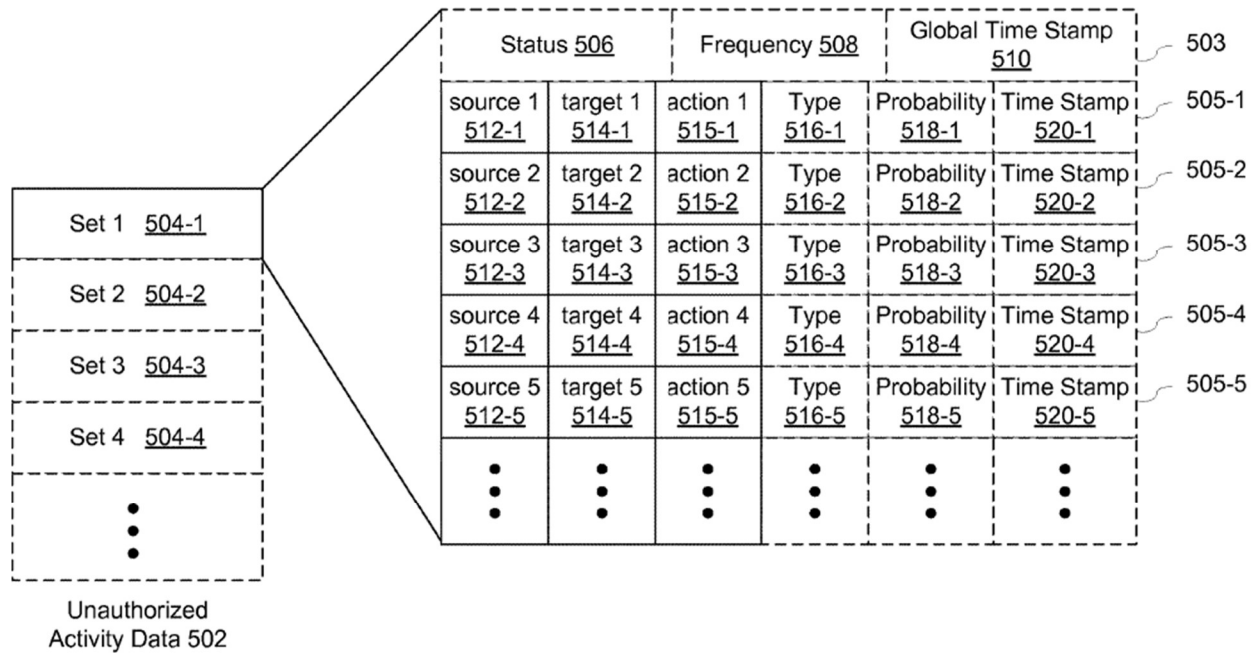
II. The '872 patent discloses and claims important endpoint-focused technology.

Intrusion detection systems, which used a library of signatures to detect unauthorized activity, were known to have significant weaknesses in detecting zero-day attacks. '872 patent 1:63-2:15. These attacks exploited previously unknown vulnerabilities, and therefore were unlikely to match an existing signature in the IDS's library. *Id.* 2:12-15. Further, IDS systems generally monitored network traffic to detect unauthorized activity. *Id.* 2:2-7.

The '872 patent discloses and claims a solution that enables a computer system to monitor significantly more activity than the prior art does, create a data structure that organizes the data in a way to show relationships for each activity, and propagate information to other computer systems for use in monitoring their own

activity. These features are claimed, for example, in claim 1, which recites “storing in the memory a data structure that identifies the activity sources, the activity targets, and the associations for the plurality of activities” identified, and transmitting the information to other computer systems for preventing future attacks to those systems—as well as claim 2, which recites “monitoring *all* activity on the first computer system.”

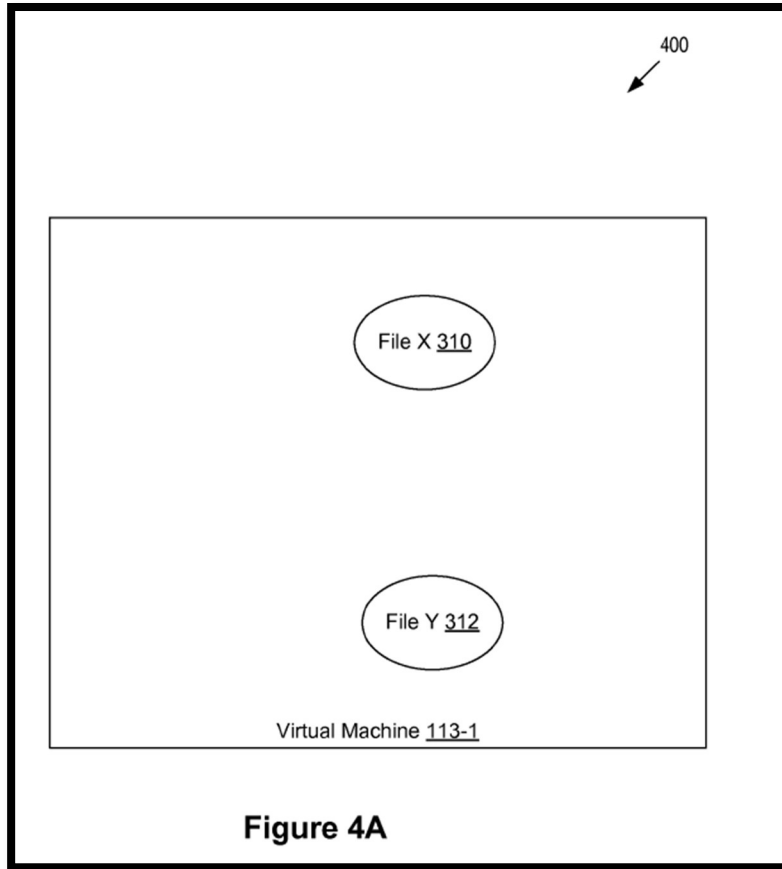
The '872 patent's data structure contains organized information not available in IDS libraries, which are limited to analyzing network traffic. Figure 5A, reproduced below, illustrates such a data structure. In this embodiment, a set of data includes a Status 506 that indicates whether the stored events are “complete” or “incomplete”—that is, whether a source of unauthorized activities is identified and whether each activity target is associated with the source through an association. '872 patent 12:58-67. This activity status is claimed in, e.g., claim 16.

**Figure 5A**

The data structure of Figure 5A also includes an association type 516 for each activity entry 505. *Id.* 13:39-56. The association type indicates whether the association is malignant or benign—i.e., whether “the activity source 512 has modified the activity target 514 to perform unauthorized activities.” *Id.* 13:43-45. This association type is claimed in claim 19.

Figures 4A-4D (reproduced below) illustrate some of the advantages of the claimed solution. As shown in Figure 4A, “unauthorized activities on the file X 310 and the file Y 312 in the virtual machine 113-1 are detected.” *Id.* at 12:2-4. Notably, detection of these unauthorized activities does not require the computer system to have been previously alerted to unauthorized network activity. For example, the disclosed system may “us[e] checksum values of the file X 310 and the file Y 312

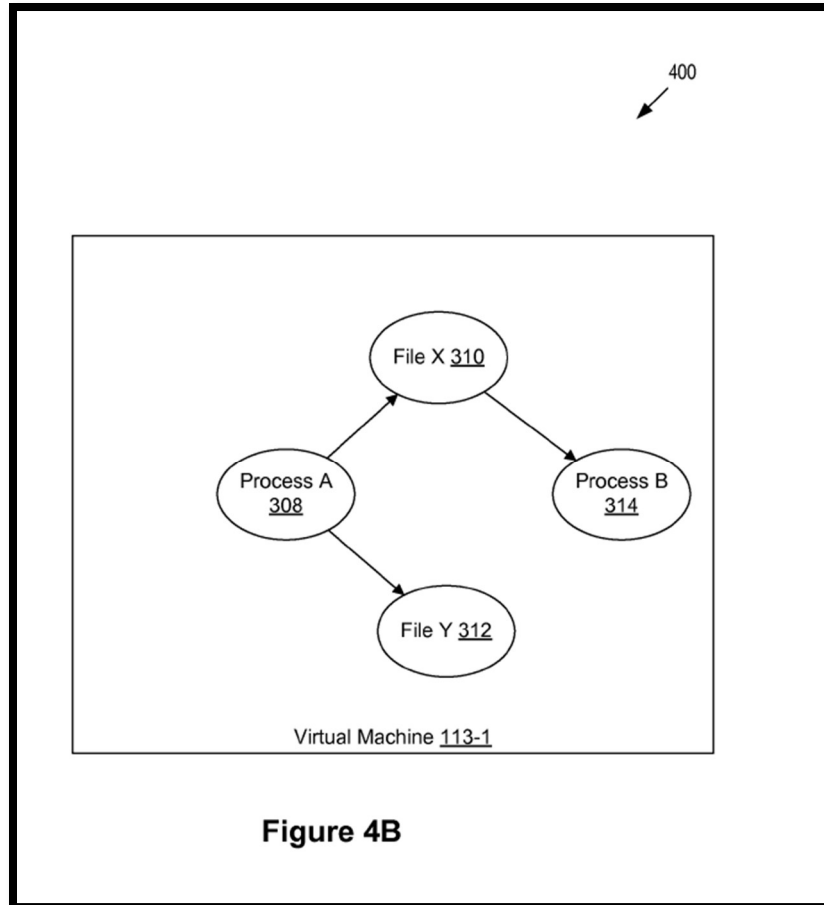
(e.g., a different checksum value can indicate that a corresponding file has been modified by an unauthorized process).” *Id.* at 12:4-8.



'872 Patent Fig. 4A

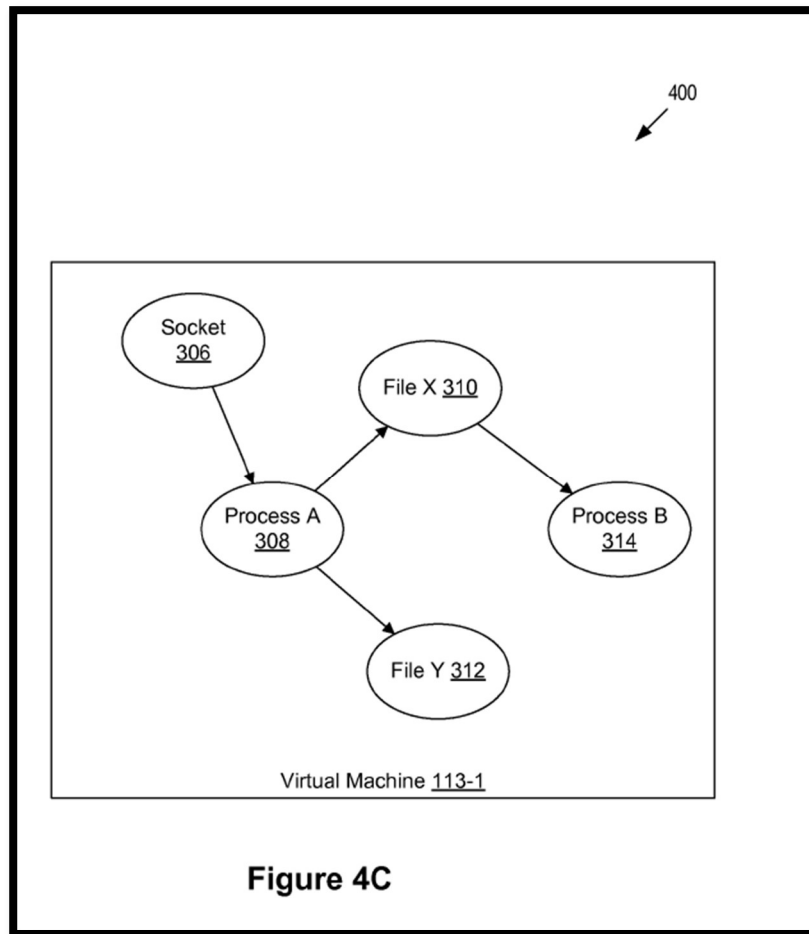
Next, as shown in Figure 4B, the '872 patent discloses techniques for tracking the unauthorized activity to other components within the computer system. For example, “process A 308 accesses both the file X 310 and the file Y 312” and therefore is “treated as affected by unauthorized activities.” *Id.* at 12:15-17.

Likewise, because “file X 310 initiates the process B 314, ... the process B is treated as affected by unauthorized activities” as well. *Id.* at 12:17-19.

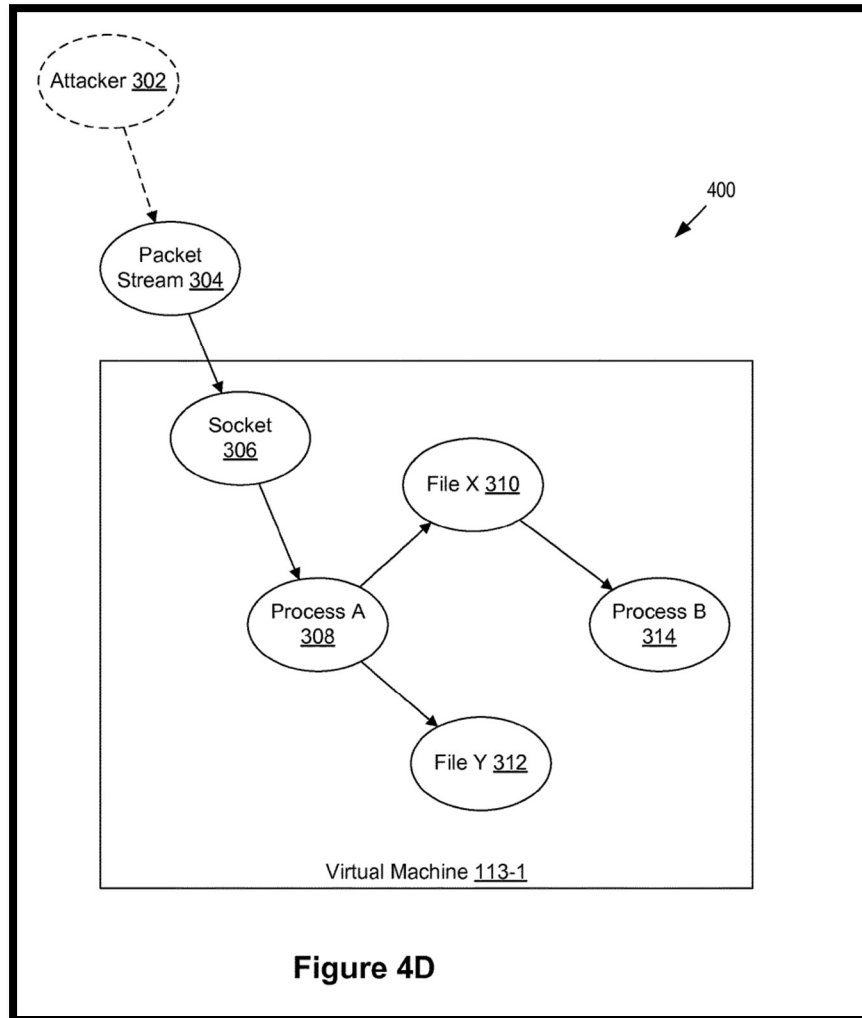


'872 Patent Fig. 4B

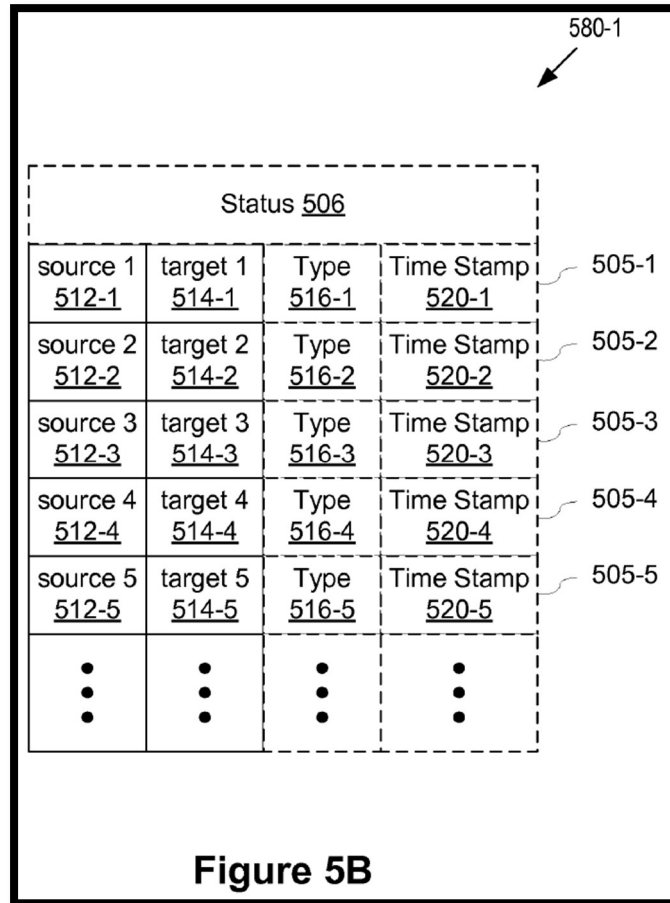
Next, as shown in Figure 4C, the '872 patent discloses techniques for tracking the unauthorized activities to a socket. *Id.* at 12:20-22 (“In FIG. 4C, data from the socket 306 is directed to the process A 308, and the socket 306 is treated as affected by unauthorized activities.”).

**'872 Patent Fig. 4C**

Lastly, as shown in Figure 4D, the '872 patent teaches that the “packet stream 304” received over socket 306 “is treated as affected by unauthorized activities,” which may allow “the attacker 302 who sends the packet stream 304 to the socket 306 [to be] identified (e.g., by the IP address).” *Id.* at 12:24-26. Only here does an external access come into view. Prior art systems that only monitor for unauthorized accesses—like *Capalik*—may not have detected the change in Files X and Y that led to identifying this unauthorized activity. Stavrou ¶28-29.

**'872 Patent Fig. 4D**

Critically, the techniques disclosed and claimed in the '872 patent also allow for improved security through the sharing of such information. For example, the computer system claimed in the '872 patent can form a fingerprint as shown in Figure 5B, which includes a status 506 as well as association type 516 for sources 512 and targets 514 of the monitored activity.

**'872 Patent Fig. 5B**

The monitored activity of the '872 patent, and the associated signatures, are different in kind from the types of activity that can be monitored via “low-level IP data packets” of these prior art IDS/IPS systems. '872 patent 2:2-5. To the contrary, the computer system of the '872 patent can share this expanded set of monitored/fingerprint data that *another computer system* can use to detect and prevent future attacks *against itself*. '872 patent 18:44-48. For example, the '872 patent discusses techniques by which the other computer systems can use the transmitted information to protect themselves by “modifying the activity targets”

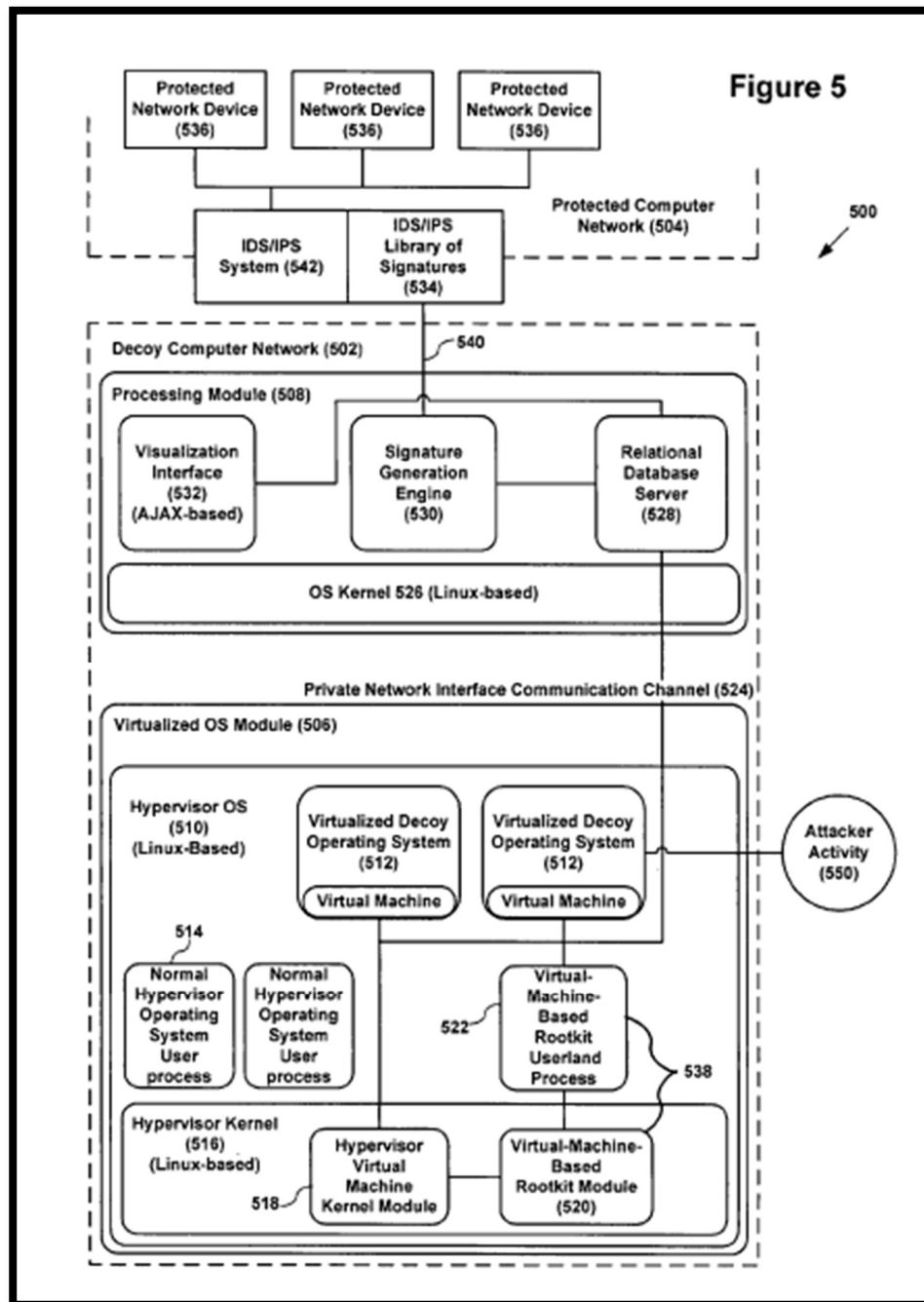
(*id.* at 14:5-16) or “blocking activity sources that match a pattern of unauthorized activities” (*id.* at 14:43-50). This groundbreaking approach allowed computers to detect more unauthorized activity and share it with other computing systems rather than relying exclusively on separate intrusion detection mechanisms of the prior art.

III. *Capalik* discloses a system for analyzing network intrusions based on detection of network activity.

As reflected in its title, GoSecure's earlier *Capalik* application discloses analyzing “unauthorized intrusion into a computer network.” Ex. 1004 (“*Capalik*”) Cover. As such, *Capalik* starts by detecting an unauthorized attempt to access a decoy network by a potential attacker. Only after detecting such an attempt does *Capalik*'s system begin monitoring activity, and only activity related to that attempt. Stavrou ¶35-36. After creating a signature related to the detected attempt, the signature is transmitted only to a signature library of a conventional intrusion detection and prevention system (IDS/IPS), thereby “allow[ing] for real-time updates to a network's library of attack signatures.” *Capalik* at Cover, [0007].

Specifically, *Capalik* discloses “a decoy computer network” that includes: 1) “a virtualized operating system module 506” and 2) “a processing module 508.” *Id.* [0038]. Ground 1 is limited to an embodiment where these two modules are “hosted on the same computing device,” which CrowdStrike calls decoy computing device 502. *Id.* [0039]; Pet. 15. As shown in Figure 5, below, *Capalik*'s decoy computing device includes a virtual machine with a virtualized operating system 512 and

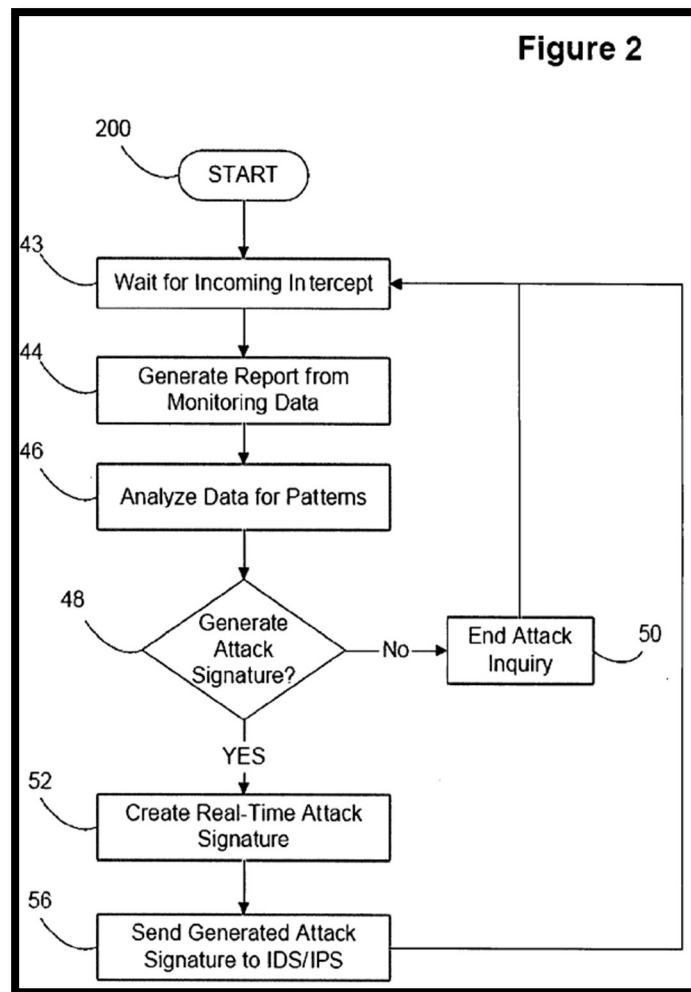
hypervisor kernel 516. *Capalik's* system also includes an IDS/IPS system 542 and an IDS/IPS library of signatures 534, both separate and distinct from the protected network devices 536 in protected network 504.



Capalik Fig. 5

Capalik discloses that “[w]hen an attacker connects to a network port and begins interacting with a virtualized decoy operating system 512, the introspection module 538 monitors and captures information from the connection, including port numbers, data streams, file uploads, keystrokes, ASCII or binary files, malicious payloads, memory manipulation attempts, and any other data transfers or malicious attempts.” *Id.* [0049].

Capalik's Figure 2 illustrates an example flowchart.



***Capalik* Fig. 2**

After waiting for an incoming intercept at step 43, the “processing module 508 generates a report of the attack” in step 44. *Capalik* [0055]. “This report is for the use and review by a system administrator.” *Id.* The system also determines whether to generate an attack signature based on whether “the system is responding in a manner that shows the attacker has gained access.” *Id.* If there is no data transfer, then the “attack inquiry may be ended at step 50.” *Id.*; *see also id.* [0059] (“[P]rocessing module 508 may not generate a signature when it is found that no data has been transferred through the socket.”). Otherwise, the processing module generates an attack signature at step 52 “by analyzing all the data passed through the socket,” which includes “analyzing the attack payload including the keystrokes or transferred bytes and any files uploaded to the client operating system of an ASCII or binary nature.” *Id.* [0013]. Thus, *Capalik* generates signatures for intrusions by analyzing *network traffic* passed through the decoy’s socket. The generated attack signature is “sent to the IDS/IPS signature library 534 at Step 56” and the method returns to step 43. *Id.* [0055]; *see also id.* [0054], [0059]. Specifically, “the attack signature is sent [to the IDS signature library] through a standard network connection... and is generally sent on a private portion of the protected network 504 that the attacker cannot observe.” *Capalik* [0059].

The IDS/IPS, in turn, can use the signature to “examine[] the low-level IP data packets” transmitted to a protected network device, separate from the IDS, providing a notification of a match with a previously-received signature. *Capalik* [0004].

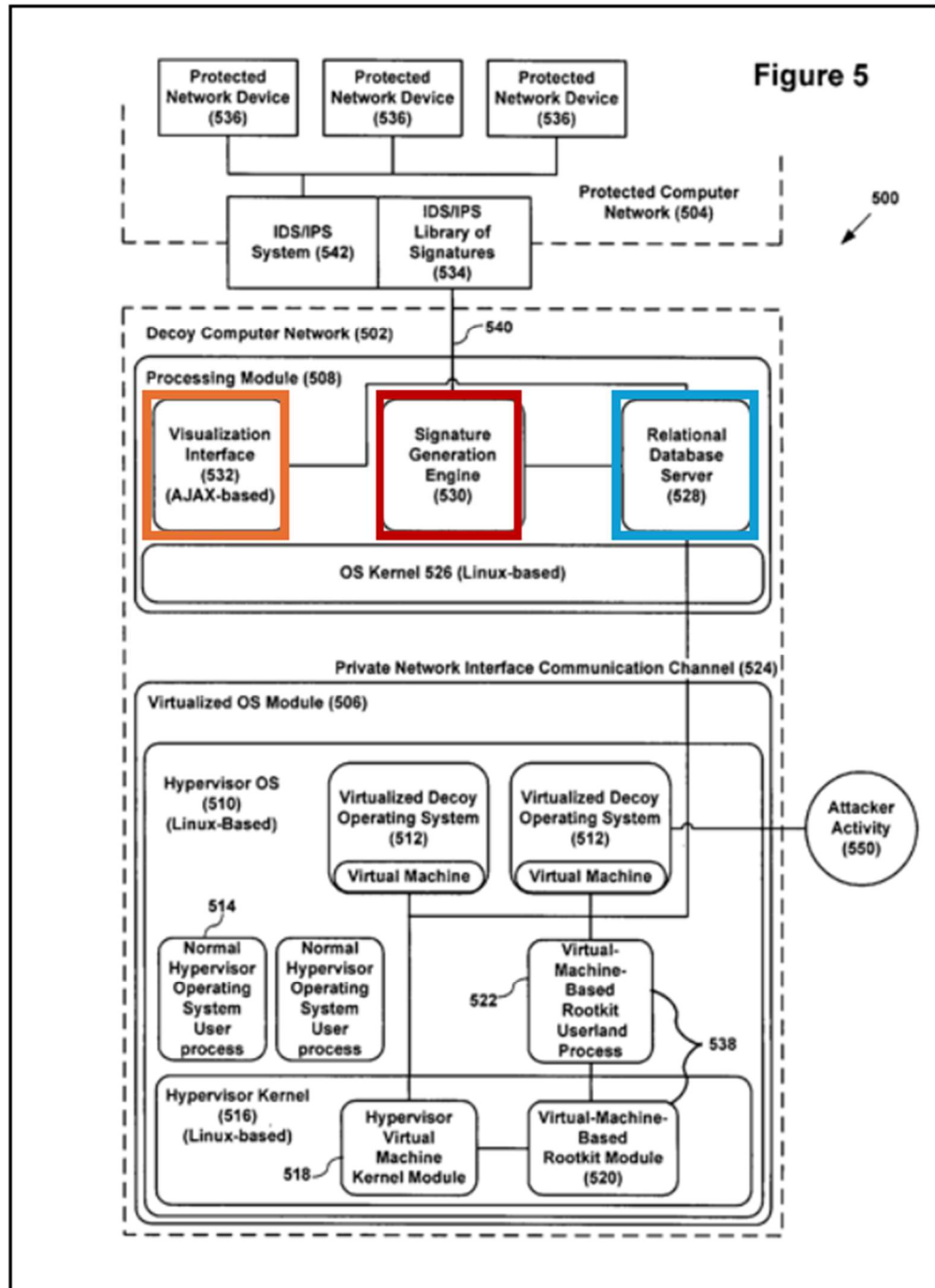
IV. Ground 1: CrowdStrike failed to show that *Capalik* renders any claim obvious.

A. Limitation 1[C]: *Capalik* does not teach or suggest storing in the memory a data structure that identifies the activity sources, the activity targets, and the associations for the plurality of activities.

The Petition has not shown that *Capalik* discloses or renders obvious “storing in the memory a data structure that identifies the activity sources, the activity targets, and the associations for the plurality of activities.”

In both grounds challenging claim 1, CrowdStrike asserts that the recited “data structure” 1[C] is disclosed by *Capalik*, which is GoSecure’s own published patent application. Pet. 27-30 (Ground 1), 40-46 (not relying on *Goel* for any “data structure” for limitation 1[C]). CrowdStrike’s analysis for 1[C] fails for two reasons.

First, the Petition is unsupported by *Capalik*’s disclosure, but instead based on its expert’s conclusory say-so. In both asserted grounds, CrowdStrike cites to *Capalik*’s relational database server 528 as “stor[ing] ‘attack information’ as forensic data for later analysis.” Pet. 28. Petitioner annotates Figure 5 to highlight the relational database server 528, as shown below. Pet. 27-28.



Capalik, Fig. 5 (as annotated at Pet. 28)

CrowdStrike points to zero explicit disclosure in *Capalik* of a data structure, let alone one that identifies for each of a plurality of activities, the sources, targets,

and associations between the sources and targets. Instead, CrowdStrike attempts to use its expert's conclusory testimony about the purported knowledge of a POSITA to bridge the gap between *Capalik* and 1[C]. Pet. 29 ("A POSITA would have understood or found obvious that *Capalik*'s 'forensic data' is stored in a data structure, as claimed, at least because the stored data is used" to generate a signature) (citing *Capalik* [0052] and Ex.1003 ¶¶134-136). That *ipse dixit* does not substitute for disclosure, and is insufficient as either an argument for inherency or a hindsight-free obviousness rationale.

Capalik does not mention a data structure and CrowdStrike is unable to find one in the four corners of *Capalik*. And CrowdStrike does not articulate any theory of inherency or obviousness, including any basis or rationale for modifying *Capalik*. Nor can CrowdStrike show obviousness by handwaving over this limitation due to its alleged "breadth" or its expert's testimony that data structures were generally known in the art. Pet. 29.

CrowdStrike argues that "the claims do not impose any specific structure of the stored data." Pet. 29. Not so. The limitation specifically requires storing a data structure "that identifies the activity sources, the activity targets, and the associations." CrowdStrike has failed to show disclosure or obviousness of this requirement.

Second, a POSITA would not have inferred a “data structure that identifies the activity sources, the activity targets, and the associations” from the mere fact that *Capalik*'s forensic data is used to generate a report or signature. Stavrou ¶59. Reports and signatures can be generated from unstructured data using intermediate processing; neither the reports nor the act of generating such reports retroactively supply the missing data structure. Stavrou ¶59.

CrowdStrike's own analysis under 1[D] underscores the gap: it does not show that *Capalik*'s signatures themselves identify an “association” between a source and a target for each of a plurality of activities. Indeed, CrowdStrike argues in 1[D] that the '872 patent only requires creating a signature from “one or more” of a source, target, or association, and does not attempt to identify in *Capalik* any such association in the signature. *See* Pet. 31-33. Of the three categories of information in 1[D], CrowdStrike identifies an alleged target (“specific port on the virtual machine through which the attack occurred”) as a component of *Capalik*'s signature. Pet. 32-33; *see also id.* 31 n.8 (analogizing *Capalik* and the '872 patent as both teaching “a fingerprint/signature of an attack that identifies the port (or socket) targeted by the attack”). CrowdStrike made no attempt—at all—to show that the signature identifies an “activity source” and an “association,” much less a data structure that identifies source, target, and associations for each of a plurality of activities as required by 1[C]. Thus, CrowdStrike's logic—that the signatures

imply a stored data structure that identifies, among other things, an association between a source and target—falls apart based on its own analysis.

Capalik is completely silent about data structures. *Capalik* merely stores “data” in a database with no detail about any sort of data structure identifying sources, targets, and associations for each of the identified activities. *Capalik* [0052]. *Capalik*'s sparse disclosure stands in stark contrast to the '872 patent, which not only includes “Using A Data Structure Model” in its Title, but explicitly illustrates and discusses multiple detailed data structures. See '872 patent 12:45-48, 13:65-66, 14:21-22, Figs. 5A-5C. These illustrated data structures, like the claimed data structure, identify sources, targets, and associations for each of the identified plurality of activities. *Id.*; Stavrou ¶¶60, 30.

Accordingly, *Capalik* does not teach or suggest limitation 1[C], and does not render claim 1 obvious.

B. Limitation 1[D]: *Capalik* does not teach or suggest transmitting activity information to other computer systems to prevent future attacks on those systems.

The computer-implemented method of claim 1 is performed “on a first computer system,” which is “attached to a computer network” and “comprises one or more processors and memory.” The next three steps require “monitoring activity on the first computer system,” “identifying ... activities being performed at the first

computer system,” and “storing” a data structure in the memory of the first computer system.

The last step of this method, which the Petition labels limitation 1[D], requires information be (1) transmitted from the first computer system (2) to one or more computer systems other than the first computer system, (3) for preventing future attacks to those other computer systems. Further, the transmission includes “information identifying one or more of the activity sources, the activity targets, and the associations.” For example, as the specification explains, such information “may be used for providing detailed forensic analysis of the character and nature of the attack and its impact on the target operating system, which can be used as the basis for generating triggers.” ’872 patent 10:49-55. The “one or more other computer systems” may, in turn, use these triggers “to deny malicious system or library calls[.]” *Id.* 10:53-55.

1. The Petition failed to show that Capalik discloses or renders obvious the transmission of 1[D].

In both grounds challenging claim 1, CrowdStrike asserts that 1[D] is disclosed by *Capalik*, which is GoSecure’s own published patent application. Pet. 30-34 (Ground 1), 40 (Ground 2 “relies on *Capalik*’s teachings as set forth in Ground 1” for 1[D]).

CrowdStrike cited to the ’872 patent’s embodiment of sending a fingerprint to IDS/IPS 142 and asserted that *Capalik* “similarly teaches” transmission of a

signature to an IDS. Pet. 33 (citing '872 patent at 18:47-51). Specifically, CrowdStrike pointed to *Capalik's* teaching of an “attack signature” that can be sent “to the intrusion detection system (IDS) or intrusion prevention system (IPS) for the protected network 504.” Pet. 33. Thus, at first blush, the Petition appears to identify *Capalik's* IDS/IPS as the “other” computer system of 1[D], relying on the specification of the '872 patent.

But CrowdStrike does not, and cannot, contend that the IDS/IPS in *Capalik* is the computer system on which future attacks need to be prevented. That is because, even if *Capalik's* IDS/IPS were a computer system in the abstract, there is no dispute that transmission of a signature to an IDS/IPS does not prevent future attacks *on the IDS/IPS*. Instead, the fingerprint is sent to the IDS/IPS to protect other network devices. *See, e.g.*, Pet. 33 (“The attack signature is then applied to the IDS/IPS library of signatures, updating the library, to prevent an attacker from accessing protected network 504 in a future similar attack.”); Ex.2033 at 47:12-48:4 (“...So given my understanding that the IDS/IPS system 142 is connected to the protected network devices 136, which is supported by Figure 5, I do say that it's – *the fingerprint is sent to the IDS/IPS to protect the other network devices....*”); Stavrou ¶63. This theory fails to meet the claims because of the fundamental disconnect between the recipient of the information and the computer system to be protected.

The Petition, however, attempts to bridge that disconnect by asserting that the information is further transmitted to the protected network devices themselves. CrowdStrike asserts that “a POSITA would have understood that signatures updated in the signature library are *also transmitted to the multiple protected network devices* 536 coupled to the IDS/IPS system, ensuring those network devices receive protection consistent with the most up to date signatures collected in the library.” Pet. 33-34 (citing Ex.1003 ¶¶143-144).

But there is no such disclosure in *Capalik*, nor is there any evidence that a POSITA would have understood the signatures to be transmitted to protected devices in *Capalik*. Tellingly, CrowdStrike and Dr. Jakobsson did not cite any portion of *Capalik* that discloses transmitting signatures to the protected network devices, because no such disclosure exists. Both CrowdStrike and Dr. Jakobsson handwaved over the very limitation that the Examiner relied on when allowing the claims, impermissibly using the '872 patent as a roadmap for their hindsight-based obviousness argument. *See* Ex.1002 55 (amendment to 1[D]), 15 (citing amendment as reason for allowance).

Neither CrowdStrike nor Dr. Jakobsson articulated any rationale for obviousness of this claim requirement. Dr. Jakobsson simply declared that a skilled artisan “would have understood” that signatures are transmitted to the protected network devices, but he provided no explanation of why this would have been

obvious, no citation to any technical literature supporting this conclusion, and no analysis of what problem such transmission would solve or what benefit it would provide. The Petition likewise offered no rationale. It quoted Dr. Jakobsson's conclusory statement and moved on, as if the mere invocation of what a skilled artisan "would have understood" suffices to establish obviousness. It does not. *KSR Int'l. Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007) ("Rejections on obviousness grounds cannot be sustained by mere conclusory statements.") (cleaned up).

To meet its burden, CrowdStrike was required to either (1) show this disclosure is inherent in *Capalik* or (2) articulate a reason why a skilled artisan would have modified *Capalik* such that the signatures were transmitted to the protected devices themselves, supported by rational underpinning or evidence in the record. CrowdStrike provided neither. Without any explanation of why a skilled artisan would have made this modification, Dr. Jakobsson's testimony amounts to impermissible hindsight reconstruction—working backward from the '872 patent's claims to assert that *Capalik* must have done the same thing. *KSR*, 550 U.S. at 421 ("A factfinder... must be cautious of arguments reliant on *ex post* reasoning.") (citing *Graham v. John Deere Co.*, 383 U.S. 1, 36 (1966)). The Board should reject this approach. Because this is the only theory articulated in the Petition, across all Grounds, this defect is fatal to CrowdStrike's arguments, and the Board should find all claims patentable.

The Board should find that the Petition failed to meet its burden of showing that *Capalik* discloses the transmitting limitation.

2. CrowdStrike's new theory cannot salvage the Petition's deficiencies.

Faced with a deficient Petition and a mismatch between the '872 claims and *Capalik*, CrowdStrike modified its theory and introduced a belated and erroneous claim construction in support.

In its Reply to the POPR, CrowdStrike changed its theory of unpatentability by arguing that the “other” “computer systems” recited in claim 1 are taught by “the protected network, including protected endpoint devices and the IDS/IPS.” Prelim. Reply 2. Notably, CrowdStrike made no attempt to show that such a theory was articulated in the *Petition*. In addition, CrowdStrike urged the Board to construe the claim “broadly [enough] to capture” this mapping, but declined to offer an actual construction. Prelim. Reply 2. This was improper.

Neither CrowdStrike nor Dr. Jakobsson ever articulated that theory in the Petition. The Petition did not construe “computer system” or identify *Capalik*'s protected network 504 itself as the “computer system[] other than the first computer system” that receives the information for preventing future attacks to it. On cross-examination, Dr. Jakobsson admitted that he did not construe “computer system” in his declaration and could not point to any disclosure of a theory that *Capalik*'s

network 504 is the “other” computer system of the claim. Ex.2033 (Jakobsson Dep.) at 44:10-45:13.

As the Supreme Court has held, the Petition is “the centerpiece of the proceeding both before and after institution.” *SAS Inst. Inc. v. Iancu*, 584 U.S. 357, 369 (2018). An inter partes review must proceed “in accordance with or in conformance to the petition.” *Id.* at 365 (cleaned up). In addition, the Board’s rules require “the petition” to identify “[h]ow the challenged claim is to be construed.” 37 C.F.R. § 42.104(b)(3). The Board should not consider CrowdStrike’s new theory, which is not in the Petition.

Nevertheless, even if the Board were to allow CrowdStrike’s new theory, it is premised on an erroneous claim construction. The intrinsic record makes it clear that the “computer system” recited in the claims is an individual computer system, not any random collection of computing devices. Specifically, the record shows that the claimed “computer system” does not encompass a “computer network.”

a. The claimed “computer system” is a single computing device and not a network.

The '872 patent uses the term “computer system” to refer to a computing device and not a network.

The context of the claims supports GoSecure.

CrowdStrike's proposed construction—where a computer system can include a network—would render several terms in the claim superfluous and the claim as a whole meaningless.

The preamble of claim 1 recites that the “first computer system [is] attached to a computer network.”¹ By distinguishing between a “computer system” (which has processors and memory) and the “computer network” to which it is attached, the

¹ The preamble is limiting on the “computer system,” at least because it is not an intended use of the computer system and imparts structure that is necessary for the bodies of Claim 1 and its dependent claims. *See, e.g., Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 952 (Fed. Cir. 2006) (“[W]hen the limitations in the body of the claim rely upon and derive antecedent basis from the preamble, then the preamble may act as a necessary component of the claimed invention.”) (cleaned up). For example, the preamble states that the “first computer system comprises one or more processors and memory,” and the body of Claim 1 requires that a data structure is stored “in *the memory*.” And Claim 14 requires that “instructions executed by the *one or more processors* have respective privilege levels.” Likewise, the preamble of Claim 1 recites that the “first computer system” is “attached to a computer network,” and Claim 4 requires “receiving data through *the network*.”

claim makes clear that a computer system is not the same as the network. If a computer system were a network, the phrase “attached to a computer network” would be superfluous and nonsensical. *See Wasica Fin. GmbH v. Cont'l Auto. Sys., Inc.*, 853 F.3d 1272, 1288 n.10 (Fed. Cir. 2017) (“It is highly disfavored to construe terms in a way that renders them void, meaningless, or superfluous.”). It makes no sense for a computer system to be attached to a network if the computer system also encompasses the network.

The Federal Circuit's decision in *Becton, Dickinson & Co. v. Tyco Healthcare Grp., LP*, 616 F.3d 1249, 1255 (Fed. Cir. 2010) is instructive. There, the Court held that, in a claim that required a “spring means connected to said hinged arm,” the question was whether the spring means could be the same as “hinged arm.” The Court held it could not, because “[i]f the hinged arm and the spring means are one and the same, then the hinged arm must be ‘connected to’ itself, rendering the claims ‘facially nonsensical.’” *Id.* at 1255. The same result follows here.

Additionally, if a “computer system” encompassed multiple computers, GoSecure would not have had to specify “one or more computer systems” in 1[D] because “one or more” would be superfluous. For example, Dr. Jakobsson opined that “computer system” means “one or more computational devices.” Ex.2033 53:5-7. Inserting Dr. Jakobsson's construction into the claim—“transmitting to *one or more [one or more computing devices]*”—shows that his construction renders

superfluous the phrase “one or more.” *Bicon Inc. v. Straumann Co.*, 441 F.3d 945, 950-52 (Fed. Cir. 2006) (improper to contradict “the principle that claim language should not be treated as meaningless”).

The specification refutes CrowdStrike's construction.

The '872 patent's specification confirms the claimed steps are performed on a single computing device. Stavrou ¶¶43, 46-47, 55.

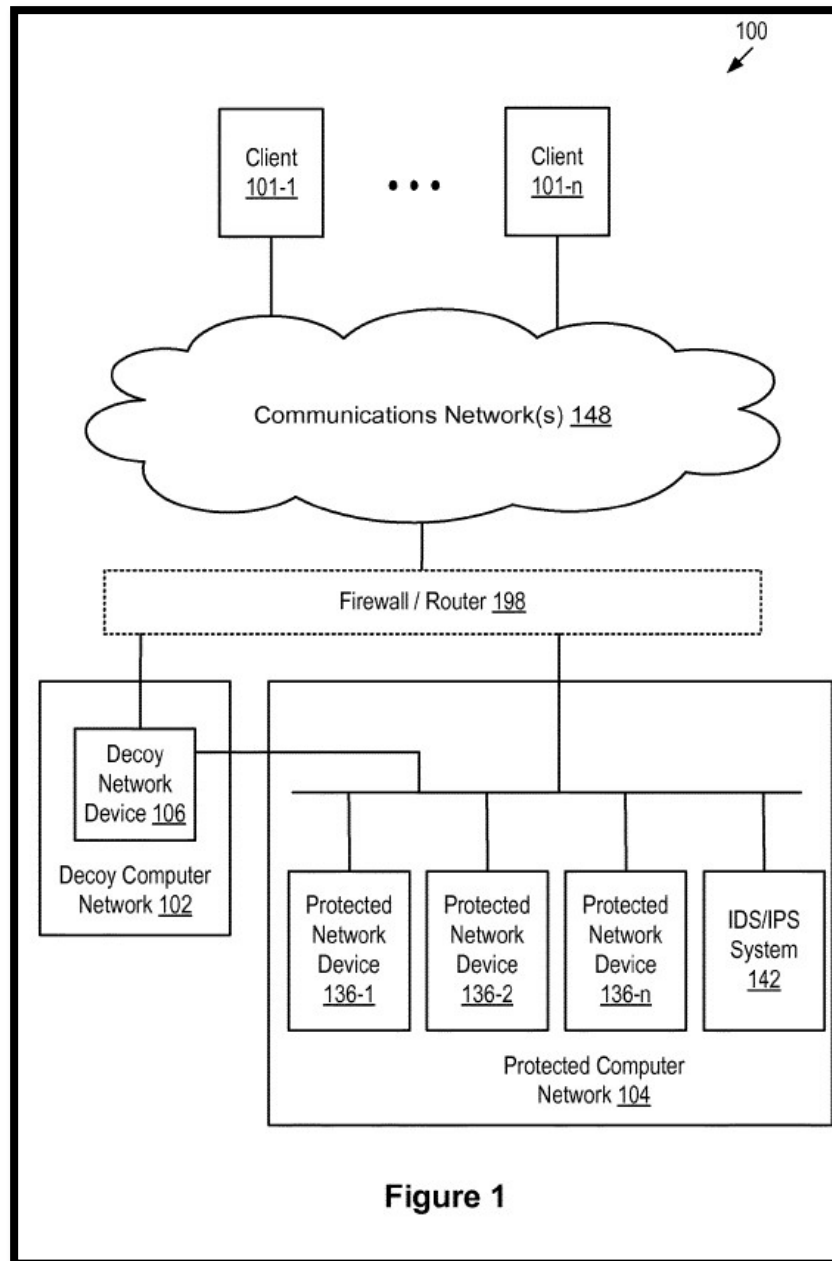
First, the '872 patent uses “device” and “computer system” interchangeably. It describes that in some embodiments, the fingerprint generation engine and the virtualized decoy OS “are part of or stored within the decoy network *device*” and in “other embodiments” they “are implemented using separate *computer systems*.” '872 patent at 8:29-34; *see also id.* at 18:52-53 (“one or more other computer systems (e.g., protected network devices 136) on the network”). Moreover, the flowcharts of Figures 6A-6C represent “a method 600 of identifying unauthorized activities on a *computer system* (e.g., the virtual machine 113-1) attached to a computer network.” '872 patent 14:61-64. This “***computer-implemented method is performed at a decoy network device*** (e.g., 106 in FIG. 1B) having one or more processors and memory storing a virtual machine.” '872 patent 14:64-67. In turn, “virtual machines” are “one or more software applications emulating one or more ***physical machines (e.g., computer systems)***.” '872 patent 7:59-61. Thus, the specification uses “computer system” interchangeably with machine or device. Stavrou ¶46. This

favors construing a “computer system” as a machine/device. *SAS Inst., Inc. v. ComplementSoft, LLC*, 825 F.3d 1341, 1348 (Fed. Cir. 2016) (construing “graphical representation of data flow” to be limited to the description of “data flow diagram” in the specification because those terms were used interchangeably), *rev'd on other grounds*, 584 U.S. 357 (2018).

CrowdStrike asserts that the '872 patent uses “computer system” to “refer to a virtual machine module within a decoy network device” because it discloses “a computer system (e.g., the virtual machine 113-1).” Paper 9 at 2 (quoting '872 patent 11:15-19). But here, the specification is describing the virtual machine as an example component of the computer system where activity is occurring. Stavrou ¶46 n.2; *see also* '872 patent 11:23-25 (another example of “e.g.” used in a similar manner). Rather, the '872 patent defines a VM as emulating a computer system. '872 patent 7:59-61. Dr. Jakobsson provided a similar definition. *See, e.g.*, Ex. 1003 ¶55 (“*Colbert* teaches that “[a]s is well known in the field of computer science, a virtual machine (VM) is an abstraction—a ‘virtualization’—of an actual physical computer system.”) (quoting Ex. 1020 [0059]). But even if “computer system” can encompass a virtual machine, that does not support reading it to encompass a computer network.

Second, in contrast to the consistent usage of “computer system” to mean a single device, the '872 patent consistently uses *other* terms—network or distributed

computer system—to describe groups of multiple devices. For example, the '872 patent discloses that “computers are connected to computer networks” and that computers “access[] and shar[e] data over the networks.” '872 patent 1:52-55. The '872 patent discloses that a fingerprint can be “transmitted to one or more other computer systems on the network to prevent future attacks.” '872 patent 2:44-45. Figure 1, furthermore, illustrates three networks: communications network 148, decoy computer network 102, and protected computer network 104. The '872 patent never uses the term “computer system” to refer to any of these (or any other) networks. Stavrou ¶47.

**'872 Patent Fig. 1**

The '872 patent uses the term “distributed computer system” to refer to the entire collection of networks, computer systems, and appliances in Figure 1. Specifically, Figure 1 depicts “distributed computer system 100” that includes decoy computer network 102, communications network 148, and protected computer

network 104, as well as firewall/router 198 and clients 101. '872 patent Fig. 1; 4:5-9. The distributed computer system is therefore the collection of every computer system, network, and appliance discussed in the '872 patent—including the Internet. Stavrou ¶53; '872 patent 6:1-2 (listing the Internet as an example communications network 148). No other “distributed computer system” exists in the '872 patent. In fact, no *component* is disclosed to be outside of the distributed computer system. Stavrou ¶53. As a result, the '872 patent does not disclose any transmissions to any computer system outside of this distributed computer system, nor could there be any. Stavrou ¶53.

Even so, the Board in its Institution Decision stated that “‘distributed computer system’ refers to a particular type of computer system” and that “[l]ogically..., the language ‘computer system’ omits the modifier ‘distributed’ and thus, refers to all types of computer systems, including distributed computer systems.” Paper 15 (“Inst. Dec.”) 15-16. This is incorrect for two reasons.

First, a “distributed computing system” is not necessarily a type of “computer system” any more than a “stuffed animal” is necessarily a type of animal, or “virtual reality” is necessarily a type of reality.

Second, in the context of the '872 patent, the “distributed computer system 100” is indisputably *not* an embodiment of the claimed “one or more computer systems other than the first computer system” to be protected. There can be no

transmission outside of the distributed computer system 100 because there are *no* computer systems other than the distributed computer system. As shown in the annotated Figure 1, below, all transmissions (orange) would be within the alleged “computer system” 100 (purple). Stavrou ¶53.

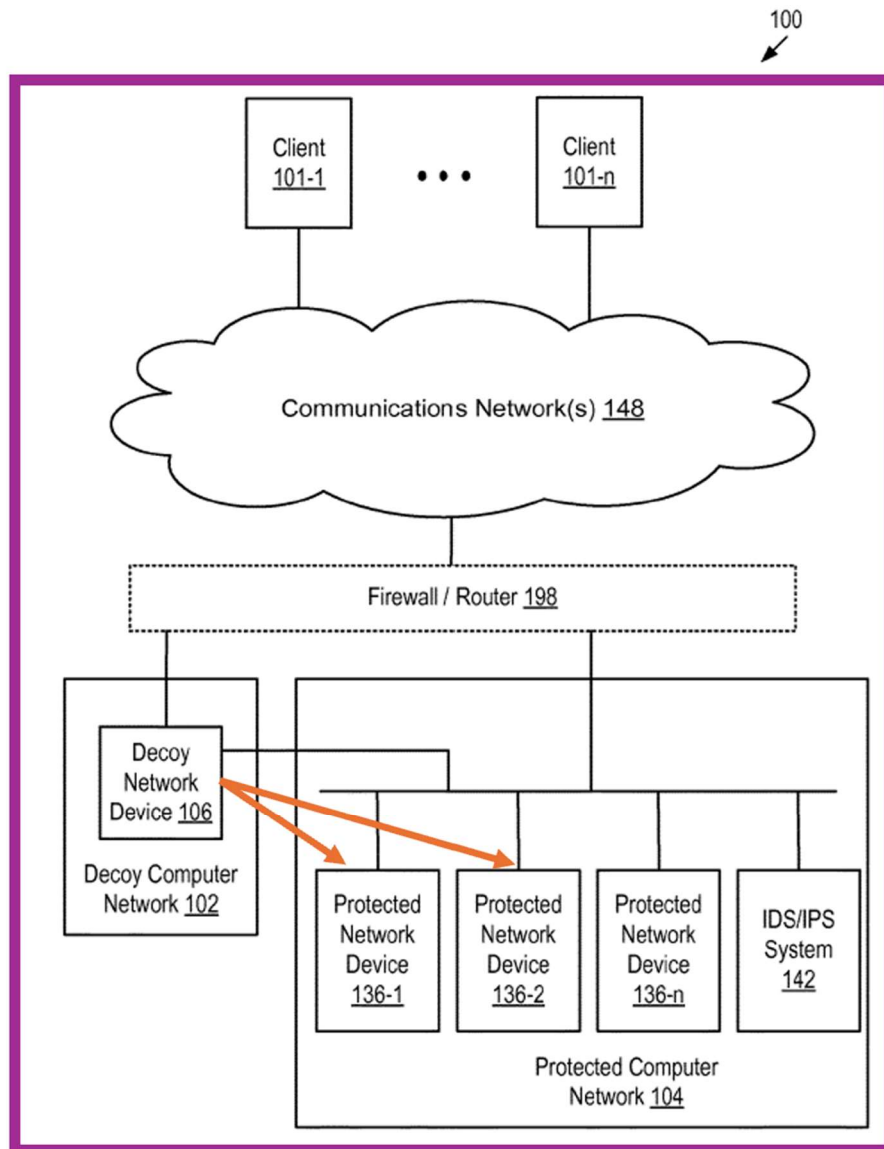


Figure 1

'872 Patent Fig. 1 (annotated)

Reading “computer system” to cover the “distributed computer system” thus contradicts the claims and the specification as a whole—rendering the claims inoperable—and therefore cannot be correct. *On Demand Mach. Corp. v. Ingram Indus., Inc.*, 442 F.3d 1331, 1344 (Fed. Cir. 2006) (“Care must be taken lest word-by-word definition, removed from the context of the invention, leads to an overall result that departs significantly from the patented invention.”). To be sure, a computer system can encompass multiple components that work together as one unit, but that does not mean the claimed computer system covers any collection or network of computers.

The Board also asserted that the protected network devices 136 are “only an example of ‘other computer systems,’ demonstrating that ‘computer systems’ refers to things other than protected network devices.” Inst. Dec. 15. The Board cites a passage describing “one or more other computer systems (e.g., protected network devices 136).” Inst. Dec. 15 (quoting ’872 patent 18:52-53). But nothing in this passage states or implies that “computer system” encompasses anything more than a single computing device. The fact that protected devices 136 (i.e., those illustrated in Figure 1) are computer systems does nothing to expand the scope of “computer system” beyond a single computing device.

b. The claimed transmission to another computer system does not cover mere transmission to an IDS/IPS.

The intrinsic and extrinsic evidence contradicts CrowdStrike's underlying argument—the intrinsic record repeatedly distinguishes between (1) the claimed transmission to a computer system for the computer system's protection, and (2) transmission to an IDS.

The '872 patent expressly distinguishes (1) transmitting signatures or fingerprints to computer systems to be protected (the protected network devices) as claimed from (2) transmitting to an IDS/IPS. In one embodiment, the “decoy network device transmits [a] fingerprint to *one or more protected network devices 136* to prevent future attacks.” *Id.* at 18:44-47.

In other embodiments, however, the “decoy network device transmits the fingerprint *to the IDS/IPS system 142* to prevent future attacks that comprise the same or similar activities on one or more protected network devices 136.” *Id.* at 18:47-51. This explicitly distinguishes between transmitting (1) to a protected network device to prevent future attacks to that device and (2) to an IDS/IPS to prevent future attacks to other devices. In view of this disclosure, a POSITA would understand the claim language (which requires transmission to “one or more computer systems” to prevent future attacks to those systems) to refer to the former embodiment. Stavrou ¶65.

The Federal Circuit has held that where the specification describes two embodiments and the claim language pertains to one and not the other, the claim covers the former, not the latter. The Federal Circuit's decision in *Baran v. Med. Device Techs.* is instructive. There, the Federal Circuit construed "detachable" to mean separation without loss or damage according to the "reusable embodiment," but not the "single-use embodiment" described elsewhere in the specification. 616 F.3d 1309, 1314-16 (Fed. Cir. 2010). The court emphasized that the patentee deliberately used different terminology to contrast embodiments and that it is not necessary that each claim read on every embodiment. *Id.*

So too here. The '872 patent uses different terms to distinguish different transmissions: transmission to "protected network devices 136" (the specification's example of "other computer systems") on the one hand, and in other embodiments transmission to an "IDS/IPS system 142" as a distinct alternative. '872 patent 18:44-58.

The specification also consistently refers to an IDS as a distinct entity from protected network devices. *See, e.g.*, '872 patent 4:36-39 (in addition to protected network devices, protected network 104 "*also* includes" an IDS/IPS system 142 in "some embodiments"). In contrast, the specification never identifies the IDS as part of a "computer system" with devices to be protected. That distinction further underscores why it is improper to group the IDS and the protected network devices

in the same “computer system” bucket. *See Perfect Surgical Techniques, Inc. v. Olympus Am., Inc.*, 841 F.3d 1004, 1013 (Fed. Cir. 2016) (“[T]he specification’s separation of the terms perforated and passages with the disjunctive phrase ‘or otherwise’ makes clear that the patentee intended that the term ‘perforated’ is not the same as ‘passages.’”).

Unlike its interchangeable references to computer systems as computing devices, the ’872 patent never refers to the IDS/IPS as a “computer system.” Indeed, the specification states that the IDS or the IPS protects other computer systems, not that it is one. ’872 patent at 2:1-2 (“[T]he IDS or the IPS can detect attempts to access computer systems without authorization.”); *see also* Ex.2033 at 47:12-48:4 (“...So given my understanding that the IDS/IPS system 142 is connected to the protected network devices 136, which is supported by Figure 5 [of the ’872 patent], I do say that it’s – *the fingerprint is sent to the IDS/IPS to protect the other network devices....*”).

Construing “computer system” to encompass any collection of computing devices (and even the entire Internet) leads to the claim becoming unintelligible. Reinforcing that point Dr. Jakobsson analogized his (and the Board’s) construction of “computer system” to “**dough**” that can be shaped and cut in “many, many ways” to suit his needs. Ex.2033 57:7-24. In other words, a computer system can be *anything* and *everything*—one computer, a network, an arbitrary collection of

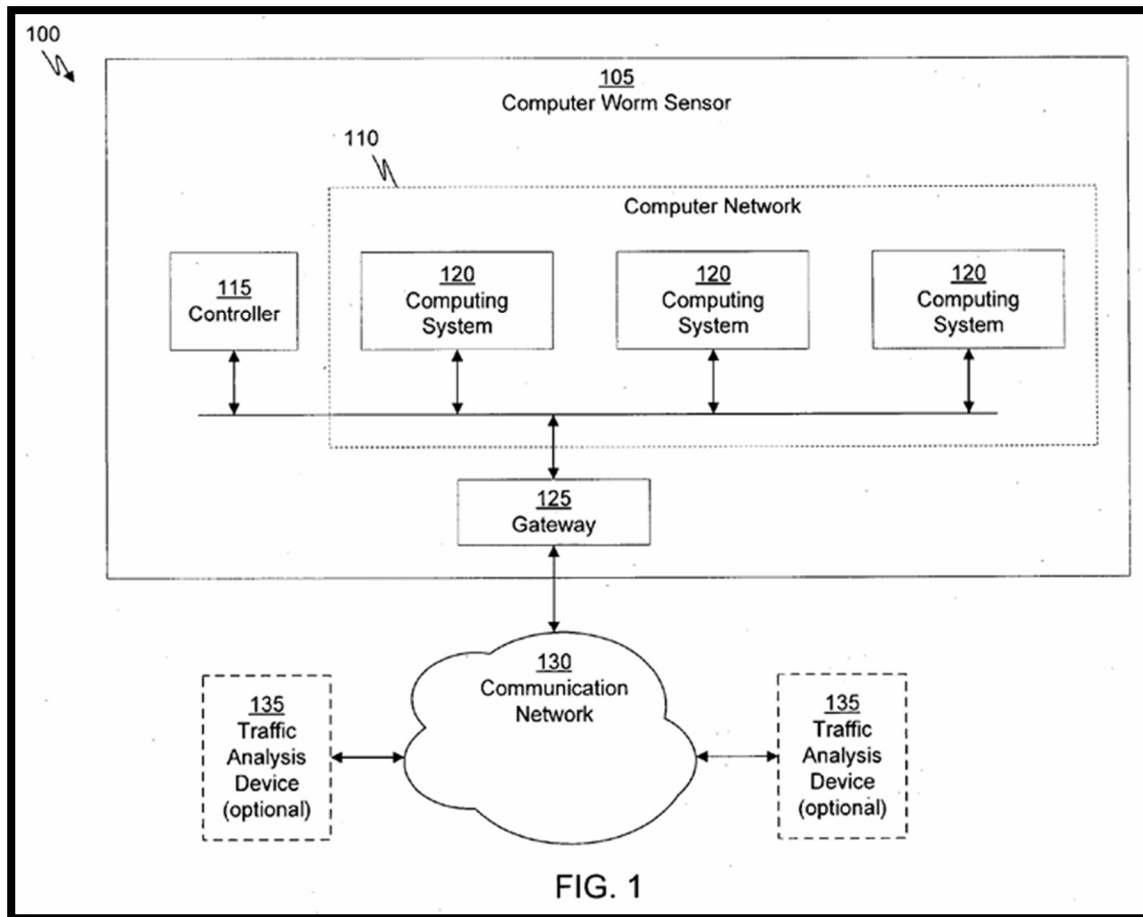
unconnected computers, or “*every computer on earth.*” Ex.2033 57:9-13. That boundless interpretation of “computer system” cannot be correct.

Additionally, allowing the “computer system” to be anything and everything would render the claim indefinite, contravening Federal Circuit precedent. *See, e.g., Geneva Pharms., Inc. v. GlaxoSmithKline PLC*, 349 F.3d 1373, 1384 (Fed. Cir. 2003). That is, any transmission between two devices would simultaneously both (1) satisfy 1[D] as a transmission to another computer system and (2) not satisfy 1[D] because the two devices are part of the same computer system. Dr. Jakobsson admitted that he “cut the dough” of *Capalik*’s Figure 5 simply to match the claim—but that other ways of cutting the dough would not meet the claim. Ex.2033 57:22-58:23 (Dr. Jakobsson cutting the “dough” of a computer system purposely “to map the claim to the situation”); 59:15-60:10 (“[I]f the claim says this thing, what would that have been in the context of *Capalik*.... [I]f you’re thinking of... claim one, and you’re pointing to Figure 5 and you would say does this correspond to claim one, I would say yes.... [I]f you’re saying but call the other parts computer system, well, those computer system may not correspond to claim one, but the ones I have identified do.”).

The prosecution history further confirms that the claimed transmission of 1[D] does not encompass transmission to an IDS. GoSecure knew how to claim transmitting information to an IDS. *See, e.g., Laryngeal Mask Co. v. Ambu*, 618 F.

3d 1367, 1373 (Fed. Cir. 2010) (relying on prior art patents with same inventor to construe disputed claim term). Indeed, GoSecure explicitly claimed transmitting information to an IDS in the *Capalik* reference and other related patents that are incorporated by reference into the '872 patent, and therefore part of the intrinsic record. See *Capalik* at claim 2 (“providing the attack signature to an ***intrusion prevention system*** configured to control access to a protected network.”). Additionally, U.S. Patent 8,656,493 (Ex.2036) (“*Capalik 493*”), which is a continuation of the same parent application as *Capalik*, expressly recites both a “***computer system***” and an “***intrusion prevention system***” in the claims. See Ex. 2036 at Claim 1. Notably, the back and forth between the examiner and GoSecure in prosecution of *Capalik* and *Capalik 493* were cited to the examiner during prosecution of the '872 patent. Ex. 1002 at 163 (citing actions from *Capalik 493* as C5 and actions from *Capalik* prosecution as C2, C3); see also '872 patent at 2-3 (listing various *Capalik*-related items submitted to the examiner during prosecution). Accordingly, a POSITA and the examiner would have understood that GoSecure made a deliberate choice in the '872 patent of claiming transmission to the same “computer system” that is to be protected, rather than to a separate IDS/IPS system (as described in the specification of the '872 patent). That choice, reflected in the claims, must be given effect. *Scantibodies Lab’y, Inc. v. Immunotopics, Inc.*, 374 F. App’x 968, 971 (Fed. Cir. 2010).

Not only does the context of the claim and the specification demonstrate that the “computer system” to be protected refers to a single computing device, but other intrinsic evidence shows that the ’872 patent’s usage of “computer system” was the ordinary and customary meaning of the term. During prosecution, the Examiner rejected previous claims as obvious over the combination of *Jiang* (Ex.2037) and *Aziz* (Ex.2038). Ex.1002 76-77. But *Aziz* uses the term “computing system” in a similar manner as the ’872 patent uses “computer system.” Stavrou ¶44. Specifically, “*computing systems 120 are computing devices typically found in a computer network.*” *Aziz* [0048]. Figure 1 of *Aziz*, reproduced below, further illustrates multiple computing systems 120 forming a computer network.



Atiz Fig. 1

Similarly, *Armstrong* (Ex.2039) was cited by the Examiner during prosecution of the '872 patent, making it part of the intrinsic record of the patent. Ex.2039 (*Armstrong*); '872 patent at 2 (citing *Armstrong*); Ex.1002 at 140 (citing *Armstrong*); *Phillips v. AWH Corp.*, 415 F.3d 1303, 1317 (Fed. Cir. 2005) (en banc) ("The prosecution history, which we have designated as part of the 'intrinsic evidence,' consists of the complete record of the proceedings before the PTO and includes the prior art cited during the examination of the patent."). *Armstrong* uses the term "computer system" to mean a physical machine—i.e., a computing device—that is

specifically distinct from a network. “[E]ach computer system in the network or array is a physically separate entity with its own network attachment, its own central processing unit(s), its own instance of an operating system, etc.” Armstrong [0005]. This language mirrors the language of the ’872 patent’s claims, which recite the computer system comprises a processor, memory, and that it is *attached to a network*. Stavrou ¶45. Armstrong—cited by the Examiner—informed his understanding of “computer system” in allowing the claims over *Capalik* and other references. *See V-Formation, Inc. v. Benetton Grp. SpA*, 401 F.3d 1307, 1311 (Fed. Cir. 2005) (relying on cited prior art patent “to set the context for [] claim construction”). Again, these cited references support construing “one or more computer systems” to be protected as separate computing devices.

In summary, the claims, specification, and other intrinsic evidence demonstrate that CrowdStrike’s proposed construction cannot be correct.

c. Extrinsic evidence is consistent with the ’872 patent’s usage of “computer system.”

Extrinsic evidence supports that the ordinary and customary meaning of “computer system” was a single computing device. In 2002, the *Microsoft Computer Dictionary* defined “computer system” as “[t]he configuration that includes all functional components of a computer and its associated hardware.” Ex.2049 at 105; Stavrou ¶50.

CrowdStrike's Dr. Jakobsson used the term "computer system" in the same manner in a 2002 patent. Stavrou ¶48 (citing Ex.2040, 3:10-33, 4:27). And on its website, CrowdStrike repeatedly distinguishes "computer systems" from networks. *See* Ex.2041 ("computer system *or* network"); Ex.2042 (same); Ex.2043 ("network *or* computer system"); Ex.2044 ("networks *or* computer systems"); Ex.2045 (same). These and other webpages of CrowdStrike's use "computer system" to refer to a computing device. *See* Ex.2046 ("connects an infected drive to a computer system"); Ex.2047 (equating "computer system" with a PC); *see also* Ex. 2048 (equating monitoring a computer system to endpoint monitoring).

To be sure, some authors have used "computer system" to encompass more than one computing device—but as shown in the previous section, including CrowdStrike's website, Dr. Jakobsson's patent, and the Microsoft Computer Dictionary, that is not the ordinary and customary use of the term in the context of the '872 patent and was not how the parties viewed the term in light of the intrinsic evidence. Stavrou ¶55. Rather, the ordinary and customary use is as a single computing device.

In sum, the claimed "other computer systems" are other such computing devices (e.g., protected network devices) subject to future attacks based on the information transmitted to them. This construction preserves the claim's internal distinctions, gives meaning to every term, avoids indefiniteness concerns, and aligns

with how the specification, the file history, and the technical literature use the term. Claim 1 as a whole only makes sense when a “computer system” is a computing device, not multiple computing devices and not a computer network.

Under this correct construction, CrowdStrike has no argument and all the grounds fail against each claim.

C. Claim 2: CrowdStrike failed to demonstrate Grounds 1-2 render obvious “monitoring all activity on the first computer system.”

Claim 2 depends from claim 1 and recites “wherein the monitoring further comprises *monitoring all activity on the first computer system.*” Because CrowdStrike has not established that claim 1 is obvious, claim 2 is also not unpatentable. But even setting aside claim 1, CrowdStrike has not met its burden to show that claim 2 is obvious over *Capalik* alone or in combination with *Goel*.

CrowdStrike relies only on *Capalik* for disclosures relating to claim 2 in both Grounds 1 and 2. *See* Pet. 34 (relying only on *Capalik* in Ground 1); 46 (referring only to Ground 1 discussion for claim 2 in Ground 2). Thus, Grounds 1 and 2 rise or fall together for claim 2.

CrowdStrike's argument for claim 2 consists of three sentences:

- “As explained above, *Capalik* teaches monitoring activity on the computer system.”
- “*Capalik* further teaches monitoring all activity *on the VM within the computer system.*”

- “Accordingly, a POSITA would have understood that *Capalik* teaches monitoring all activity on the computer system.”

Pet. 34 (emphasis added).

CrowdStrike's argument fails for two reasons. First, CrowdStrike has not shown that *Capalik* teaches monitoring “all activity on the VM.” Pet. 34. Second, even if *Capalik* did disclose monitoring all activity on the VM, CrowdStrike has not even attempted to show that such disclosure constitutes monitoring “all activity” on decoy computer 502, which CrowdStrike mapped to the “first computer system.”

First, CrowdStrike has not shown that *Capalik* monitors all activity even within its VM. CrowdStrike cites two passages of *Capalik* in support of its argument. In paragraph 59, *Capalik* discloses that “the processing module 508 may generate a signature ***whenever data is found to be transferred through the socket.***” Pet. 34 (quoting *Capalik* [0059]) (emphasis in Petition). But this passage from *Capalik* relates only to monitoring that is (1) triggered by a specific event (receipt of data) at a specific component (a socket/port) of the system and (2) limited to monitoring particular information from that event. *See also* *Capalik* [0027] (“When an attacker connects to a port, and begins interacting with decoy operating system 32, sentinel 34 monitors and captures information *from the connection* including port numbers, data streams, keystrokes, file uploads and any other data transfers.”) (emphasis added); Stavrou ¶70. Monitoring that begins at a network-port/socket

connection is not “monitoring all activity” on the first computer: it omits local activity that occurs absent any socket event, including file and process activities that the '872 patent detects before any network access.

The monitoring of all activity in the '872 patent is a key distinction from GoSecure's earlier *Capalik* application. Stavrou ¶¶29, 68-69. Namely, *Capalik* assumes that the monitored system is uncompromised, and so all activity before “data is found to be transferred through the socket” is *unmonitored*. Stavrou ¶68; *Capalik* [0027] (“*When an attacker connects to a port, and begins interacting with decoy operating system 32, sentinel 34 monitors and captures information from the connection*”) (emphasis added); [0049] (similar). “Generally all relevant data coming through the socket is captured” and “[i]n most cases this means whatever data is received as part of an incoming attack is captured by the sentinel driver.” *Capalik* [0010]. *Capalik*'s attack signature is generated by the processing module “by analyzing all the data *passed through the socket*.” *Capalik* [0013] (emphasis added). Thus, the IDS/IPS of *Capalik* employs signatures based on network activity that the separate IDS/IPS can monitor, as shown, for example, in *Capalik*'s Figure 4. Stavrou ¶37.

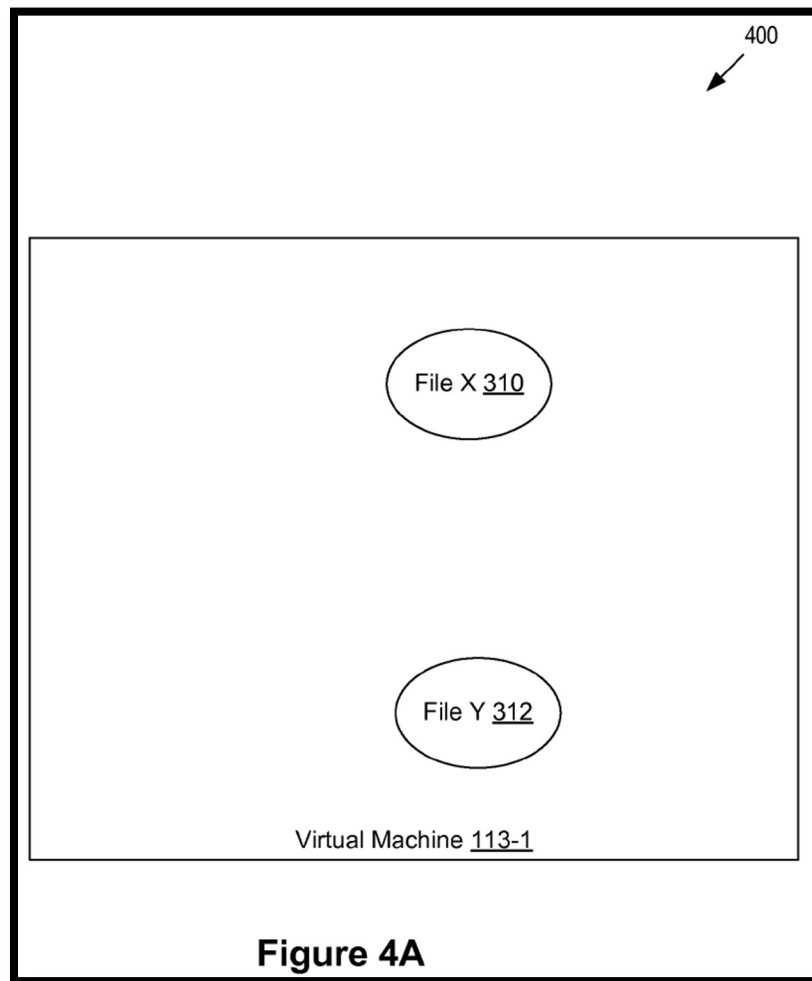
```
<item type="STRING.TCP"> ← 400
  <struct name="STRING.TCP">
    <var name="version" protected="true">4.0</var>
    <array name="signatures">
      <entry dontDelete="true" nda="false" sfr="85">
        <var name="SigName" default="RBOT.CBQ Worm Activity" protected="true" />
        <var name="SIGID" default="5571" protected="true" />
        <var name="SubSig" default="0" protected="true" />
        <var name="AlarmSeverity" default="high" />
        <var name="Enabled" default="True" />
        <var name="EventAction" default="alarm">alarm|reset</var>
        <var name="SigVersion" default="S185" />
        <var name="SigStringInfo" default="RBOT.CBQ" />
        <var name="AlarmThrottle" default="Summarize" />
        <var name="Direction" protected="true" default="ToService" />
        <var name="MinHirs" default="1" />
        <var name="Protocol" default="TCP" />
        <var name="RegexString" protected="true"
default="\xb5\x71\xc4\x43\x1e\x34\xce\x3a\x18\x37\xef\xc3\x22\xa1\x20\x1f\x6c\x10\x8f\x6
8\x3d\x44\xef\x51\x92\x19\x4f\xbc\x46\xe9\x05\xdc" />
        <var name="ServicePorts" default="445" />
        <var name="StorageKey" default="STREAM" />
        <var name="SummaryKey" default="Axxx" />
        <var name="ThrottleInterval" default="15" />
      </entry>
    </array>
  </struct>
</item>
```

Capalik, Fig. 4

Capalik's attack signature is therefore based on monitoring the *socket*, and monitoring activity without such socket activity is not contemplated by *Capalik*. Stavrou ¶¶37-38, 68. At best, *Capalik* can only detect unauthorized activity originating from an outside network connection through a socket.

In contrast, claim 2 requires monitoring *all* activity—regardless of whether any socket is accessed. Figures 4A-4D provide an example showing how monitoring *all* activity allows for detection of activity (and threats) that *Capalik* would not have caught. Specifically, the endpoint device is monitoring all of the activity, so it detects an unauthorized activity through an unauthorized file access. In Figure 4A, “unauthorized activities on the file X 310 and the file Y 312 in the virtual machine

113-1 are detected.” ’872 patent 12:2-4. These unauthorized activities originate from information that *Capalik* does not monitor, such as “checksum values of the file[s]” or “monitoring data execution prevention (DEP) exceptions associated with the file[s].” ’872 patent 12:4-14; Stavrou ¶¶70. Monitoring of these files is not (and does not need to be) triggered by activity at a port or socket. Stavrou ¶¶70, 24-25.



’872 Patent Fig. 4A

In Figure 4B, “process A 308 accesses both the file X 310 and the file Y 312.” ’872 patent 12:15-17. Further, “file X 310 initiates the process B 314.”

'872 patent 12:17-18. Just as the unauthorized activity in Figure 4A, the file accesses in Figure 4B are detected by monitoring in the '872 patent that would not occur in *Capalik*. Stavrou ¶¶26-29.

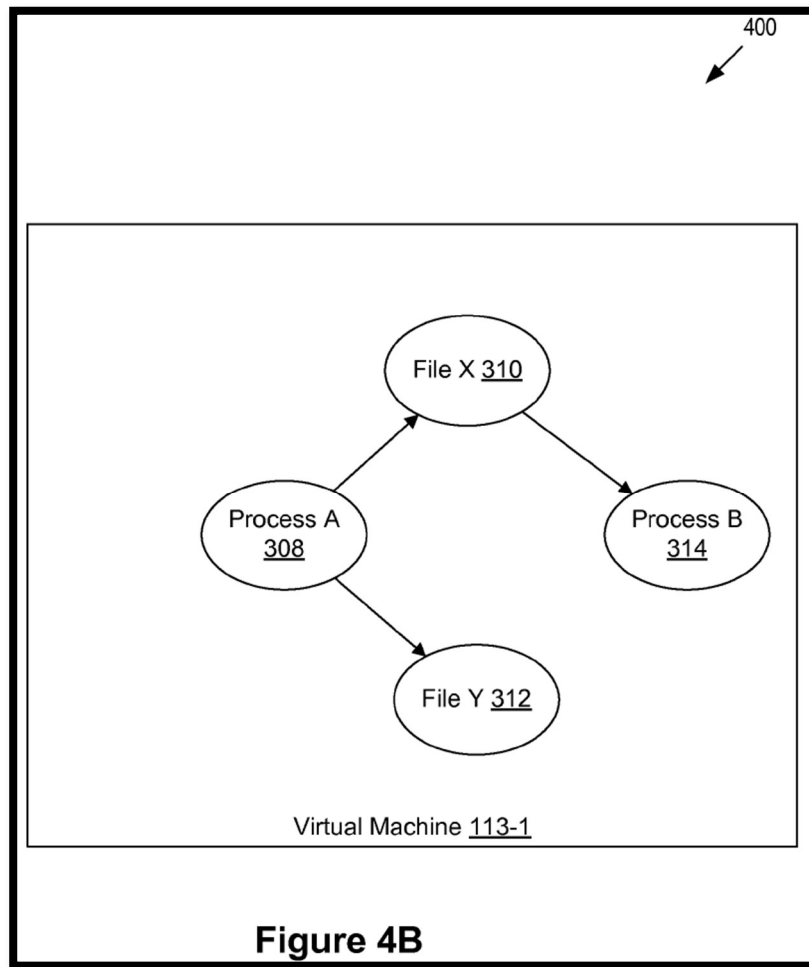


Figure 4B

'872 Patent Fig. 4B

It is not until Figure 4C that the socket is discovered to be involved in unauthorized activity. There, “data from the socket 306 is directed to the process A 308.” '872 patent 12:20-22. In Figure 4D, the socket is shown to “receive[] the packet stream 304.” '872 patent 12:22-23. Notably, in this scenario the packet

stream and socket access were not suspicious and were able to avoid intrusion detection. Stavrou ¶28.

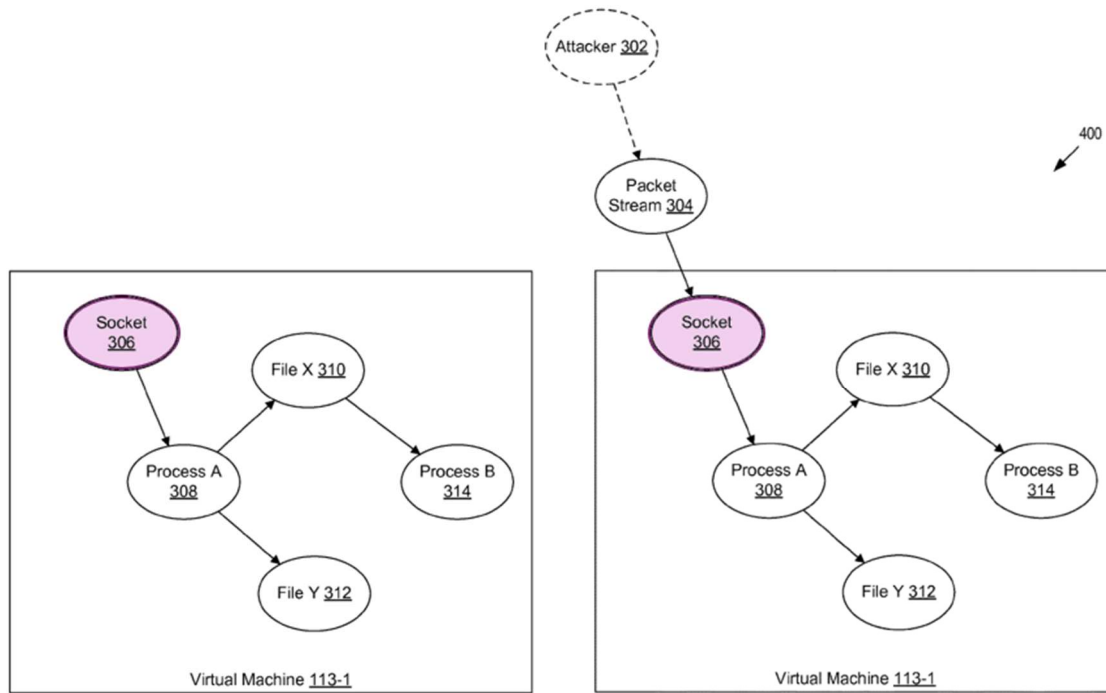


Figure 4C

Figure 4D

Capalik's system would not have detected the unauthorized activity in, e.g., Figure 4A. *Capalik* relies on an IDS/IPS to use its signature library to monitor network activity (and not, e.g., file accesses) to protect the protected devices. Further, *Capalik's* decoy would not have detected unauthorized activity that originates as, e.g., modification of a file. Stavrou ¶¶25-29.

For his part, Dr. Jakobsson stretches beyond *Capalik* to state, without explanation or evidentiary support, that decoys in general “were known to both monitor all activity at the decoy system as well as assume all activity was unauthorized” in order to “be an efficient decoy” and “for exhaustive purposes.”

Ex.1003 ¶146. But neither CrowdStrike nor Dr. Jakobsson propose any modification to *Capalik* with respect to that purported knowledge of a POSITA. To the contrary, the Petition characterizes Dr. Jakobsson's declaration as "explaining that a POSITA would have understood *Capalik's* detailed forensic data monitors all activity *on the VM that begins with a connection* through a VM network port." Pet. 34. Nevertheless, both CrowdStrike and Dr. Jakobsson merely conclude that "*Capalik* teaches monitoring all activity on the computer system." Pet. 34; Ex.1003 ¶146. Dr. Jakobsson's citation to monitoring as being "known" for decoys does not mean that such monitoring is inherent or obvious in *Capalik*, and neither CrowdStrike nor Dr. Jakobsson proffer such an argument.

Moreover, even if CrowdStrike had argued that such monitoring was inherent or obvious (they didn't), Dr. Jakobsson's testimony is insufficient to support either argument. As a threshold issue, Dr. Jakobsson does not try to show that monitoring all activity is inherent in all decoy devices like *Capalik's*. At best, he argues merely that "efficient" and "exhaustive" decoys monitor all activity. Ex.1003 ¶146. That is not inherency.

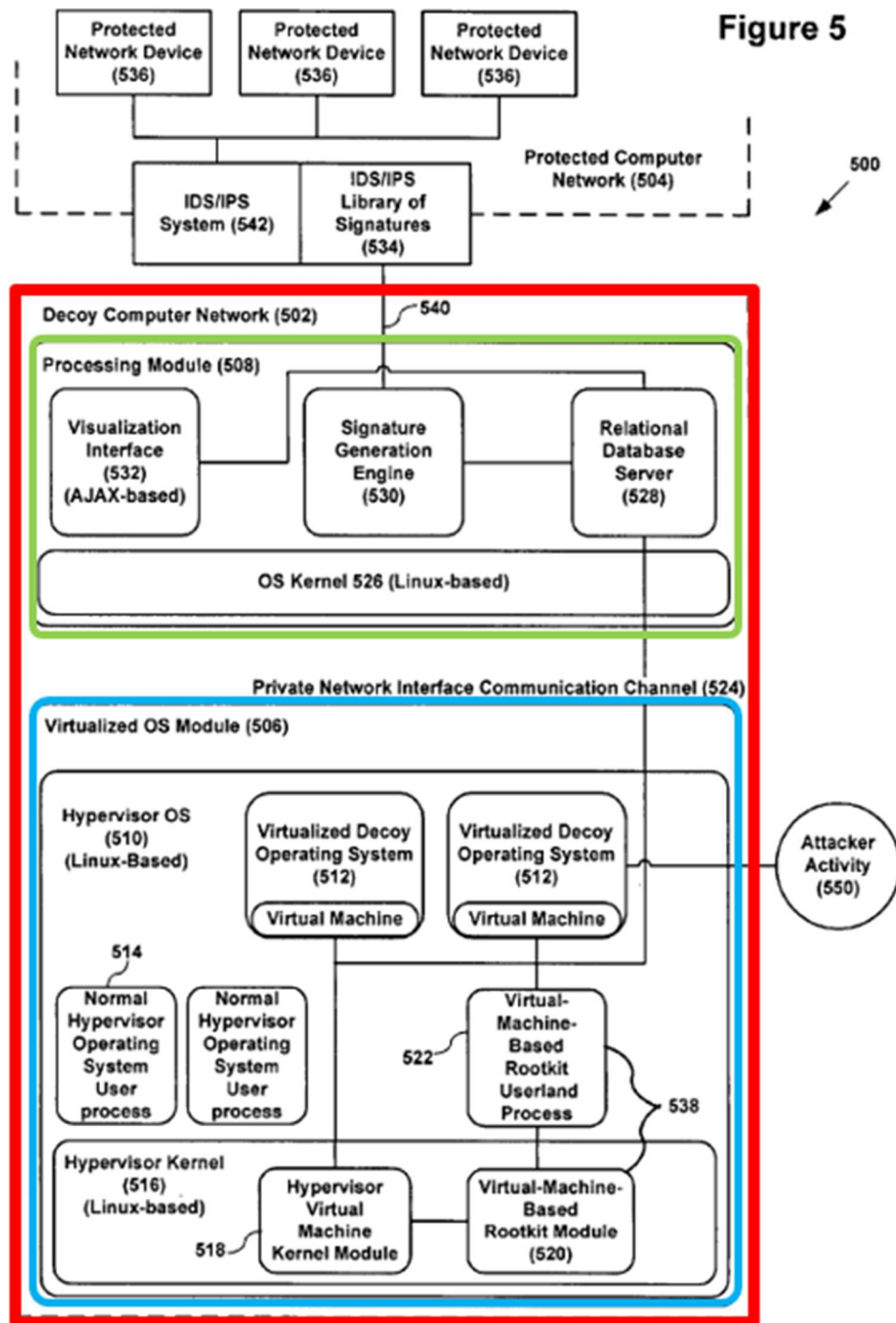
But even further, Dr. Jakobsson's discussion of honeypots is consistent with *Capalik* in that it merely motivates monitoring *network activity at a socket*. According to Dr. Jakobsson, the decoy "act[s] as a fake port" such that "any *activity that reaches the honeypot* is assumed unauthorized." Ex.1003 ¶146. Dr.

Jakobsson's discussion of honeypots provides no motivation for modifying *Capalik* to monitor activity *before* the socket is accessed, or activity that does not involve a socket being accessed. Ex.1003 ¶¶49-54; 146; Stavrou ¶71.

Thus, CrowdStrike's assertion that *Capalik* monitors all activity on the VM is unsupported and contrary to *Capalik*.

But the Petition fails on claim 2 for a ***second*** reason. Even if *Capalik* did disclose monitoring all activity on the VM as CrowdStrike asserts (it doesn't), monitoring all activity "on the VM within the computer system" is ***not*** monitoring all activity "on the first computer system" as required by claim 2. CrowdStrike mapped the claimed "first computer system" to a "decoy computer 502."² Pet. 15 ("[A] POSITA would have understood Decoy Computer 502 is a first computer system ('computer system') attached to a computer network, as claimed."). As shown in annotated Figure 5 below, *Capalik*'s decoy system 502 (**red**) includes both processing module 508 (**green**) and virtualized operating system module 506 (**blue**). *Capalik* [0040]; Stavrou ¶71.

² In fact, CrowdStrike could not map the "first computer system" to the virtual machine to satisfy the claim 1 limitations, including 1[C]. Specifically, claim 1 requires the first computer system to store a data structure in memory, whereas CrowdStrike points to storage outside of the VM for that requirement.



Capalik Fig. 5 (annotated)

Accordingly, for claim 2, the Petition should have shown that *Capalik's* decoy computer 502 is monitored for all activity, not just the VM within the decoy computer. But CrowdStrike does not identify *any* discussion in *Capalik* of monitoring activity within **processing module 508**—which has its own operating system within the decoy 502. *Capalik* [0051]. The processing module 508 is “part of a secure and separate administrative network” from the introspection module 538 and is therefore *trusted* and not monitored by the introspection module. *Capalik* [0051]; Stavrou ¶67. Important and substantial *unmonitored* activity is performed by the processing module 508, including storage of the attack information in relational database server 528, creation of the signature by signature-generation engine 530, and transmission of the signature to the IDS/IPS system. *Capalik* [0052]. CrowdStrike fails to address monitoring of this activity of the Petition's purported “first computer system” (*Capalik's* decoy computer 502). Thus, CrowdStrike has not and cannot establish that *Capalik* discloses monitoring all activity on the claimed “first computer system.” Stavrou ¶71.

In summary, CrowdStrike failed to show that *Capalik* monitors all activity on the decoy computer it identified as the first computer system in claim 1. To the contrary, *Capalik* monitors sockets and activity related to such socket activity. This key distinction results in faster and better detection of unauthorized activity in the

'872 patent. As such, the Board should find that CrowdStrike failed to meet its burden for claim 2.

D. CrowdStrike failed to meet its burden for claims 3-6, 12, 15, and 20-21.

Dependent claims 3-6, 12, and 15 all depend (directly or indirectly) from claim 1. As shown in Sections IV.A-B, CrowdStrike failed to meet its burden to show that claim 1 is obvious over *Capalik*. Accordingly, CrowdStrike also failed to meet its burden for claims 3-6, 12, and 15.

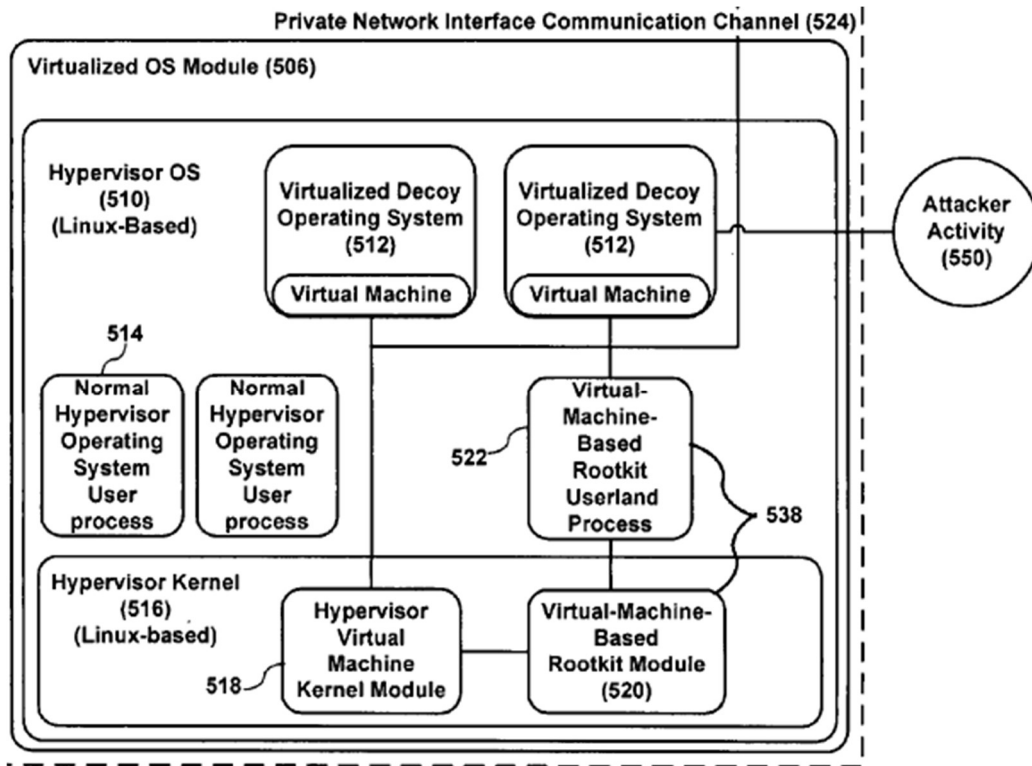
Independent claims 20-21 each recite the same “storing” and “transmitting” limitations of 1[C] and 1[D]. CrowdStrike did not separately address these limitations in its Petition, instead relying on its claim 1 analysis. *See* Pet. 39-40. Accordingly, claims 20-21 are patentable over Ground 1 for the same reasons as claim 1.

V. Ground 2: CrowdStrike failed to show that *Capalik-Goel* renders any claim obvious.

Ground 2 challenges the same claims as Ground 1 in addition to dependent claim 14, which depends from claim 1. CrowdStrike relies entirely on its Ground 1 analysis of *Capalik* for the “data structure” of limitation 1[C] as well as all of limitation 1[D] and claim 2. Pet. 40, 46. Thus, Ground 2 fails for the same reasons as Ground 1 for each challenged claim. Ground 2 also fails for the independent reasons discussed below.

A. A POSITA would not have combined *Capalik* and *Goel*.

A POSITA would not have combined *Capalik* and *Goel*. Undetectability is a key tenet of *Capalik*'s design. Stavrou ¶74. For example, *Capalik* discloses that prior art honeypots “are generally not useful against sophisticated attackers who can detect and effectively avoid the honeynet, never unloading their zero-day attack or payload for the honeynet to capture or analyze.” *Capalik* [0006]. To that end, *Capalik* discloses a decoy system that records and analyzes certain attacker activity “without knowledge of the attacker.” *Capalik* [0024], [0045]. One way that *Capalik* designs its system to be undetectable is through the use of a virtual machine that runs a virtualized decoy operating system. *Capalik* does not modify the decoy operating system. Instead, *Capalik* places a rootkit module in the hypervisor kernel outside of the virtual machine. *Capalik* [0042]; Fig. 5.



Capalik Fig. 5 (excerpt)

As a result of this design, the rootkit “remain[s] hidden from the instigator of the attack, with no discoverable impact on the decoy operating systems’ performance.” *Capalik* [0047]. Otherwise, *Capalik*’s system would be “not useful” against sophisticated attacks as prior art honeynets it disparages. *Capalik* [0006]; Stavrou ¶76.

A POSITA would not have been motivated to combine *Capalik* with *Goel* because *Goel* would have undermined *Capalik*’s requirement that activity logging be undetectable, rendering the system “not useful” against sophisticated attacks. *Capalik* [0006]. Unlike *Capalik*, *Goel* does not disclose a decoy system and is not concerned with detectability. See *Goel* 2 (“Design Goals” not mentioning

detectability); Stavrou ¶¶75-77. *Goel* is primarily concerned with logging enough information to enable a user—post facto—“to reconstruct all currently known attacks and all future attacks.” *Goel* 2; *see also id.* 1 (discussing post-facto nature of the analysis).

It is therefore unsurprising that *Goel*'s system would have been readily detectable by an attacker. Stavrou ¶75. Unlike *Capalik*'s system that has “no discoverable impact on the decoy operating systems' performance” to help it remain hidden, *Goel* explicitly discloses a 7%-36% decrease in web server throughput. *Goel* 6-7, Table 3. This disclosure alone shows the detectability of *Goel*'s logging system compared to *Capalik*'s. Stavrou ¶75. But even further, *Goel*'s approach to logging—modifying the operating system kernel with hooks to log system-call activity—is also detectable by attackers. Stavrou ¶75; Ex.2050(HookFinder) 1 (2007 paper discussing “several tools” that detect kernel hooks and proposing a framework to detect installation of “any hooks into the system”). This is the kind of system that *Capalik* teaches away from, describing detectable honeypots as “not useful” against sophisticated attacks. *Capalik* [0006]; Stavrou ¶¶74, 76-77. Also, combining *Capalik* with *Goel* would destroy the undetectability of activity logging, rendering the decoy system inoperable for its intended purpose.

CrowdStrike explicitly proposes incorporating *Goel*'s logging into *Capalik* but does not address these fundamental differences in the respective systems. Pet.

43-46. To the contrary, CrowdStrike admits that *Goel's* logging results in an “increase in processing power to modify *Capalik* per *Goel's* teachings” but dismisses that concern as apparently outweighed by “security improvements.” Pet. 44-45. CrowdStrike ignores *Capalik's* teaching that those same performance degradations would make the monitoring system detectable to an attacker and therefore render *Capalik's* decoy ineffective. Stavrou ¶¶75-76; *Capalik* [0006]. CrowdStrike further admits that “*Goel* is silent on the use of a decoy system” and for good measure: *Goel's* system would be an ineffective decoy as taught by *Capalik*. Stavrou ¶¶74-76; *Capalik* [0006]. And though CrowdStrike asserts that the authors were “working on implementing the Forensix system on a VM” (Pet. 45 (citing *Goel* 2)), *Goel* provides no details about such a system and there is no indication that the authors had detectability in mind for that implementation, either. Stavrou ¶79.

CrowdStrike asserts that a POSITA would have had a reasonable expectation of success implementing *Goel's* monitoring in *Capalik's* VMM/hypervisor. Pet. 46. Not so. *Goel* discloses monitoring from the kernel of the operating system being attacked—not a hypervisor outside of that operating system. Stavrou ¶77. *Goel* specifically teaches that “capturing activity *at this point* thus addresses the problem [associated with prior approaches] at a more fundamental, unified level.” *Goel* 3. A POSITA would not have had any reasonable expectation of achieving *Goel's* level

of logging, or any of its Design Goals, by capturing activity from a hypervisor outside of the system, or at any other point. Stavrou ¶¶77-79.

Accordingly, a POSITA would not have combined *Capalik* and *Goel* and CrowdStrike failed to show that Ground 2 renders any claim obvious (including claim 14, which is not addressed in Ground 1).

VI. Ground 3: CrowdStrike failed to show that claims 7-11, 14, 16, and 18-19 are obvious over a combination of *Capalik*, *Goel*, and *King*.

Ground 3 challenges dependent claims 7-11, 14, 16, and 18-19 based on “*Capalik* alone or in view of *Goel* and further in view of *King*.” Pet. 48. CrowdStrike does not rely on *King* to cure any of the deficiencies identified above for Grounds 1 or 2. Thus, Ground 3 fails for the same reasons as Grounds 1 and 2 for each challenged dependent claim due to their dependence on claim 1. Ground 3 also fails for the independent reasons discussed below.

A. Claim 8: CrowdStrike failed to demonstrate that claim 8 is obvious.

Claim 8 depends from claim 7, which recites “graphically displaying at least a subset of the stored activities.” Claim 8 further recites “wherein the displaying includes visually distinguishing at least a subset of the associations.”

For the “displaying” of claim 7, CrowdStrike relies on *King*’s GraphGen process, which displays the dependency graph shown in Fig. 3.2(c). Pet. 49-52. For clarity, although CrowdStrike cites to *King*’s Fig. 3.2(b) to demonstrate that the

displayed graph is a subset of the stored activities, *King* does not contemplate actually displaying Fig. 3.2(b). *See King*³ 24 (stating that “only the relevant portion of the graph” is presented), 27 (describing Fig. 3.2(c) as the graph “generated by GraphGen” and characterizing it as the “useful picture of the events”); Pet. 51 (“*King*’s filtered graph is then displayed to a system administrator”); Stavrou ¶84.

For the “visually distinguishing” of claim 8, CrowdStrike again relies on *King*’s dependency graph, shown in Fig. 3.2(c). Pet. 55. According to CrowdStrike, this graph displays “only malignant associations,” and *King* visually distinguishes a subset of the associations by merely displaying them and not displaying others.⁴

Merely displaying a subset does not “visually distinguish” the subset. In addition, merely displaying the subset cannot satisfy claim 8 at least because claim 7, from which claim 8 depends, already recites displaying the subset. *See AK Steel Corp. v. Sollac and Ugine*, 344 F.3d 1234, 1242 (Fed. Cir. 2003) (“[D]ependent claims are presumed to be of narrower scope than the independent claims from which they depend.”); 35 U.S.C. §112(d) (“a claim in dependent form shall contain a reference to a claim previously set forth and then specify a further limitation of the

³ Citations to *King* refer to pagination added by CrowdStrike.

⁴ CrowdStrike annotates *King*’s Fig. 3.2 to add color to different arrows. That color is not present in *King*. *See King* 27.

subject matter claimed.”). That would be improper, especially here where CrowdStrike admits that the '872 patent discloses specific ways of “visually distinguishing” associations—e.g., by representing malignant associations with a bold arrow and benign associations with a regular arrow. Pet. 55 (citing '872 patent 12:32-41).

CrowdStrike has no other argument for claim 8. Accordingly, Ground 3 fails and claim 8 is not unpatentable.

B. Claim 16: CrowdStrike failed to demonstrate that claim 16 is obvious.

Claim 16 depends from claim 1 and recites “determining an activity status for the stored events, the activity status comprising one of: a complete status... and an incomplete status....” The complete status represents that “a source of unauthorized activities is identified” and “each activity target in the stored activities is associated with the source through one or more associations.” Conversely, an incomplete status represents that the source of unauthorized activities is *not* identified, *or* at least one activity target is not associated with the source through one or more associations.

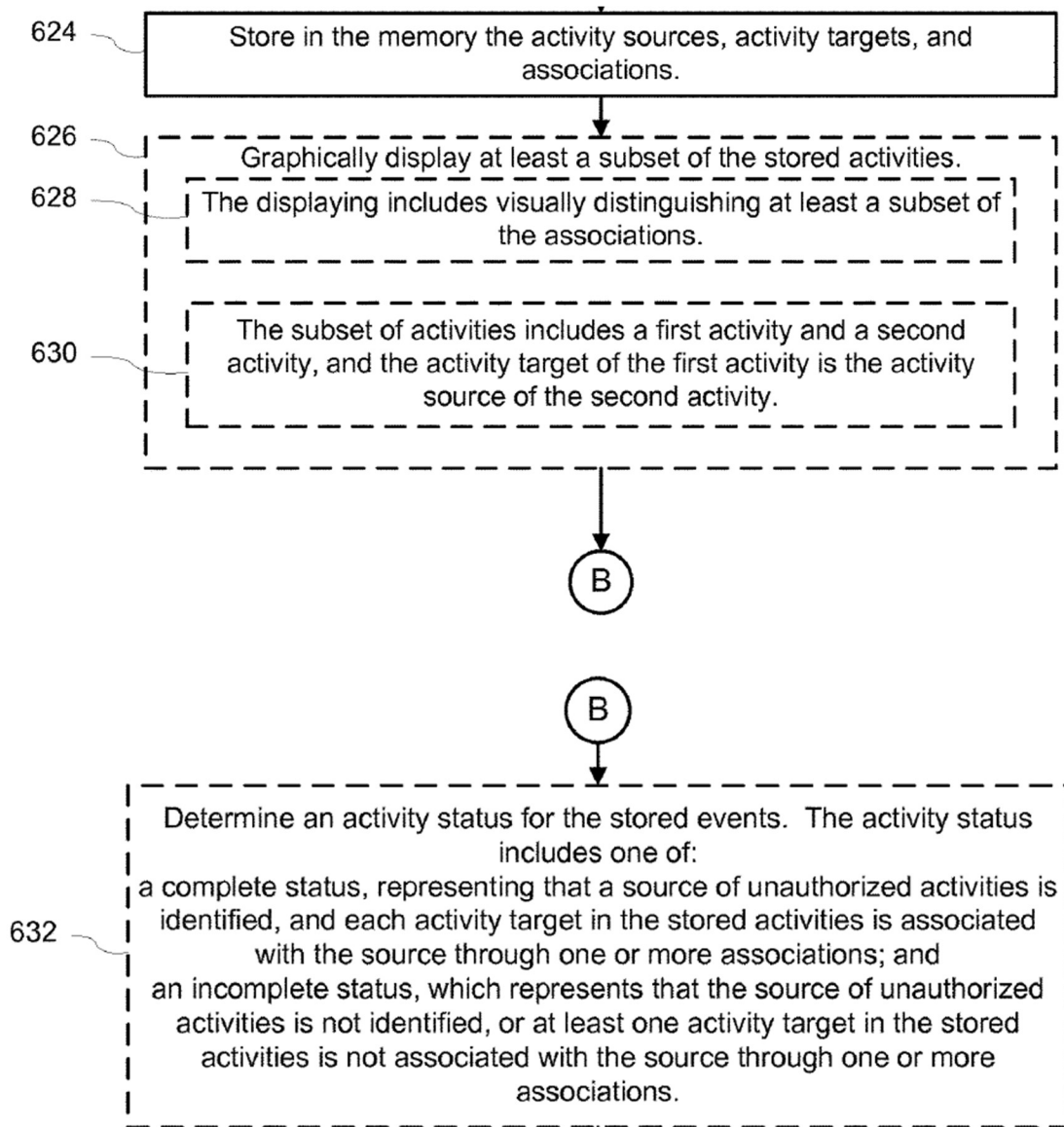
CrowdStrike asserts claim 16 is satisfied and points to *King's* display of the dependency graph in Fig. 3.2(c). Pet. 63-64. But CrowdStrike makes no attempt to explain why this graph allegedly shows any *determination* of an activity status, as required by the claim. Indeed, the Petition's only reference to this concept is in the last sentence, which simply concludes that the references teach “determining a

complete activity status for the stored events.” Pet. 64. This is insufficient to satisfy CrowdStrike’s burden.

Moreover, the mere display of *King’s* graph cannot satisfy the claim. First, claim 16 requires an affirmative step of determining status—complete or incomplete—for the stored events. The claim does not recite displaying a graph from which a user may infer that the activity status is complete. Rather, the “determining” step is part of the “computer implemented method” of claim 16. By the plain language of the claim, a computer must determine a status for the “stored events.” *King* does not disclose this express, binary status determination. Second, *King’s* Fig. 3.2(c) does not display a “complete status” because its graph does not represent that (1) a source of unauthorized activities is identified and (2) each activity target in the stored activities is associated with the source through one or more associations. *King’s* GraphGen reconstructs dependency graphs based around a specific “detection point, which is file X at time 10.” Ex.1006 (*King*) at 26-27. A “filtered” graph in *King* is merely the result of excluding from the graph data deemed irrelevant to the chosen detection point; it is not a determination whether each target in the stored activities are associated to a source as required by claim 16.

CrowdStrike implies that the ’872 patent describes mere depiction of a graph as satisfying the determining step of claim 16. Pet. 62 (citing ’872 patent 17:23-28). Not so. The ’872 patent explicitly determining an activity status as a step *wholly*

separate from graphical display. In Figure 6B, step 626 is “[g]raphically display at least a subset of the stored activities,” whereas step 632 in Figure 6C is “[d]etermine an activity status for the stored events.”



'872 patent Figs. 6B (top) and 6C (bottom), excerpted

Moreover, Figure 5A illustrates a Status 506 as part of the data structure for unauthorized activity data. '872 patent 12:45-67. Thus, the specification

demonstrates that mere graphical display of a graph is not an affirmative determination of activity status.

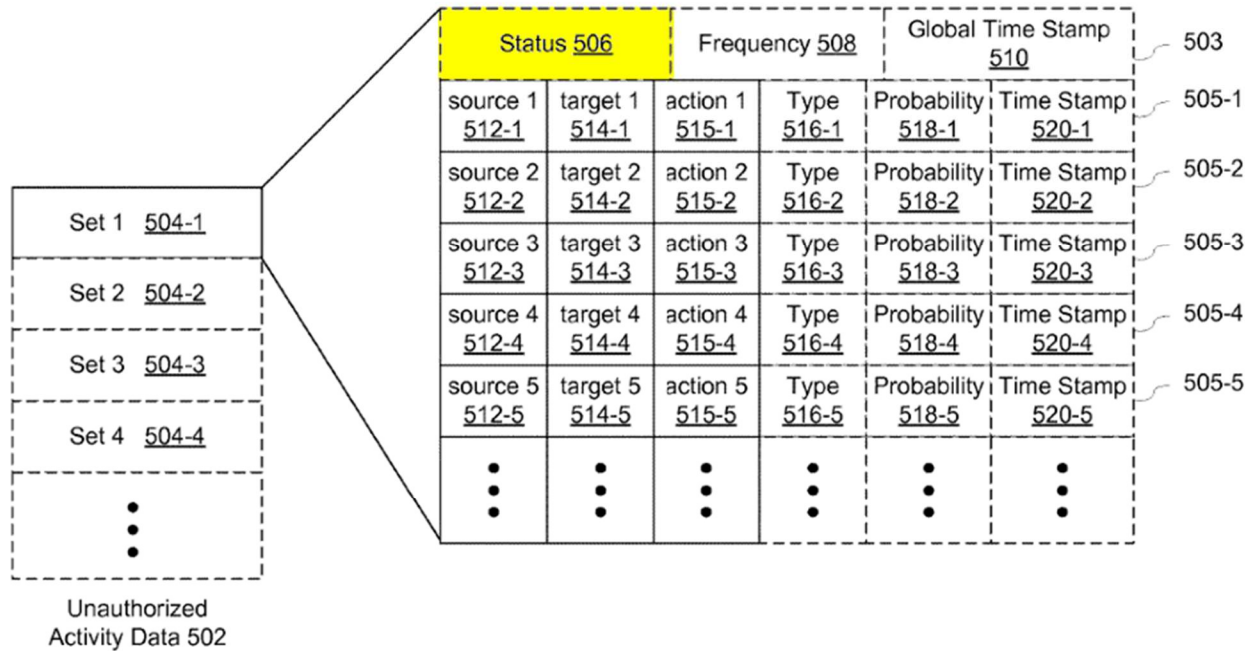


Figure 5A

'872 Patent, Fig. 5A

Accordingly, Ground 3 fails for claim 16.

C. Claim 19: CrowdStrike failed to demonstrate that claim 19 is obvious.

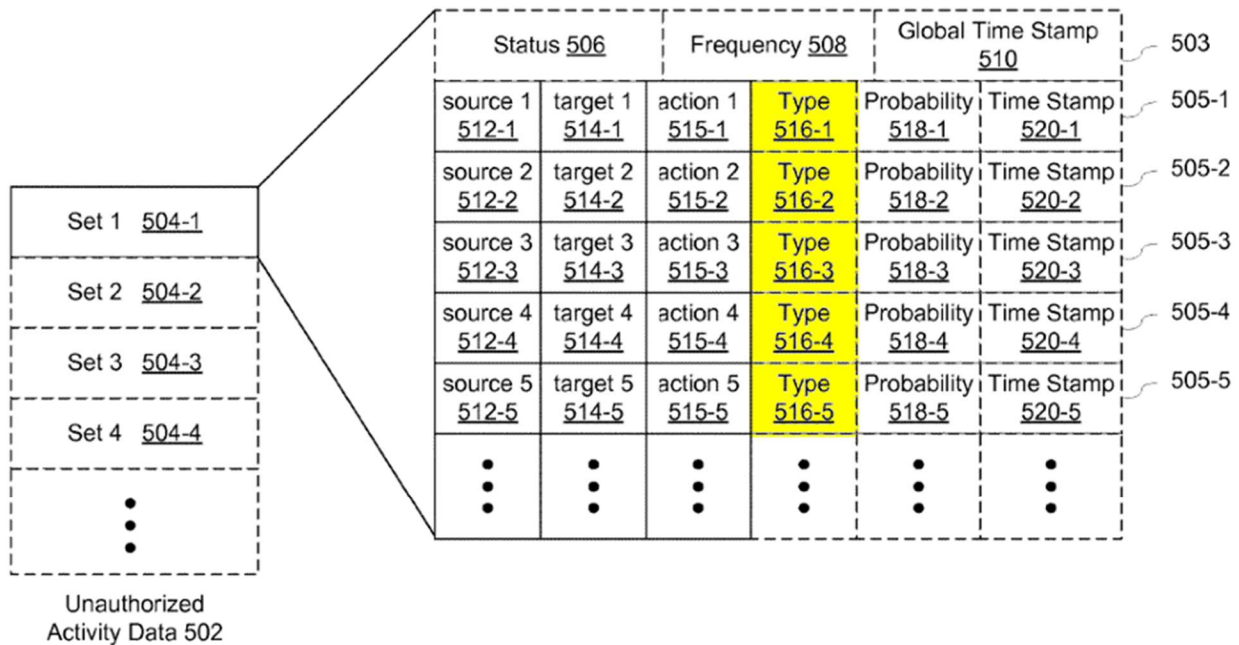
Claim 19 depends from claim 1 and recites “wherein the association includes a type that represents whether or not the activity source has modified the activity target to perform the unauthorized activities.”

CrowdStrike relies solely on *King* for disclosure of this limitation. Pet. 64-66. Specifically, CrowdStrike relies on *King*'s filtered dependency graph that does

not include “any associations that did not contributed... to the unauthorized activity.” Pet. 65. Based solely on this graph, CrowdStrike asserts that “[a] POSITA would have understood that the associations remaining after this filtering are of a type that represents a source modifying a target to perform unauthorized activities either directly... or indirectly.” Pet. 65-66.

This argument fails for two reasons. First, CrowdStrike argues that the associations displayed in the graph are all of one type, but this fails to address the claim language, which specifies that “the association includes a type that represents whether or not” a particular condition has been satisfied.

Second, claim 1 (from which claim 19 depends) requires “storing in the memory a data structure that identifies... the associations for the plurality of activities.” As discussed in 1[C], *Capalik* has no data structure, let alone one that identifies the source, target, and associations for the plurality of activities. Moreover, under claim 19, an association identified by the stored data structure includes a “type that represents whether or not the activity source has modified the activity target to perform the unauthorized activities.” For example, Fig. 5A of the '872 patent shows an embodiment of a data structure that identifies the associations and their types (highlighted).

**Figure 5A**

CrowdStrike did not even attempt to address how the Ground 3 combinations disclose storing a “data structure” that identifies the association “type” required by claim 19. That failure dooms Ground 3 for claim 19.

CrowdStrike does not assert that the alleged “data structure” in *Capalik* includes the “type” required by claim 19. Nor can CrowdStrike rely on its claim 1 analysis for inclusion of this “type” in the allegedly stored data structure because CrowdStrike did not address the “type” at all in Ground 1. In fact, CrowdStrike expressly argued that a “type” is *not* required in claim 1. Pet. 7 (relying on claim 19 to argue that “the claims demand that ‘association’ and ‘association type’ have distinct meanings.”). Nor does *King* disclose storing a “type” attribute for associations in a data structure.

Accordingly, Ground 3 fails for claim 19.

VII. Ground 4: CrowdStrike failed to show that claim 13 is obvious.

CrowdStrike fails to show claim 13 is obvious. First, claim 13 depends indirectly from claim 1, and Petitioner relies on the deficient analysis of Grounds 1-2. Second, Petitioner fails to show that *Pike* teaches the additionally-recited limitations of this claim. Claim 13 depends from claim 12, which recites “identifying activity sources that are affected by unauthorized activities.” Claim 13 further recites “wherein the identifying activity sources... includes identifying activity sources that request to access a portion of memory that is set as non-executable.” CrowdStrike cites to *Pike* for claim 13. Pet. 66-68. But *Pike* does not disclose the features of claim 13.

Pike is directed to a particular form of intrusion detection that does not create or rely on signatures. *Pike* Title (“Security method for detecting intrusions...”), 3:10-12 (“Mechanisms have been developed for configuring an [IDS].”), 3:23-29 (disclosed method does not use signatures). It is therefore unsurprising that *Pike* is only concerned with scenarios where “the potentially dangerous address is accessed.” *Pike* 8:65-67. CrowdStrike cites to these scenarios. Pet. 67 (“*Pike* discloses several techniques to trigger an action when memory of the target system is accessed.”).

But CrowdStrike is unable to find any disclosure in *Pike* of identifying sources “that *request to access* a portion of memory that is set as non-executable” as recited in claim 13. At best, *Pike* identifies a source that accesses such memory, identifying an actual intrusion but missing sources that merely request such access. CrowdStrike ignores this difference between *Pike* and claim 13.

Accordingly, Ground 4 fails against claim 13.

VIII. Ground 5: CrowdStrike failed to meet its burden for claim 17.

As shown above for Grounds 1-2, CrowdStrike failed to meet its burden to show that claim 1 is unpatentable over *Capalik* alone or in view of *Goel*. Ground 5 relies on the deficient analysis of Grounds 1-2 and does not rely on *Farley* to cure the deficiencies of Grounds 1-2. *See* Pet. 69-74. Accordingly, Ground 5 fails and CrowdStrike has not met its burden to show that claim 17 is obvious over *Capalik*, *Goel*, and *Farley*.

IX. CrowdStrike's copying and resultant commercial success favor non-obviousness.

A. CrowdStrike principals had access to GoSecure's non-public technology.

CrowdStrike had detailed, intimate knowledge of GoSecure's non-public information from two key GoSecure personnel: (1) Dmitri Alperovitch, who served on GoSecure's Board of Directors for three months before co-founding CrowdStrike as its Chief Technology Officer; and (2) Jeremy Gould, whom Mr. Alperovitch

recruited to work as an engineer at GoSecure for three months before bringing him to CrowdStrike.

In June 2011, GoSecure⁵ filed U.S. Application 13/163,590 (“the ’590 application”),⁶ which issued as U.S. Patent 9,106,697 (“the ’697 patent”). In November 2011, shortly before the ’590 application was published, Dmitri Alperovitch joined GoSecure as a member of its Board of Directors. *See, e.g.*, Ex.2012 at 1. Just three months later, Mr. Alperovitch co-founded CrowdStrike and became its Chief Technology Officer (CTO). Ex.2013 at 1. Mr. Alperovitch stayed on GoSecure’s Board of Directors for three additional months after becoming CTO of CrowdStrike, until May 2012. Ex.2105.

During his time on GoSecure’s Board of Directors, Mr. Alperovitch gained detailed knowledge about GoSecure’s technology, including detailed information about GoSecure’s pending patent applications. For example, Mr. Alperovitch attended a GoSecure Board Meeting in February 2012, where a slide deck presented details about GoSecure’s pending patent applications. *See, e.g.*, Ex.2100 at 36. Among these patent applications is the ’590 application, to which the ’872 patent

⁵ Here, “GoSecure” refers also to the company’s previous names, CounterTack and NeuralIQ.

⁶ The ’872 patent claims priority to the ’590 application.

claims priority. *Id.* at 36; Ex.1001 Cover, element (63) (“Related U.S. Application Data”). The minutes from that meeting listed Mr. Alperovitch as present and stated that CounterTack’s CTO “reviewed with the Board the Corporation’s patent portfolio and the key claims covered by such patents” as well as “product development plans.” Ex.2106 at 1. As a Board Member with a technical background, Mr. Alperovitch was also frequently invited to attend meetings about the technical direction of GoSecure’s products. *See* Ex.2101; Ex.2012 at 1 (describing Mr. Alperovitch’s technical background).

Mr. Alperovitch recommended that GoSecure hire Mr. Jeremy Gould to a senior engineering position at GoSecure in or around December 2011. *See, e.g.*, Ex.2102; Ex.2103. Mr. Gould worked at GoSecure for around 3 months, during which time he had broad access to GoSecure’s code repository and worked closely with GoSecure’s other engineers including on EventHorizon, GoSecure’s Endpoint Detection Response (EDR) product. Ex.2104. In March 2012, Mr. Gould left GoSecure and accepted a position as a senior engineer at CrowdStrike. Ex.2104; Ex.2014 at 1.

Thus, from the day of its launch and for months thereafter, CrowdStrike had access to GoSecure’s confidential information, including patent strategy and filings, code, and product strategy. In particular, CrowdStrike had access not only to the ’590 application, but also to the confidential strategic discussions about the

importance of GoSecure's patents and key claims. *See Liqwd, Inc. v. L'Oreal USA, Inc.*, 941 F.3d 1133, 1139 (Fed. Cir. 2019) (finding copying from sharing a patent application and confidential meetings under a non-disclosure agreement).

B. CrowdStrike's Falcon Platform copies the applications leading to the '872 patent.

At CrowdStrike, Mr. Alperovitch and Mr. Gould were each involved in creating CrowdStrike's Falcon Platform. CrowdStrike launched the Falcon Platform in mid-2013 as CrowdStrike's first and only technology offering at the time. Ex.2015 at 1 (Technology: CrowdStrike Falcon Platform).

As co-founder and CTO, Mr. Alperovitch led the technology direction of the Falcon Platform. *See* Ex.2015 at 1 (CrowdStrike "Technology Approach" including video of Mr. Alperovitch "talk[ing] technology."); Ex.2019 at 2 (Mr. Alperovitch leads the Technology team). Mr. Alperovitch remained CrowdStrike's CTO until 2020, well after the development and launch of the Falcon Platform. Ex.2018 at 2. Mr. Gould also stayed past the Falcon Platform's launch. Ex.2014 at 1.

From its launch, the Falcon Platform⁷ contained a combination of features disclosed in the '590 application and claimed in the '872 patent, including several key distinctions over *Capalik*. For example:

⁷ The Falcon Platform included Falcon Host, advertised as a "[c]loud-managed host-based adversary detection application." Ex.2015 at 1.

- **1[C]** The Falcon Platform stored a data structure identifying sources, targets, and associations as recited in 1[C]: “The Falcon Platform has a situational model that keeps track of all the relationships and context between all these different processes in a graph database.” Ex.2017 at 2:02; Stavrou ¶102.
- **1[D]** The Falcon Platform transmitted to other computer systems information identifying the sources, targets, and associations for preventing future attacks against those computer systems: The Falcon Platform “[s]hares information across an entire community of like-minded trusted defenders.” Ex.2015 at 1; Stavrou ¶103.
- **Claim 2** The Falcon Platform monitored all activity on a computer system: The Falcon Host “track[s] all execution on all hosts at all times.” Ex.2016 at 1. The Falcon Host further “provides maximum visibility into all adversary activities.” Ex.2016 at 1. The platform provides visibility by “very efficiently record[ing] everything that goes on in the environment.” Ex.2017 at 0:45-0:52; Stavrou ¶104.

The Falcon Platform performed the various other features recited in claim 1 as well. For example, the Falcon Platform “[i]dentifies malware and provides attribution of the attackers” (Ex.2015 at 1) like the computer-implemented method of claim 1, which “identif[ies] unauthorized activities on a first computer system

attached to a computer network.” The Falcon Platform **1[A]** monitored activity in the first computer system (*see, e.g.*, Ex.2016 at 1) and **1[B]** identified a plurality of activities being performed at the first computer system, wherein each of the activities includes an activity source, an activity target, and an association between the activity source and the activity target. Ex.2016 at 1 (“As opposed to focusing on malware signatures... Falcon Host instead... provides real-time detection by focusing on what the attacker is doing, as opposed to looking for a specific, easily changeable indicator used in an attack.”); Ex.2017 at 1:42-2:11 (“The Falcon sensor allows us to track over 100 security-related events, including things like process creation, drivers being loaded, registry modifications..., etc. But we need more than just this detail, we need the context as well. The Falcon Platform has a situational model that keeps track of all the relationships and contexts between all these different processes in a graph database.”); Stavrou ¶¶99-101. Falcon Host, which CrowdStrike also called Threat Protect or Threat Protection, included an “on-premise host-based sensor” that “shadows, captures, and correlates low-level operating system events to instantly identify the adversary tradecraft and activity.” Ex.2016 at 1.

CrowdStrike's copying has nexus to this combination of features. For example, CrowdStrike advertised that “Falcon Host's highly efficient real-time *monitoring of all system activity is the only security solution that provides*

maximum visibility into all adversary activities.” Ex.2016 at 1. This “*unparalleled visibility*” derives from both the ability to monitor all system activity *and* the storage of a data structure identifying targets, sources, and associations—i.e., the “situational model that keeps track of all the relationships and context between all these different processes in a graph database.” Ex.2017 2:03-2:21. CrowdStrike further touted its ability to “continuously monitor, share, and mitigate” attacks and its “unmatched... intelligence dissemination capabilities” including by “[s]har[ing] information across an entire community of like-minded trusted defenders.” Ex.2015 at 1. According to a January 2013 video by CrowdStrike, “I think *the product is a game changer primarily because it focuses on the concept of the community helping each other*.... [People] need to basically share indicators... of what the bad guys are doing, so as the bad guys and the adversaries use these techniques... we can actually protect ourselves.” Ex.2051 at 0:02-0:42. When CrowdStrike became a public company in 2019, the first slide of its prospectus stated that “[w]e turned the tables on the adversaries by taking a *fundamentally new approach that leverages the network effects of crowdsourced data* applied to modern technologies such as artificial intelligence, or AI, cloud computing, *and graph databases.*” Ex.2035 at 6.

It is no coincidence that CrowdStrike was founded mere months after Mr. Alperovitch joined GoSecure's Board of Directors. Nor is it a coincidence that

several key features of CrowdStrike's initial Falcon Platform were also GoSecure's innovations as claimed in the '872 patent. Mr. Alperovitch and Mr. Gould's access to GoSecure's technology and patents, followed immediately by their critical roles in developing the Falcon Platform at CrowdStrike, demonstrate that CrowdStrike copied the techniques claimed in the '872 patent.

C. CrowdStrike's industry praise and commercial success have nexus with the claimed combination of features.

CrowdStrike received industry praise and commercial success from its Falcon Platform. In 2014, CrowdStrike received the SINET 16 Innovator Award (Ex.2027), in early 2016 received CRN's Security 100 Award (Ex.2020), and was named a 2015 Technology Pioneer by the World Economic Forum (Ex.2021). CrowdStrike's website included videos of customer case studies, most of which are no longer available nor archived. Ex.2022. CrowdStrike also reported a \$100 million financing round in 2015. Ex.2023. CrowdStrike's CEO credited the investment to the "unique architectural approach" of the Falcon Platform which "was designed from the beginning to detect advanced malware and adversary activity, prevent zero day exploitation, and dramatically reduce time to remediation during incident response by allowing effortless real-time and historical searches of hundreds of thousands of systems." Ex.2024. He also credited the Falcon Platform's "unique approach to identify unknown threats" including "looking for Indicators of Attack (IOA) vs. Indicators of Compromise (IOC)." *Id.* "[C]ustomer validation was a

critical criterion for driving the investment.” *Id.* By 2015, CrowdStrike experienced a 550% growth rate of revenue for the Falcon Platform. *Id.* CrowdStrike boasted customers including 3 of the 10 largest companies in the world. *Id.* By the fourth quarter of 2017, CrowdStrike had \$59M in annual recurring revenue, which increased to \$313M just two years later. Ex.2035 at 3. As discussed above, with CrowdStrike's repeated touting of the claimed combination of features in its marketing material demonstrates a nexus between CrowdStrike's success and the claims, in particular with features 1[C], 1[D], and claim 2 (and the combination thereof).

X. Conclusion

For the reasons set forth above, the Board should deny the Petition for *inter partes* review.

Respectfully submitted,

Dated: December 12, 2025

By: /s/ Steven W. Peters

Steven W. Peters (Reg. No. 73,193)
speters@cgsh.com
CLEARY, GOTTlieb, STEEN & HAMILTON LLP
2112 Pennsylvania Ave., NW
Washington, D.C. 20036
Telephone: 202.974.1813

Counsel for Patent Owner
GoSecure, Inc.

CERTIFICATE OF COMPLIANCE WITH 37 C.F.R. § 42.24

I hereby certify that this Patent Owner's Response complies with the word count limitation of 37 C.F.R. § 42.24(b)(1) because the Patent Owner's Response contains 13,240 words using Microsoft Word's counting feature, excluding the cover page, signature block, and the parts of the Patent Owner's Response exempted by 37 C.F.R. § 42.24(b)(1).

By: /s/ Steven W. Peters

Steven W. Peters (Reg. No. 73,193)
speters@cgsh.com
CLEARY, GOTTlieb, STEEN & HAMILTON LLP
2112 Pennsylvania Ave., NW
Washington, D.C. 20036
Telephone: 202.974.1813

*Counsel for Patent Owner
GoSecure, Inc.*

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6(e), I certify that on this 12th day of December, 2025, a true and correct copy of the foregoing **Patent Owner's Response and Exhibits 2031-2052** were served by electronic mail on Petitioner's lead and backup counsel at the following email addresses:

Paul R. Hart, (Reg. No. 59,646)
Paul.Hart@eriseip.com
PTAB@eriseip.com
ERISE IP, P.A.
717 17th St., Suite 1400
Denver, Colorado 80202
Telephone: 913.777.5600
Fax: 913.777.5601

Adam P. Seitz (Reg. No. 52,206)
Adam.Seitz@eriseip.com
ERISE IP, P.A.
7015 College Blvd., Suite 700
Overland Park, Kansas 66211
Telephone: 913.777.5600
Fax: 913.777.5601

By: /s/ Steven W. Peters

Steven W. Peters (Reg. No. 73,193)
speters@cgsh.com
CLEARY, GOTTlieb, STEEN & HAMILTON LLP
2112 Pennsylvania Ave., NW
Washington, D.C. 20036
Telephone: 202.974.1813

Counsel for Patent Owner
GoSecure, Inc.