

Filed on behalf of: GoSecure, Inc.

Entered: February 13, 2025

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CROWDSTRIKE, INC.,
Petitioner,

v.

GOSECURE, INC.,
Patent Owner.

Case IPR2025-00069
Patent 9,106,697

**PATENT OWNER'S
PRELIMINARY RESPONSE**

TABLE OF CONTENTS

I.	Introduction.....	1
II.	Background.....	1
	A. The '697 Patent (Ex. 1001)	1
	B. Prosecution History	11
	C. Person of Ordinary Skill in the Art (“POSITA”).....	12
	D. Claim Construction	12
	E. Overview of Petition	13
	F. Overview of Capalik (Ex. 1004)	14
III.	Argument	19
IV.	Conclusion	26

TABLE OF AUTHORITIES**Page(s)****CASES**

<i>Laitram Corp. v. NEC Corp.</i> , 163 F.3d 1342 (Fed. Cir. 1998)	24
<i>Realtime Data, LLC v. Iancu</i> , 912 F.3d 1368 (Fed. Cir. 2019)	13
<i>Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.</i> , 200 F.3d 795 (Fed. Cir. 1999)	13

Exhibit List

Exhibit	Description
2001-2009	Reserved
2010	Disclaimer Pursuant to 37 C.F.R. 1.321(a)

I. Introduction

Every ground presented by CrowdStrike, Inc. (“Petitioner”) in its Petition for *inter parties review* of U.S. Patent No. 9,106,697 (“’697 patent”) fails to establish a reasonable likelihood of success with respect to any of claims 1-18 or 21-23 of the ’697 patent.¹ Claim 1 of the ’697 patent, from which each of the remaining claims depends, requires the claimed “virtual machine monitor” to be involved in generating a fingerprint of an attack. Each of Petitioner’s grounds relies entirely on Capalik (Ex. 1004) for this limitation; but Petitioner only points to activity occurring entirely outside Capalik’s “virtual machine monitor,” contrary to the express claim language. That dooms institution.

II. Background

A. The ’697 Patent (Ex. 1001)

The ’697 patent is entitled “System and method for identifying unauthorized activities on a computer system using a data structure model,” and issued on August 11, 2015. Stemming from a non-provisional application filed in 2011,

¹ Patent Owner has disclaimed independent claims 19 and 20, with no admission that those claims would have been invalidated under the Petition’s grounds. Ex. 2010.

which in turn claims priority to a provisional application filed June 24, 2010², the '697 patent is directed to “systems and methods for analyzing malicious activities on a computer system in order to better protect the computer system from future malicious activity.” '697 patent at 1:33-37.

The specification of the '697 patent explains that “networked computers are ... vulnerable to attacks, unwanted intrusions, and unauthorized access” by malicious computer programs such as malware. '697 patent at 1:41-46. The specification highlights that prior security systems suffered significant drawbacks because they focused on stopping known threats by matching “low-level IP data packets” against a preexisting database of “malware fingerprints.” *Id.* at 1:47-64. But relying on a library of “malware fingerprints” in connection with “low-level IP data packets” left computer systems vulnerable to previously unknown threats: “A critical threat to computer networks is the so-called zero-day attack that exploits security vulnerabilities previously unknown” *Id.* at 1:65-67. Likewise, because conventional solutions using an IDS (intrusion detection system) or IPS (intrusion prevention system) relied on “match[ing]” fingerprints, they were vulnerable to

² For purposes of this Preliminary Response, Patent Owner adopts this date as the critical date, but reserves the right to provide evidence in support of an earlier date of invention in this or in another proceeding.

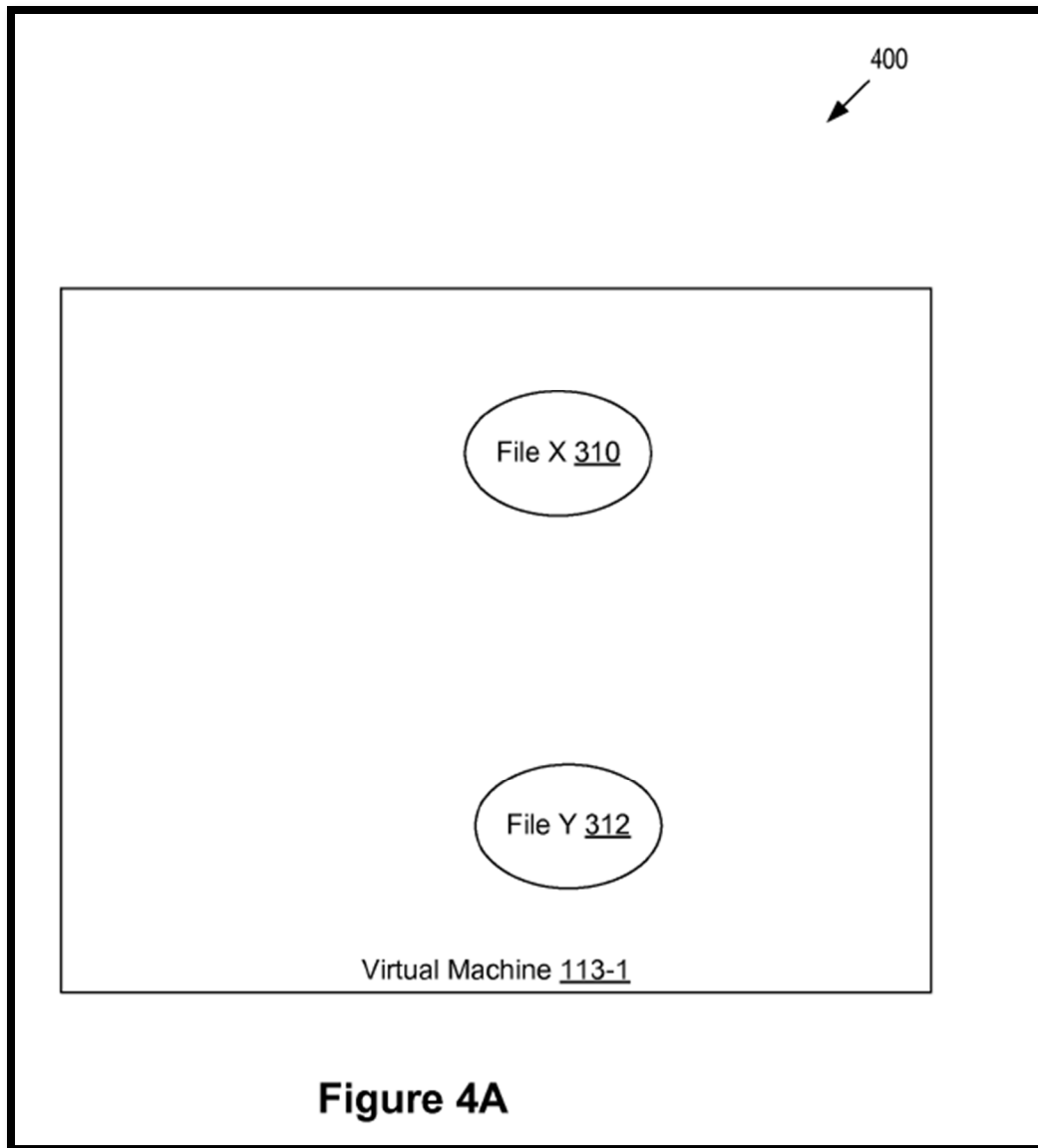
slight modifications in the packaging of malicious code, which could suffice to avoid signature-based detection while still exploiting the same or similar security vulnerabilities. *Id.* at 1:61-64.

The inventors of the '697 patent, therefore, saw “a great need for efficient methods of identifying fingerprint data for previously unknown types of malicious and/or unauthorized activities.” *Id.* at 2:9-13. However, “identifying the fingerprints of malicious activities in the middle of numerous other non-malicious processes is not a trivial task.” *Id.* at 2:7-9. The inventors solved that technological problem with specific techniques for “monitoring and identifying associations between activity sources and activity targets without using pre-determined fingerprints” and propagating that information to other computing devices. *Id.* at 3:1-6. The '697 patent describes a series of inventive steps to that end.

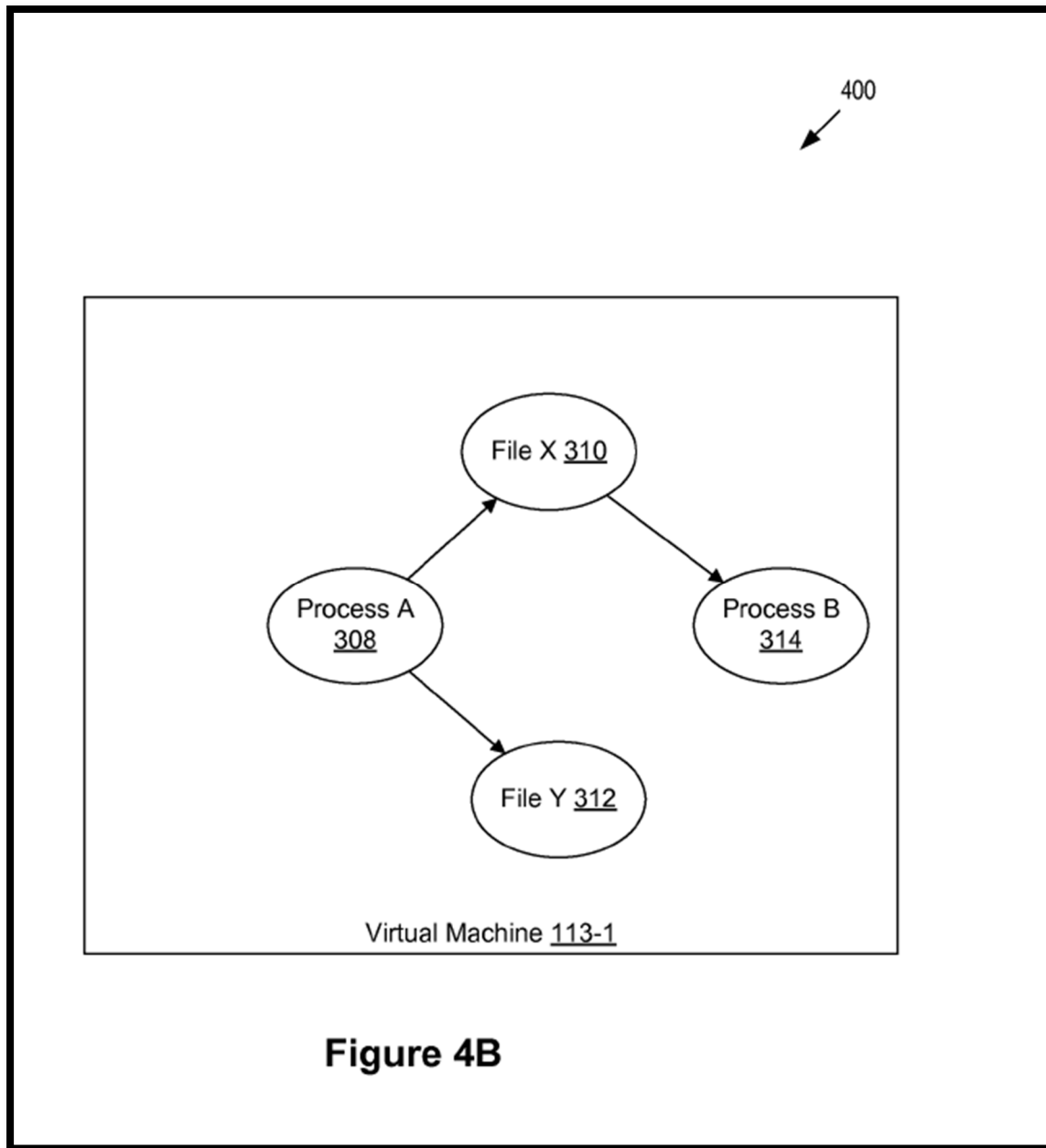
First, the invention monitors activity directly on a virtual machine. '697 patent at cl. 1. The invention thereby shifts the focus of malware detection and prevention away from monitoring incoming data at the gateway to a network to instead monitoring activity on a virtual machine. *Compare id.* at 1:58-60 (discussing prior art systems that “examine[d] the low-level IP data packets and compare[d] them to its library of fingerprints for a match”) *with id.* at 4:44-46 (“The decoy network device 106 is a decoy system that is monitored to collect fingerprint data of unauthorized activities.”), *id.* at 5:15-20 (“Although the methods described herein

are described with respect to monitoring the decoy network device 106, persons having ordinary skill in the art would recognize that similar methods may be used for monitoring one or more non-decoy systems, such as the protected network device 136.”).

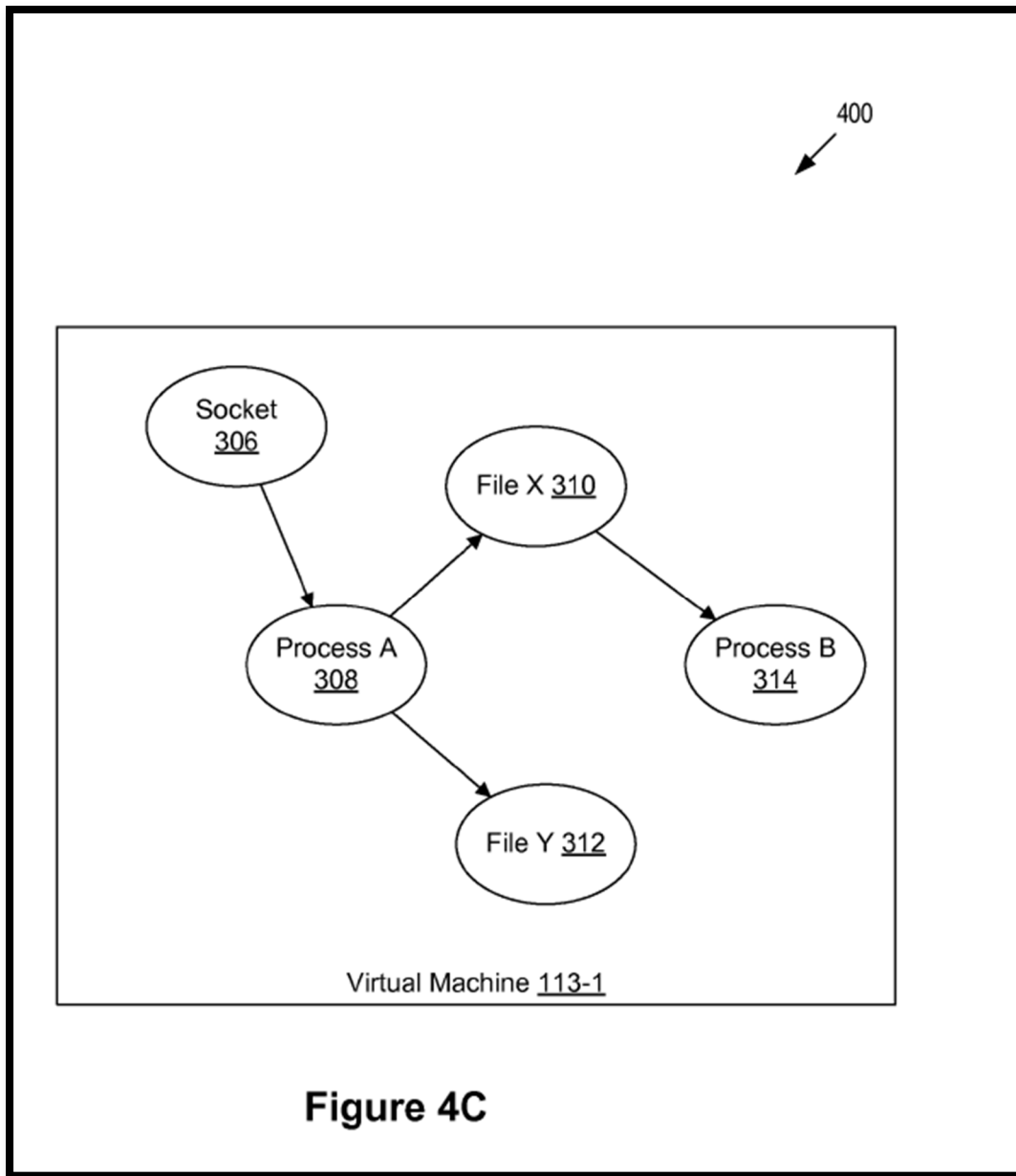
Figures 4A-4D (reproduced below) and the accompanying portions of the specification (*id.* at 11:39-65) are illustrative. As shown in Figure 4A, “unauthorized activities on the file X 310 and the file Y 312 in the virtual machine 113-1 are detected.” *Id.* at 11:42-44. For example, the '697 patent teaches that the claimed invention 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 11:44-48. Alternatively, the “unauthorized activity may be detected by monitoring data execution prevention (DEP) exceptions associated with the file X 310 and the file Y 312, generated by the one or more processors.” *Id.* at 11:48-51. In either case, detection of unauthorized activity begins by monitoring activity within virtual machine 113-1.



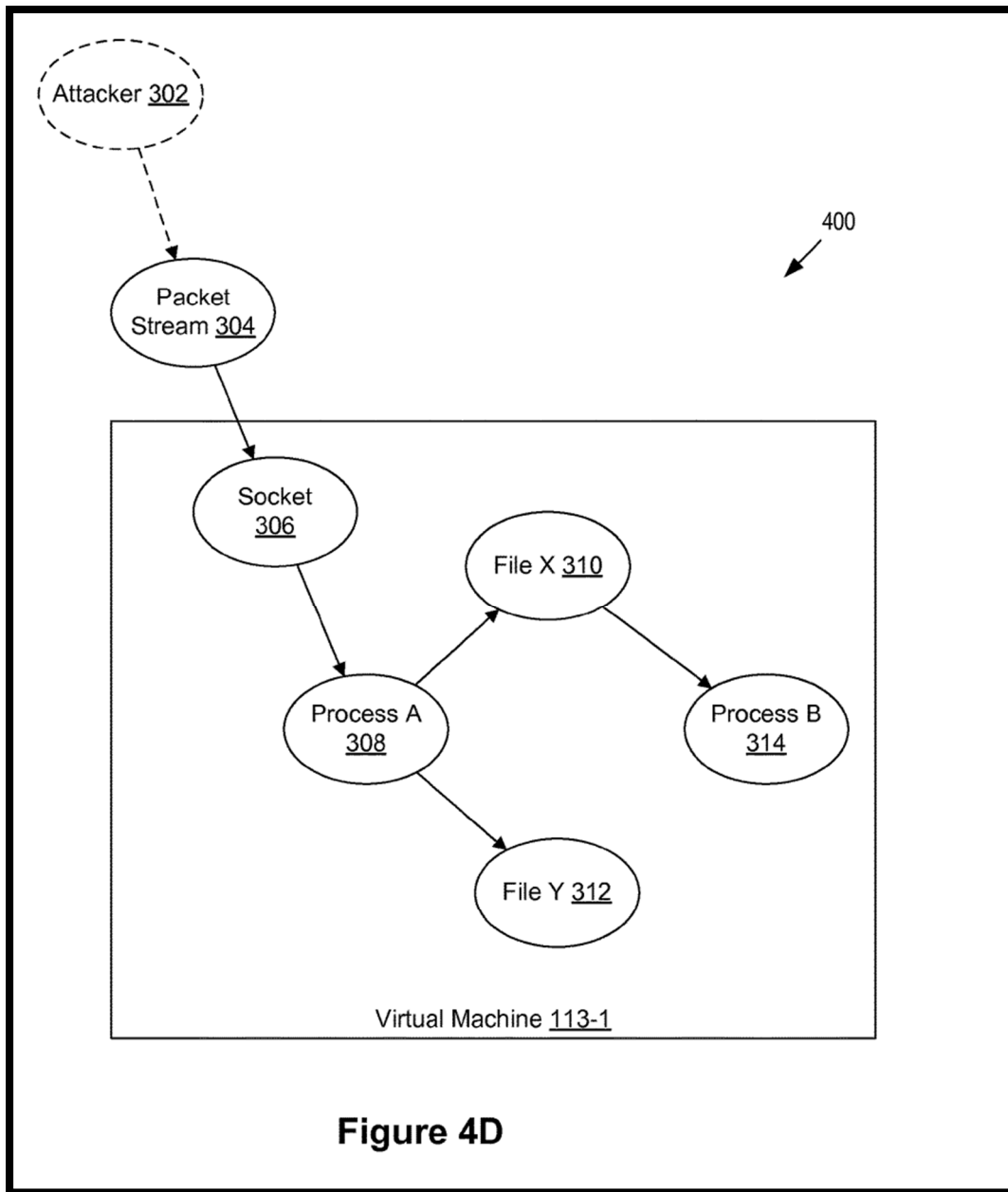
Next, as shown in Figure 4B, the '697 patent discloses techniques for tracking the transfer of malicious activity within the computer system. For example, “process A 308 [which] accesses both the file X 310 and the file Y 312” is “treated as affected by unauthorized activities.” '697 patent at 11:54-56. Likewise, because “file X 310 initiates the process B 314, [] the process B is treated as affected by unauthorized activities” as well. *Id.* at 11:56-58.

**Figure 4B**

Next, as shown in Figure 4C, the '697 patent discusses techniques for tracing the unauthorized activities back to the socket from which they originated. *Id.* at 11:59-61 (“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.”).



Lastly, the '697 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 11:61-65.



The '697 patent thus describes techniques for analyzing how malware manifested within a virtual machine, as opposed to relying on signature-based approaches that were restricted to identifying malware at the threshold to a network. As just one example, the '697 patent explains how the disclosed techniques “are able to follow a chain of unauthorized activity across multiple virtualized operating

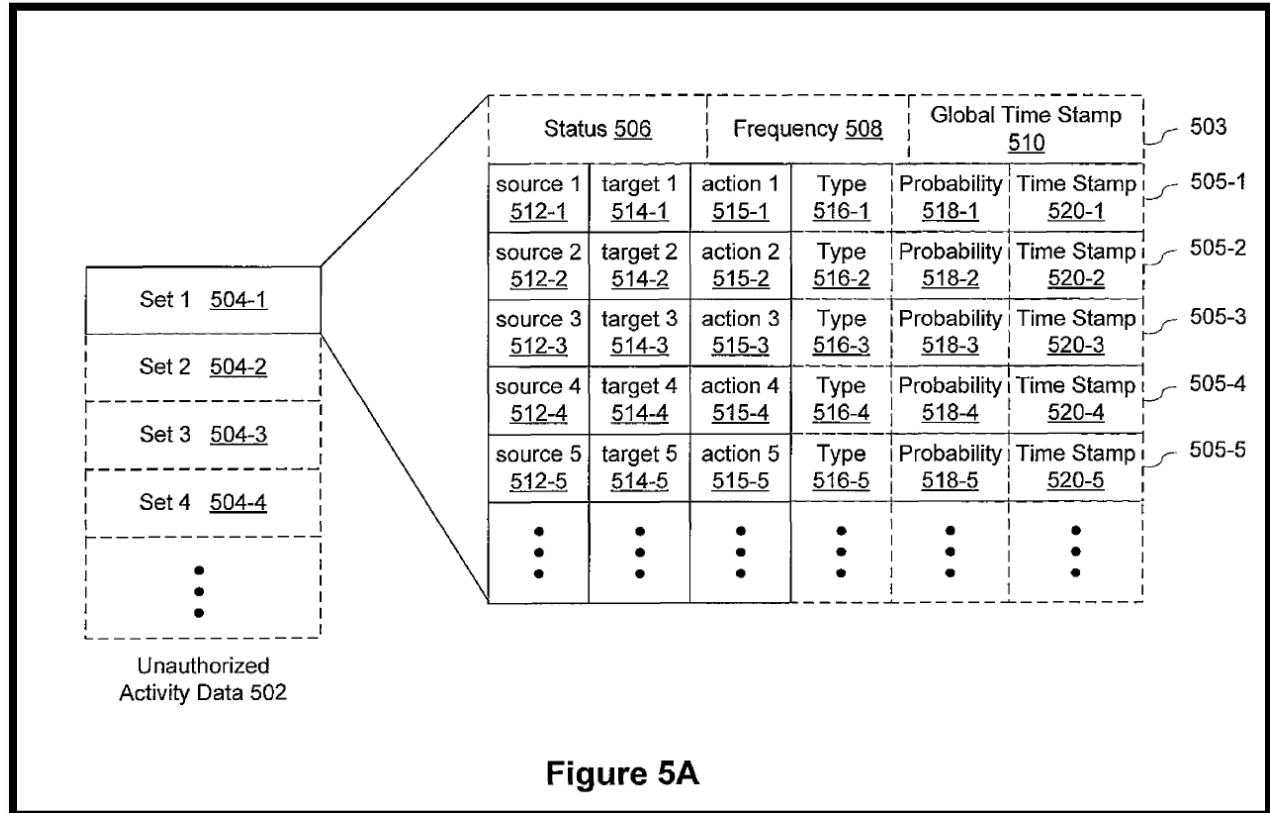
systems,” which is “important in identifying and thwarting advanced threats that involve multiple attackers spread across multiple systems and that evolve over long periods of time.” ’697 patent at 18:54-60. By focusing on the “chain of unauthorized activity” as opposed to the conventional, signature-based approach for monitoring “low-level IP data packets” whenever “a connection is attempted to a network port” (*id.* at 1:58-60), the techniques disclosed in the ’697 patent represented a paradigm-shifting approach to protecting computers.

Second, and particularly pertinent here, at least some steps of the method for identifying unauthorized activities described in the ’697 patent are “executed *at the virtual machine monitor*.” *Id.* at 14:39-41.³ To that end, the virtual machine monitor includes an “introspection module 120” that “performs introspection (e.g., monitoring) into the physical memory segments assigned to each of the virtualized decoy operating system 112.” *Id.* at 6:35-38.

The flowchart of FIGS. 6A-6C and accompanying description are illustrative. For example, “[t]he decoy network device monitors (602) activity on the virtual machine.” *Id.* at 14:42-43. “The decoy network device identifies (604) a plurality of activities being performed at the virtual machine. Each of the activities includes an activity source, an activity target, and an association between the activity source

³ All emphases added unless noted otherwise.

and the activity target.” *Id.* at 14:59-62. As the patent also explains, the “decoy network device stores (624) the activity sources, activity targets, and associations in the memory. For example, the decoy network device 106 stores unauthorized activity data 502 (e.g., as illustrated in FIG. 5A [reproduced below]) in memory 206.” *Id.* at 16:20-25. “Unauthorized activity data 502 includes unauthorized activity entries 505.” *Id.* at 12:21-22. “Each unauthorized activity entry 505 includes one or more of the following: an activity source 512 . . . [an] activity target 514, . . . [an] association type 516, which when identified, represents the type of association between the activity source 512 and the activity target 514.” *Id.* at 12:57-13:37.



Then, “decoy network device creates (636) a fingerprint indicative of the activity on the virtual machine from the stored activities.” ’697 patent at 17:33-35. For example, the fingerprint may be generated by the “introspection module,” which resides within the virtual machine monitor itself. *Id.* at 6:32-35 (“The hypervisor kernel 116 also includes an introspection module 120....”), 6:65-7:2 (“In this manner, the introspection module examines the contents of the memory segments of the virtualized decoy operating systems 112 in an instrumented context that generates and retrieves forensic data (or fingerprint).”). The fingerprint can be created by analyzing unauthorized activity data 502, and may include activity sources, files that are affected by unauthorized activities, processes that are affected by unauthorized activities, among other data. *See id.* at 17:37-67.

Lastly, “[t]he decoy network device transmits (644) the fingerprint to one or more protected network devices 136 to prevent future attacks that comprise the same or similar activities.” *Id.* at 18:5-8. With respect to the above description of FIGs. 6A-6C, “the one or more programs executed by the one or more processors to perform the method are executed at the virtual machine monitor.” *Id.* at 14:39-41.

B. Prosecution History

The ’697 patent issued from Application No. 13/163,590, filed on June 17, 2011. *See* ’697 patent, Cover. The patent claims benefit of Provisional Application 61/358,367, filed June 24, 2010. *Id.*

During prosecution, the Examiner issued a first non-final Office Action rejecting claims 1-20 under §§ 103 and 101. Ex. 1002 at 218. Patent Owner traversed these rejections without any substantive amendment to the claims. The Examiner issued a first notice of allowance in response to this traversal. *Id.* at 269.

Patent Owner subsequently entered a Request for Continued Examination in order to submit additional disclosure. *See id.* at 283. In response, the Examiner reopened prosecution and issued a new non-final Office Action, again rejecting claims 1-20 under § 103. Patent Owner again traversed the rejection without any amendments to the existing claims and added new dependent claims 21-23. *See id.* at 324-29. The Examiner issued a second notice of allowance in response. *See id.* at 335.

C. Person of Ordinary Skill in the Art (“POSITA”)

Solely for the purpose of this Preliminary Response, Patent Owner does not challenge Petitioner's proposed definition of a POSITA. Patent Owner reserves the right to provide an alternative definition of a POSITA if an IPR is instituted.

D. Claim Construction

The prior art relied on in the Petition fails to disclose the subject matter of the challenged claims under any reasonable construction of the claims' terms, including their plain meaning. Although Petitioner advances a proposed construction for one term (Pet. 3-7), denying institution does not require explicit construction of any term.

See, e.g., Realtime Data, LLC v. Iancu, 912 F.3d 1368, 1375 (Fed. Cir. 2019) (“The Board is required to construe ‘only those terms . . . that are in controversy, and only to the extent necessary to resolve the controversy.’” (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))). Patent Owner reserves the right to advocate for claim constructions in the event the Board institutes review in this case or based on arguments made in other proceedings, such as the parties’ concurrent district court litigation.

E. Overview of Petition

Though the Petition labels five grounds, Petitioner asserts eight grounds in its Petition. Pet. 10. As shown in the table below, Capalik is the primary reference in each ground:

Ground	Claim(s)	Reference(s)
1	1-7, 13, 16, and 19-20	Capalik
2	1-7, 13, 16, and 19-20	Capalik and Goel
3[A] ⁴	8-12, 15, 17, 21-23	Capalik and King
3[B]	8-12, 15, 17, 21-23	Capalik, Goel, and King
4[A]	14	Capalik and Pike
4[B]	14	Capalik, Goel, and Pike

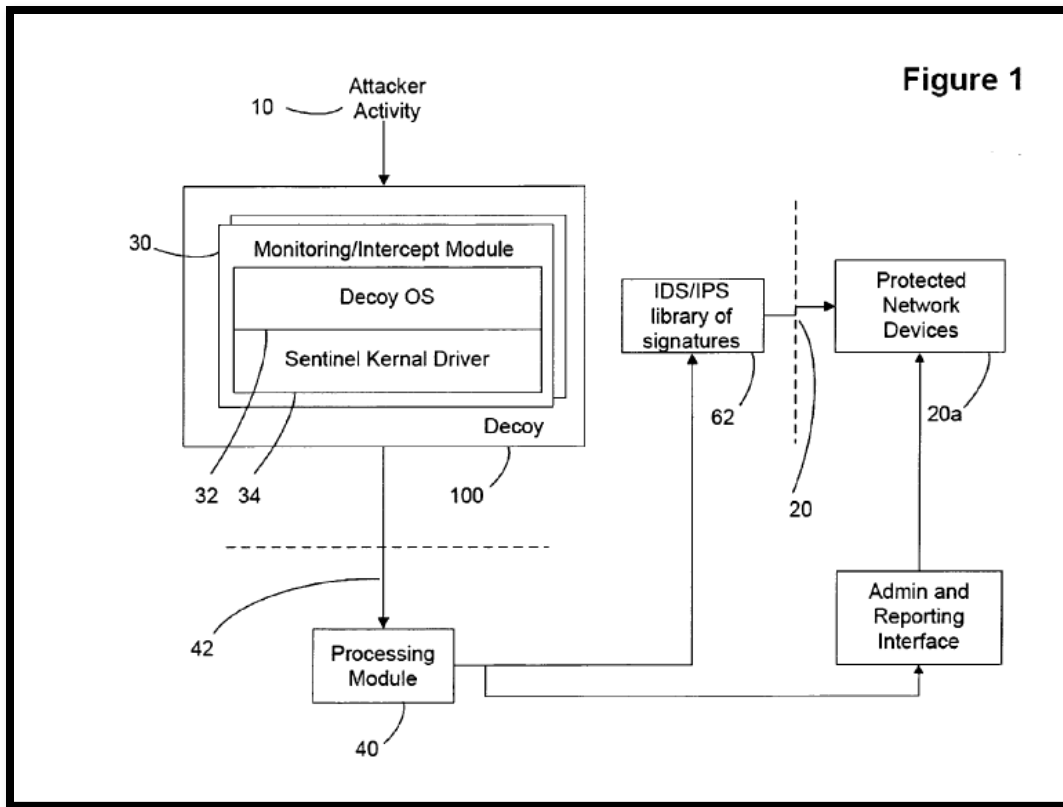
⁴ The [A] and [B] notations are added by Patent Owner to differentiate Petitioner’s Grounds relying on Capalik alone ([A] Grounds) and Capalik and Goel ([B] Grounds).

5[A]	18	Capalik and Farley
5[B]	18	Capalik, Goel, and Farley

F. Overview of Capalik (Ex. 1004)

Capalik, entitled “System and method for analyzing unauthorized intrusion into a computer network,” is directed to “methods and systems that adequately protect networks from new and undefined attacks and that allow for real-time updates to a network’s library of attack signatures.” Capalik at Cover, [0007].

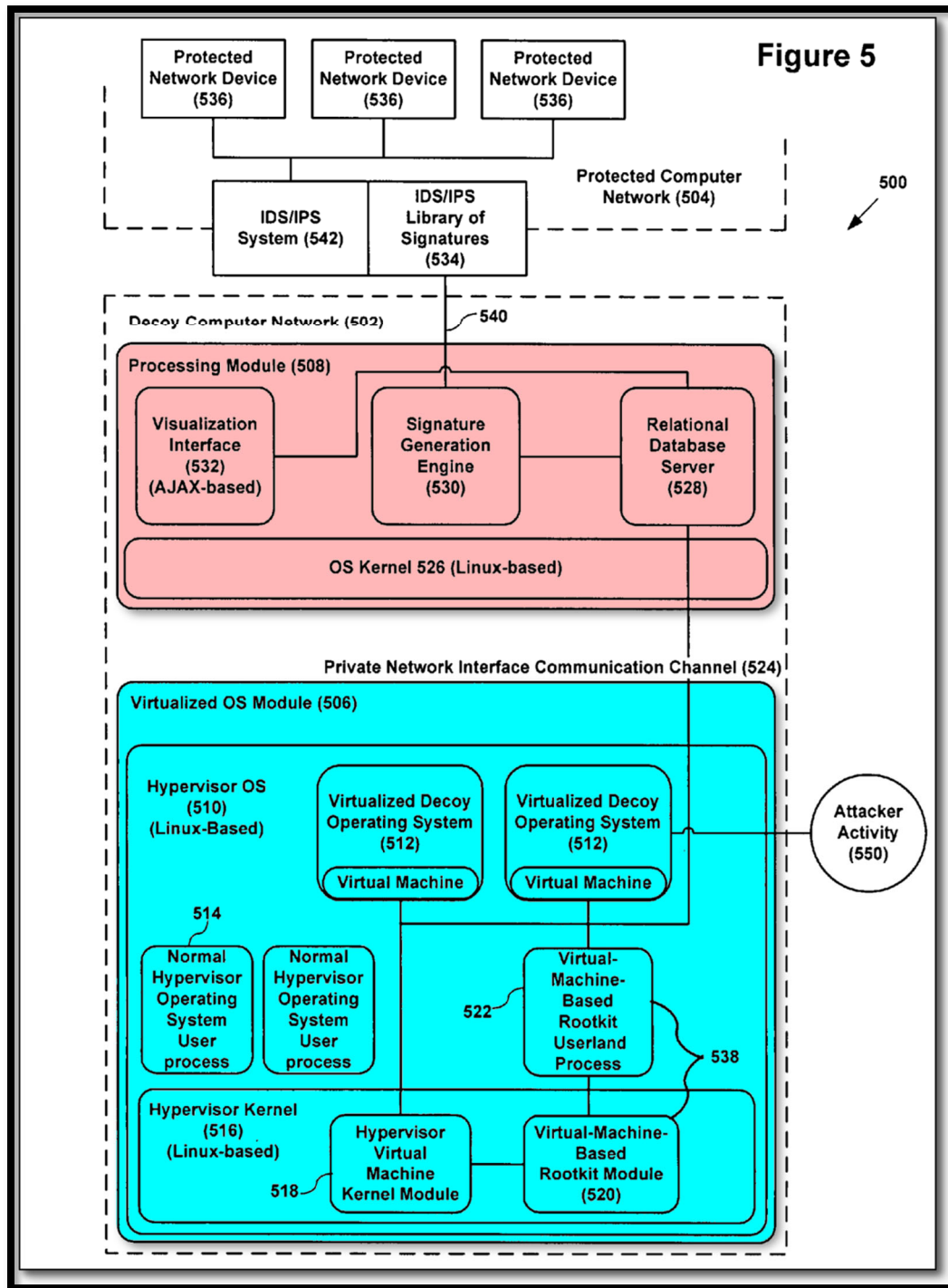
Capalik’s disclosure is explained with reference to Figure 1, reproduced below, which “illustrates attacker activity 10 directed at protected computer network 20.” Capalik [0022]. Capalik’s system “comprises a modular decoy network appliance” that includes: 1) “front-end fully functional operating system modules” and 2) “a *separate* processing back-end module.” *Id.* [0008].



Capalik's "front-end presents a standard fully functional operating system ... that returns a standard operating system fingerprint when it is scanned by tools that attackers typically use to identify vulnerable systems" such that an "attacker is thus lured into accessing the identified operating system and running custom or known exploits on that system." *Id.* [0009]. More specifically, "[w]hen 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." *Id.* [0027]. "The captured information, or attack-identifying information, is then sent for processing to a processing module 40 as illustrated in FIG. 1." *Id.* [0028]. The processing

module 40 then generates a signature associated with the attack. *See id.* [0033]. “[T]he signature may be sent from the back end processing module 40 to IDS/IPS 62.” *Id.* [0028].

Relevant to Petitioner's grounds, Capalik discusses the use of virtual machines in connection with its decoy computer, as illustrated in Figure 5 (reproduced below with annotations). Specifically, Capalik's decoy computer 502 contains two separate modules: 1) “a virtualized operating system module 506 for monitoring the decoy network 502” (blue), and 2) “a processing module 508 for obtaining, analyzing, and responding to exploits” (red). Capalik [0038].



As Capalik explains with respect to Figure 5, when an attacker “connects to a virtualized decoy operating system 512 through an open port, the attacker sees a fully-functional standard operating system fingerprint,” such that “[t]he instigator of

the attack 550 is [] lured into accessing the decoy 512.” *Id.* [0045]. By interacting with decoy 512, “the attacker provides decoy 512 with the data used to obtain control of the decoy 512.” *Id.* Specifically, Capalik’s introspection module 538 “examines the memory assigned to virtualized decoy operating systems 512 in order to acquire low-level data about the interaction between the decoy operating systems and [the] attack activity.” *Id.* [0048]. Examples of such “low-level data” include “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].

The “captured information ... is then sent from introspection module 538,” which resides within virtualized operating system module 506, to the other component of decoy computer 512: processing module 508. *Id.* [0048]. Capalik’s “*separate* processing back-end module” (*id.* [0008]) then performs *all* described analyses of the collected data. *See id.* [0055] (“[T]he *processing module 508* *generates* a report of the attack that includes attack-identifying information.”), *id.* [0059] (“If a determination is made that attack signature generation is warranted . . . *the processing module 508 may generate a signature.*”), *see also id.* at [0052] (“In use, the introspection module 538 captures . . . attack information. The attack information is . . . stored on the relational database server 528 [of the processing module 508] as forensic data for later analysis. The signature-generation engine 530

[of the processing module 508] then uses this forensic data to generate a signature of the attack.”).

Once generated, the signature is sent “from the back-end processing module 508 to the intrusion detection and/or prevention (IDS/IPS) signature library 534.” *Id.* [0054]. “The generated attack signature is thus applied to the IDS/IPS library of signatures 534, thereby updating the information contained in the signature library and preventing the attacker from accessing the protected network 504.” *Id.* [0059].

III. Argument

Claim 1 requires steps of its “computer implemented method” to be performed “at the virtual machine monitor.” *See* ’697 patent cl. 1.⁵ Specifically, claim 1 recites:

1. A computer implemented method of identifying unauthorized activities on a decoy computer system attached to a computer network, wherein the decoy system comprises:
 - one or more processors; and
 - memory storing:
 - a virtual machine; and
 - a virtual machine monitor supervising the virtual machine,the method comprising, ***at the virtual machine monitor:***
 - monitoring activity on the virtual machine;

⁵ Claim 1 stands in contrast to now-disclaimed claims 19 and 20, which are not expressly limited to steps “at the virtual machine monitor.”

identifying a plurality of activities being performed at the virtual machine, wherein each of the activities includes an activity source, an activity target, and an association between the activity source and the activity target;

storing in the memory the activity sources, activity targets, and associations;

creating, from the stored activities, a fingerprint indicative of the activity on the virtual machine; and

transmitting the fingerprint to prevent future attacks that comprise the same or similar activities as indicated by the fingerprint.

'697 patent at cl. 1. The structure of the claim plainly requires the virtual machine monitor to be involved in creating a fingerprint indicative of the activity on the virtual machine.

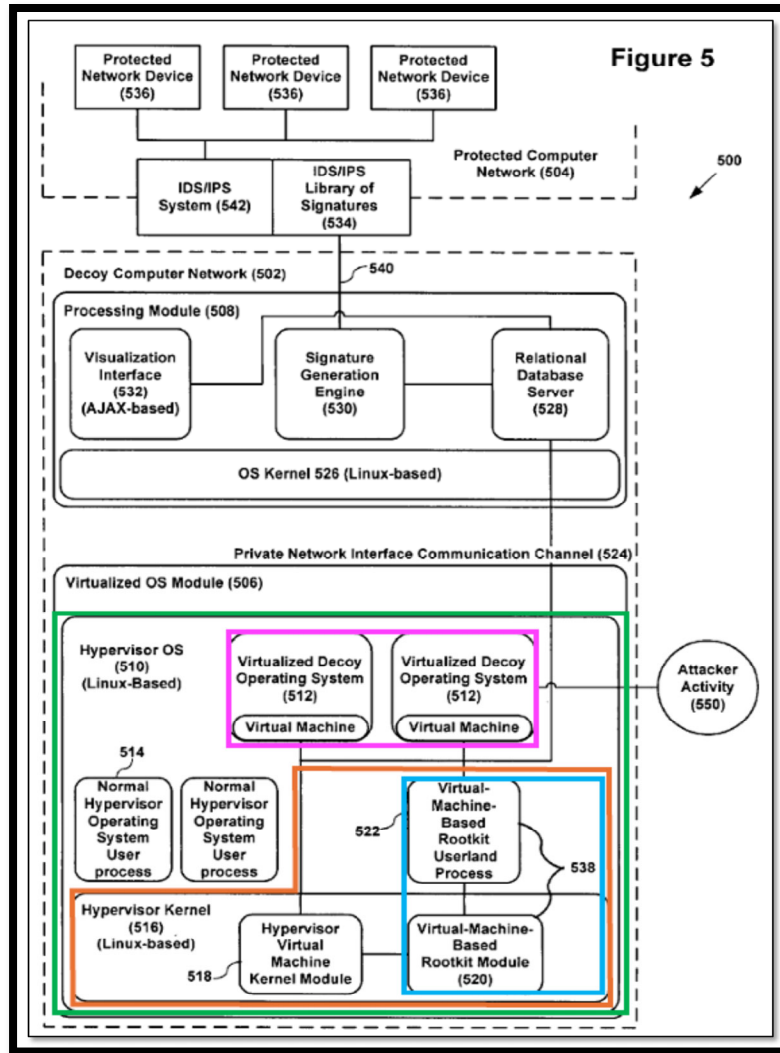
However, as demonstrated below, Petitioner's analysis of Capalik fails to show that ***any part of*** this creating a fingerprint step (i.e., limitation 1[e] in the Petition) is performed at the virtual machine monitor. In fact, Petitioner's arguments confirm that the opposite is true: under Petitioner's theory, any fingerprint generation in Capalik occurs entirely ***outside*** the virtual machine monitor. None of Petitioner's other references cures Capalik's deficiencies, and Petitioner does not argue otherwise. That is dispositive.

In its discussion of this limitation, Petitioner relies exclusively on Capalik's description of generating a signature. Pet. 30-32. Specifically, Petitioner contends that "*Capalik* discloses decoy system ***processing module 508 ... generates an attack***

signature.” *Id.* at 31. Because Capalik’s “processing module 508” is indisputably located entirely outside of Capalik’s virtual machine monitor, Capalik does not disclose this limitation.

Petitioner’s analysis of claim 1 centers on Figure 5 of Capalik (reproduced below with Petitioner’s annotations). *See* Pet. 21. Based on the color-coding used in the Petition, Petitioner appears to map the “virtual machine monitor” of claim 1 to Capalik’s hypervisor kernel 516 (annotated orange by Petitioner in the figure below). *Compare* Pet. 19 (annotated version of ’697 patent Figure 2) *with* Pet. 21 (annotated version of Capalik Figure 5). For purposes of this Preliminary Response, Patent Owner has assumed that Petitioner intended this mapping. However, Petitioner never specifies what, exactly, it contends to be the “virtual machine monitor” in Capalik. Instead, Petitioner asserts (without citation) that “*Capalik* teaches a virtual machine monitor (i.e. Hypervisor module) that monitors activity on multiple virtual machines.” Pet. 17. Presumably, Petitioner intended “Hypervisor module” to refer to Capalik’s hypervisor kernel; otherwise Petitioner’s arguments are inscrutable given that Capalik never uses the phrase “Hypervisor module.”⁶

⁶ While Petitioner’s cursory analysis is particularly problematic in view of claim 1’s requirement that certain aspects of the method be performed “at the virtual machine monitor,” Capalik’s shortcomings persist regardless of what mapping Petitioner



intended. That is so because Capalik's "separate processing back-end module" (Capalik [0008]) is indisputably located entirely outside of its "Virtualized OS Module," which Petitioner contends "includes the ... [virtual machine monitor]." Pet. 23. Thus, it follows that any steps performed at Capalik's processing module 508 are necessarily *not* performed "at the virtual machine monitor" as required by claim 1.

Capalik does not disclose *any* involvement of the “virtual machine monitor” in “creating...a fingerprint,” as required by claim 1. Instead, as Petitioner concedes, once data is collected by Capalik’s introspection module 538 (annotated **blue** by Petitioner in Figure 5 above), Capalik’s “processing module 508,” which is located entirely *outside* of the virtual machine monitor, performs all subsequent analysis. Capalik [0050]. According to Petitioner’s theory, processing module 508 uses data stored in “relational database server 528” (which is also outside of the hypervisor or alleged “virtual machine monitor”) to generate the signature. *See* Pet. 30-31 (“decoy system comprising Relational Database Server 528 [] stores ‘attack information’ as forensic data *Capalik* uses a signature-generation engine to create a ‘signature of the attack’ from the stored forensic data in a relational database.”) (quoting Capalik [0052]). Capalik confirms that the processing module, not the virtual machine monitor (i.e., the “hypervisor kernel” in Petitioner’s mapping), creates fingerprints or signatures: “*processing module 508* generates a report of the attack that includes attack-identifying information.” Capalik [0055]; *see also id.* [0059] (“If a determination is made that attack generation is warranted . . . the processing module 508 may generate a signature.”). Thus, the “virtual machine monitor” plays no part in “creating...a fingerprint” under Petitioner’s theory, contrary to the claim language.

Petitioner alleges that “like the ’697 Patent, *Capalik* uses a signature-generation engine to create a ‘signature of the attack’ from the stored forensic data in a relational database.” Pet. 30-31 (citing ’697 patent at 5:63-67). But Petitioner must compare the prior art *to the ’697 patent’s claims*, not the specification. *See, e.g., Laitram Corp. v. NEC Corp.*, 163 F.3d 1342, 1347 (Fed. Cir. 1998). Claim 1 specifically requires that the “virtual machine monitor” be involved in the claimed fingerprint creation, yet Petitioner identifies no such discussion in *Capalik*.

Additionally, Petitioner neither advocates nor establishes that a POSITA would have been motivated to *modify*⁷ *Capalik* to move any part of the signature-generation functionality of processing module 508 within the hypervisor OS. Instead, Petitioner merely asserts that *Capalik* “teaches” this limitation (Pet. 30), without any consideration of what component(s) of *Capalik*’s system are involved in generating the signature. But *Capalik* does not.

⁷ Although the Petition applies *Goel* to other aspects of claim 1 (*see* Pet. 43), Petitioner’s analysis of this limitation does not allege that *Goel* cures any deficiency of *Capalik* with respect to the “creating” requirement, nor is there any argument in the Petition that *Goel* creates a fingerprint as recited. (*Goel* does not do so.) Thus, even a hypothetical *Capalik-Goel* combination would not disclose or render obvious this subject matter of claim 1.

Accordingly, because Capalik does not disclose or render obvious the involvement of a virtual machine monitor in “creating, from the stored activities, a fingerprint indicative of the activity on the virtual machine” as required by claim 1, Petitioner has not shown a reasonable likelihood of prevailing against claim 1. This is not a case that merits institution for further development of the record. The claims say one thing: “at the virtual machine monitor ... creating a fingerprint.” The Petition describes another thing—creating a fingerprint without any involvement of the virtual machine monitor whatsoever. Fatally, Petitioner’s grounds all rely only on Capalik for this limitation, and thus Petitioner’s analysis cannot be supplemented without impermissibly deviating from the Petition. Thus, the Board should deny institution.

With respect to the remainder of the claims challenged in grounds 1 and 2 (i.e., claims 2-7, 13, and 16), each such claim depends from claim 1 and thus is not rendered obvious by Capalik alone or in combination with Goel for at least the same reasons as claim 1.

Grounds 3, 4, and 5 of the Petition address only the additional limitations of claims 8-12, 14, 15, 17, 18, and 21-23, each of which depends from claim 1. Pet. 50-68 (Ground 3), 68-71 (Ground 4), 71-75 (Ground 5). For at least the same reasons Petitioner fails to demonstrate a reasonable likelihood of prevailing against

claim 1, Petitioner fails to demonstrate a reasonable likelihood of prevailing against claims 8-12, 14, 15, 17, 18, and 21-23.

IV. Conclusion

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

Respectfully submitted,

Dated: February 13, 2025

By: / S. Giri Pathmanaban /

S. Giri Pathmanaban (Reg. No. 75,986)
giri.pathmanaban@lw.com
Latham & Watkins LLP
300 Colorado Street, Suite 2400
Austin, TX 78701-3295
Telephone: 737.910.7300
Fax: 737.910.7301

*Counsel for Patent Owner
GoSecure, Inc.*

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

I hereby certify that this Patent Owner's Preliminary Response complies with the word count limitation of 37 C.F.R. § 42.24(b)(1) because the Patent Owner's Preliminary Response contains 3,987 words using Microsoft Word's counting feature, excluding the cover page, signature block, and the parts of the Patent Owner's Preliminary Response exempted by 37 C.F.R. § 42.24(b)(1).

By: / S. Giri Pathmanaban /

S. Giri Pathmanaban (Reg. No. 75,986)
giri.pathmanaban@lw.com
Latham & Watkins LLP
300 Colorado Street, Suite 2400
Austin, TX 78701-3295
Telephone: 737.910.7300
Fax: 737.910.7301

*Counsel for Patent Owner
GoSecure, Inc.*

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6(e), I certify that on this 13th day of February, 2025, a true and correct copy of the foregoing **Patent Owner's Preliminary Response and Exhibit 2010** 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. Giri Pathmanaban /

S. Giri Pathmanaban (Reg. No. 75,986)
giri.pathmanaban@lw.com
Latham & Watkins LLP
300 Colorado Street, Suite 2400
Austin, TX 78701-3295
Telephone: 737.910.7300
Fax: 737.910.7301

*Counsel for Patent Owner
GoSecure, Inc.*