

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

WIZ, INC.,
Petitioner,

v.

ORCA SECURITY LTD,
Patent Owner.

Case IPR2025-00441
Patent No. 11,868,798

DECLARATION OF DR. ANGELOS STAVROU

TABLE OF CONTENTS

I.	ENGAGEMENT.....	1
II.	QUALIFICATIONS.....	1
III.	COMPENSATION.....	3
IV.	INFORMATION CONSIDERED.....	3
V.	LEGAL PRINCIPLES.....	4
VI.	THE RELEVANT TIMEFRAME FOR ANALYSIS.....	7
VII.	PERSON OF ORDINARY SKILL IN THE ART.....	8
VIII.	STATE OF THE ART.....	9
	A. Virtualization.....	9
	B. Snapshots.....	22
	C. Cloud Computing.....	31
	D. Cyber Security.....	39
	1. Types of Security Risks.....	40
	2. Detecting Security Risks.....	49
	3. Responding to Security Risks.....	66
IX.	OVERVIEW OF THE '798 PATENT.....	74
	A. Challenged Claims.....	77
	B. Prosecution History of the '798 Patent.....	79
X.	CLAIM CONSTRUCTION.....	79
	A. “[Analyzing/Analyze] the Snapshot”.....	80
XI.	OVERVIEW OF THE PRIOR ART.....	81

A.	Veselov (U.S. Patent No. 11,216,563, EX1007).....	81
B.	Basavapatna (U.S. Pub. No. 2013/0191919, EX1008)	87
C.	Kapoor (US Patent No. 10,498,845, EX1111).....	91
D.	Czarny (U.S. Patent No. 9,749,349; EX1084)	93
E.	Chari (U.S. Patent Pub. 2015/0033221; EX1056)	97
F.	Roth (U.S. Patent No. 9,524,389; EX1116).....	99
XII.	GROUND 1: CLAIMS 1-5, 7-15, 17-22, 24-32, AND 34 WERE OBVIOUS OVER VESELOV AND BASAVAPATNA.....	100
A.	Reasons to Combine Veselov and Basavapatna.....	101
B.	Independent Claims 1 and 18.....	109
1.	Preambles	109
2.	Element 18.i	113
3.	Elements 1.1 and 18.1	115
4.	Elements 1.2 and 18.2	121
5.	Elements 1.3 and 18.3.....	123
6.	Elements 1.4 and 18.4.....	138
C.	Dependent Claims	144
1.	Claims 2 and 19.....	144
2.	Claims 3 and 20.....	149
3.	Claims 4 and 21.....	160
4.	Claims 5 and 22.....	170
5.	Claims 7 and 24.....	175
6.	Claims 8 and 25.....	185
7.	Claims 9 and 26.....	187
8.	Claims 10 and 27.....	188

9.	Claims 11 and 28.....	189
10.	Claims 12 and 29.....	190
11.	Claim 13 and 30	192
12.	Claims 14 and 31.....	195
13.	Claims 15 and 32.....	197
14.	Claims 17 and 34.....	200
XIII.	GROUND 2: CLAIMS 6 AND 23 WERE OBVIOUS OVER VESELOV, BASAVAPATNA, AND KAPOOR.....	202
A.	Reasons to Combine Veselov, Basavapatna, and Kapoor.	203
B.	Claims 6 and 23.....	209
XIV.	GROUND 3: CLAIMS 11 AND 28 WERE OBVIOUS OVER VESELOV, BASAVAPATNA, AND ROTH	213
A.	Reasons to Combine Veselov, Basavapatna, and Roth.....	214
B.	Claims 11 and 28.....	218
XV.	GROUND 4: CLAIMS 14 AND 31 WERE OBVIOUS OVER VESELOV, BASAVAPATNA, AND CZARNY	222
A.	Reasons to Combine Veselov, Basavapatna, and Czarny	222
B.	Claims 14 and 31.....	228
XVI.	GROUND 5: CLAIM 13, 16, 30, AND 33 WERE OBVIOUS OVER VESELOV, BASAVAPATNA, AND CHARI.....	230
A.	Reasons to Combine Veselov, Basavapatna, and Chari.....	230
B.	Claims 13 and 30.....	234
C.	Claims 16 and 33.....	237
XVII.	CONCLUDING STATEMENTS.....	239
XVIII.	APPENDIX A – MATERIALS CITED	240

I, Angelos Stavrou, declare as follows:

I. ENGAGEMENT

1. I have been retained by counsel for Wiz, Inc. as an expert witness in the above-captioned proceeding. I have been asked to provide my opinion about the state of the art of the technology described in U.S. Patent No. 11,868,798 (the “’798 patent”) and on the patentability of claims 1-34 of this patent. The following is my written testimony on these topics.

II. QUALIFICATIONS

2. I received my M.Sc. in Electrical Engineering, M.Phil., and Ph.D. (with distinction) in Computer Science all from Columbia University. I also hold an M.Sc. in theoretical Computer Science from the University of Athens and a B.Sc. in Physics with distinction from the University of Patras, Greece.

3. I am a Virginia Tech Innovation Campus founding Professor, and the Entrepreneurship activities lead. I am also a member of the Bradley Department of Electrical & Computer Engineering at Virginia Tech. From 2017 to 2020, I was a Professor in the Computer Science Department at George Mason University (“GMU”), teaching courses including Operating Systems Security and Cyber Security Laboratory. From 2012 to 2017, I was an Associate Professor in GMU’s Computer Science Department, teaching courses including Operating Systems Security, Enterprise Security Practices, and Enterprise Security Technology. From

2014 to 2017, I was an Academic Director in GMU's School of Management for the M.S. in Management of Secure Information Systems Program. From 2013 to 2015, I was an Academic Director in GMU's Computer Science Department for the M.S. in Information Security and Assurance Program. From 2011 to 2020, I was an Associate Researcher in the Computer Security Division at the National Institute of Standards and Technology ("NIST"). I am a co-author of numerous publications involving virtualization and cyber security, among other topics. EX1003, 3-19.

4. I am also the founder of Quokka, Kryptowire Labs, Aether Argus, and Impedyme Inc. I have served as a principal investigator on research awards from NSF, DARPA, IARPA, DHS, AFOSR, ARO, ARL, and ONR. I have written more than 140 peer-reviewed conference and journal articles. Furthermore, I am an inventor of nineteen issued patents and several pending patent applications. I am an Associate Editor of IEEE Transactions on Computers, IEEE Security & Privacy, and IEEE Internet Computing magazine and a co-chair of the IEEE Blockchain initiative. I am a senior member of the ACM, USENIX, and IEEE. My current research interests include security and reliability for distributed systems, security principles for virtualization, and anonymity, focusing on building and deploying large-scale systems.

5. I received the GMU Department of Computer Science Outstanding Research Award in 2010, 2016, and 2018. Also, I was awarded the 2012 George Mason Emerging Researcher, Scholar, Creator Award, a university-wide award. In 2013, I received the IEEE Reliability Society Engineer of the Year award. In 2024, I received the IEEE Reliability Society Lifetime Award. My team at Kryptowire was awarded the DHS Cyber Security Division's "Significant Government Impact Award" in 2017 and the "Bang for the Buck Award" in 2019. Currently, I am the primary Principal Investigator (PI) in two DARPA awards focusing on Cyber Security for cloud and wireless systems, namely the DARPA OPS-5G and CASTLE efforts.

III. COMPENSATION

6. I am being compensated for my time at my standard consulting rate. My compensation is not contingent upon the results of my study and analysis, the substance of my opinions, or the specifics of my testimony. I have no financial interest in the outcome of this matter or in any litigation involving the '798 patent.

IV. INFORMATION CONSIDERED

7. My opinions are based on my years of education, research and experience, as well as my investigation and study of relevant materials. In preparing this declaration, I have reviewed the '798 patent and the '798 patent's file history

(EX1004). I have also reviewed the materials cited in this declaration, a list of which is provided in Appendix A.

V. LEGAL PRINCIPLES

8. I understand that a claim is not patentable under 35 U.S.C. §102 as anticipated if the claim as a whole is described, either expressly or inherently, in a single prior art reference. I understand that this also requires that the prior art reference disclose that each claim element is arranged as recited in the claims.

9. I have been advised that a claimed invention is not patentable under 35 U.S.C. §103 if it is obvious. A patent claim is unpatentable if the claimed invention would have been obvious to a person of ordinary skill in the field at the time the claimed invention was made. This means that even if all of the requirements of the claim cannot be found in a single prior art reference that would anticipate the claim, a person of ordinary skill in the relevant field who knew about all this prior art would have come up with the claimed invention.

10. I have further been advised that the ultimate conclusion of whether a claim is obvious should be based upon several factual determinations. That is, a determination of obviousness requires inquiries into: (1) the level of ordinary skill in the field; (2) the scope and content of the prior art; (3) what difference, if any, existed between the claimed invention and the prior art; and (4) any secondary evidence bearing on obviousness.

11. I have been advised that, in determining the level of ordinary skill in the field that someone would have had at the time the claimed invention was made, I should consider: (1) the levels of education and experience of persons working in the field; (2) the types of problems encountered in the field; and (3) the sophistication of the technology.

12. I have also been advised that, in determining the scope and content of the prior art, in order to be considered as prior art, a reference must be reasonably related to the claimed invention of the patent. A reference is reasonably related if it is in the same field as the claimed invention or is from another field to which a person of ordinary skill in the field would look to solve a known problem.

13. I have been advised that a patent claim composed of several elements is not proved obvious merely by demonstrating that each of its elements was independently known in the prior art. In evaluating whether such a claim would have been obvious, I may consider whether there are reasons that would have prompted a person of ordinary skill in the field to combine the elements or concepts from the prior art in the same way as in the claimed invention.

14. I have been further advised that there is no single way to define the line between true inventiveness on the one hand (which is patentable) and the application of common sense and ordinary skill to solve a problem on the other hand (which is

not patentable). For example, market forces or other design incentives may be what produced a change, rather than true inventiveness. I may consider whether the change was merely the predictable result of using prior art elements according to their known functions, or whether it was the result of true inventiveness. I may also consider whether there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent. I may consider whether the innovation applies a known technique that had been used to improve a similar device or method in a similar way. I may also consider whether the claimed invention would have been obvious to try, meaning that the claimed innovation was one of a relatively small number of possible approaches to the problem with a reasonable expectation of success by those skilled in the art.

15. I have also been advised, however, that I must be careful not to determine obviousness using the benefit of hindsight; many true inventions might seem obvious after the fact. I should put myself in the position of a person of ordinary skill in the field at the time the claimed invention was made and I should not consider what is known today or what is learned from the teaching of the patent.

16. Finally, I have been advised that any obviousness rationale for modifying or combining prior art must include a showing that a person of ordinary skill would have had a reasonable expectation of success.

17. With regard to secondary considerations of nonobviousness, I have been advised that any objective evidence may be considered as an indication that the claimed invention would not have been obvious at the time the claimed invention was made. I understand that the purpose of secondary considerations is to prevent a hindsight analysis of the obviousness of the claims.

18. I have been advised that there are several factors that may be considered as a secondary consideration. These factors include the commercial success of the invention, industry praise for the invention, skepticism of the invention, licensing of the invention, copying of the invention, any long-felt need that the invention solved, failure of others, and unexpected results of the invention.

19. I have been further advised that in order for evidence of secondary considerations to be significant, there must be a sufficient nexus between the claimed invention and the evidence of secondary considerations. I understand that this nexus serves to provide a link between the merits of the claimed invention and the evidence of secondary considerations provided.

VI. THE RELEVANT TIMEFRAME FOR ANALYSIS

20. I understand that the '798 patent issued from U.S. Application No. 17/821,345 ("the '345 Application"), filed August 22, 2022. EX1001, Face. The '345 application claims priority to Provisional Application No. 62/797,718, filed

January 28, 2019. I have been asked to treat January 28, 2019 as the effective filing date for purposes of my analysis.

VII. PERSON OF ORDINARY SKILL IN THE ART

21. As I noted earlier in §V, I have been advised that, in determining the level of ordinary skill in the field that someone would have had at the time the claimed invention was made, I should consider: (1) the levels of education and experience of persons working in the field; (2) the types of problems encountered in the field; and (3) the sophistication of the technology.

22. In my opinion, a person of ordinary skill in the art (“POSA”) at the relevant time would have had (1) at least a bachelor’s degree in computer science, computer engineering, electrical engineering, or a related degree, and (2) would also have 2-3 years of professional experience with cyber security analysis and virtualization. Such experience includes, for example, malware analysis, security analysis of cloud computing systems, and security analysis of virtual machines. I satisfied these requirements before January 28, 2019. *See supra*, §II. Additional experience can compensate for less education and vice versa. The State-of-the-Art section below summarizes some basic background knowledge of the type that persons of ordinary skill in the art would have had.

VIII. STATE OF THE ART

23. In this section, I provide a brief overview of the background art as understood by a POSA as of January 28, 2019. The material that I cite below confirms my own recollection and understanding, based on my experience in the field, of the state of the art before January 28, 2019. First, I provide a general overview of virtualization technology and cloud computing. I then provide a high-level discussion of cyber security, including several types of threats and vulnerabilities that were commonly exploited, how those threats were detected, and common responses to detected threats. I also explain how these cyber security threat-detection, classification, and mitigation approaches were applied in virtualized cloud computing assets available as of January 28, 2019.

A. Virtualization

24. Virtualization refers to the decades-old process of emulating physical computers. EX1009 (Wolf), xxiii (“In running as a virtual machine, a computer’s hardware is emulated and presented to an operating system as if the hardware truly

existed.”);¹ EX1010 (Waldspurger), 2 (“Virtual machines have been used for decades.”).² Indeed, the technology was described as well-known even by the 1970s. EX1011 (Goldberg), 35 (“[M]uch of the software for the simulated machine executes directly on the hardware without software interpretation. Systems of this kind are called virtual machine systems, the simulated machines are called virtual

¹ EX1009 includes excerpts from “Virtualization: From the Desktop to the Enterprise,” a book by C. Wolf and E. Halter, published in 2005 by APRESS. This book was obtained from <https://link.springer.com/book/10.1007/978-1-4302-0027-7>. To avoid overburdening the record, EX1009 includes only the front matter, chapter 1, chapter 6, chapter 7, and appendix A, since the book is quite large and I only cite to chapters 1, 6, and 7 and the appendix.

² EX1010 is a copy of “Memory Resource Management in VMware ESX Server,” by C. Waldspurger, published by USENIX in 2002 as a part of the 5th Symposium on Operating Systems Design and Implementation. This copy was obtained from <https://www.usenix.org/legacy/event/osdi02/tech/waldspurger/waldspurger.pdf>.

machines (VMs), and the simulator software is called the virtual machine monitor (VMM) ... IBM[] improved virtual machine support for System/370.”).³

25. An emulated physical computer was called a “virtual machine” or “VM.” EX1009 (Wolf), xxiii (“[W]orkstations and servers no longer need dedicated physical hardware such as a CPU or motherboard in order to run as independent entities. Instead, they can run inside a virtual machine (VM).”). A VM acted like a physical machine, in that it typically ran a standard operating system, such as Linux or Windows, and could run standard software applications, such as web servers. EX1010 (Waldspurger), 2 (Virtualization software such as VMware’s ESX Server “is in production use on servers running multiple instances of unmodified operating systems such as Microsoft Windows 2000 Advanced Server and Red Hat Linux 7.2.”). Desktop, server, cloud, and datacenter providers routinely used many

³ EX1011 is a copy of “Survey of Virtual Machine Research,” by R. Goldberg, published by the IEEE Computer Society in June 1974 in *Computer Magazine*, Vol. 7, No. 6. This copy was obtained from <https://ieeexplore.ieee.org/document/6323581>.

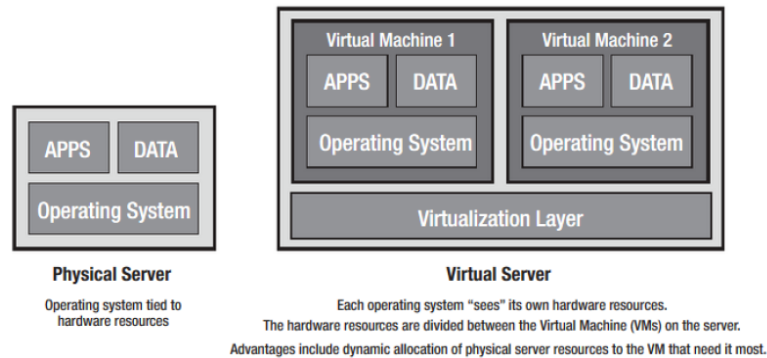
different virtualization solutions and products, spanning both closed and open-source, available for desktop, server, and cloud applications.

26. The term “virtualization” most commonly referred to a hypervisor-based process. A hypervisor, sometimes called a Virtual Machine Monitor (VMM), was a software program that enabled the emulation of the hardware of a physical machine, e.g. by “instantiating” virtual machines. The hardware running the hypervisor was called the “host” (and the operating system was called the host operating system), whereas emulated virtual machines running inside them were referred to as “guests” (and their operating systems were called guest operating systems). EX1009 (Wolf), 5. A single VM could be “instantiated,” that is, copied and spun up, into one or more running guest “instances.”

27. A guest VM’s operating system and software applications were typically isolated from the other VMs on the system and were therefore unaware that they were actually executing in a virtual, rather than in a physical, machine. EX1010 (Waldspurger), 2 (“Each virtual machine (VM) is given the illusion of being a dedicated physical machine that is fully protected and isolated from other virtual machines.”); EX1009 (Wolf), 505 (“By setting up virtual partitions, existing applications and users ‘see’ each virtual machine as an independent physical server although they share common CPU, disk, memory, and network resources”). This

“enables application isolation since malicious or greedy applications cannot impact other applications co-located on the same physical server.” EX1013 (Wood), 229.⁴

28. The figure below compares a traditional physical server to a virtual server:



EX1009 (Wolf), 505, Fig. A-5. As shown on the left, a traditional physical server included a single operating system that was tied directly to the server’s hardware. In contrast, a virtual server (shown on the right) included one or more guest VMs (such

⁴ EX1013 is a copy of “Black-box and Gray-box Strategies for Virtual Machine Migration,” by T. Wood et al., published by USENIX in 2007 as a part of NSDI ’07: 4th USENIX Symposium on Networked Systems Design & Implementation. This copy was obtained from <https://www.usenix.org/conference/nsdi-07/black-box-and-gray-box-strategies-virtual-machine-migration>.

as Virtual Machine 1 or Virtual Machine 2), each with its own operating system. But rather than interact with the physical machine's hardware, these operating systems interacted with a virtualization layer (such as a hypervisor or VMM) that gave each the impression it was interacting with its own physical machine. Thus, "virtualization technology ... allow[ed] many physical servers to be hosted and isolated from each other on fewer physical machines." EX1009 (Wolf), 505. Hosted VMs were also typically assigned IP and MAC addresses to allow them to send and receive network data. EX1009 (Wolf), xxiv ("[C]lients won't connect to a physical computer but instead connect to a logical virtual server running on top of one or more physical computers."), 22 (guest host VMs have their own IP and MAC addresses that are used to send and receive traffic through a virtual network interface mapped to a physical interface); EX1012 (Doctor), 9:9-25 (VM's virtual network interface is assigned a network address (e.g., IP address) through which it may receive data); EX1048 (Price), ¶31 ("[I]nformation can be obtained using VMM access engine 280 identifying the status and location (e.g., IP or MAC address) of each enumerated virtual machine in system 240....").

29. The host typically allocated some portion of each of its resources (e.g., CPU, memory, I/O, storage) for exclusive use by each virtual machine. The host could change this allocation in response to many possible technical and non-

technical requirements that changed over time. “For example, a sudden increase in number of customers may place a greater strain on [an existing VM],” and “[i]f the virtual machine host isn’t overloaded,” it or a “user can adjust CPU and memory shares” for the hosted virtual machines. EX1009 (Wolf), 514-15; *see also* EX1013 (Wood), 229 (“A workload increase can be handled by increasing the resources allocated to a virtual server....”).

30. Computer data and computer resources were commonly accessed or referenced at different types of virtual or non-virtual locations. For example, data’s location might refer to the data’s general location (e.g., a directory, a general computing environment, or a remote storage service where the data was located) or a more specific location of the data (e.g., an address or file path). *See, e.g.*, EX1009 (Wolf), 242 (“VMFolder: Location of VM’s files on the host system”), 246-57 (discussing “[l]ocation” of VM files, drives/folders containing VMs, and snapshots); EX1015 (VMWare vSphere), 56 (“When you manage the virtual infrastructure, you access objects and their properties and methods based on their location in the inventory.”); EX1021 (NIST Cloud Computing), 8 (discussing resources’ general locations); EX1031 (NIST IT Asset Mgmt), 1 (Executive Summary) (“The NIST Cybersecurity IT Asset Management Practice Guide is a proof-of-concept solution demonstrating commercially available technologies that can be implemented to track

the location and configuration of networked devices and software across an enterprise. Our example solution spans traditional physical asset tracking, IT asset information, physical security, and vulnerability and compliance information. Users can now query one system and gain insight into their entire IT asset portfolio”). Even for data in a virtualized environment, its location could refer to a virtual location (e.g., the virtualized environment where the data is located, or a more specific virtual location such as a virtual address or virtual file path) or a non-virtual location (e.g., a general location of a storage service or a more specific physical storage location), since data was routinely stored and accessed in non-virtual storage regardless of whether the data also exists in the virtualization layer. *See, e.g.*, EX1009 (Wolf), 242, 246-57; EX1010 (Waldspurger), 3-4 (discussing exemplary mappings between virtual and non-virtual locations). External systems typically communicated with a virtualization layer via an associated endpoint such as an IP address, MAC address, etc. EX1080 (Lambeth), 5:34-42 (“[I]n one conventional technique, some address mapping between guest physical addresses, hypervisor virtual addresses, and machine addresses associated with the network frames is performed to determine the destination of the network frame.”); EX1054 (Ackley) 1:31-42 (“[A] method for receiving, in a hypervisor, a packet including a destination media access control (MAC) address field having a MAC address of the hypervisor

and a destination Internet protocol (IP) address field having an IP address of a virtual machine (VM) coupled to the hypervisor. In turn, the hypervisor can determine a MAC address of the VM using the IP address of the VM and apply the VM MAC address to the destination MAC address field of the packet and forward the packet to the VM from the hypervisor.”); EX1074 (VMware Infrastructure),⁵ 12 (“After authentication, a view of the resources and virtual machines that belongs to the user is presented. For accessing a virtual machine console directly, the VI Client first obtains the virtual machine location from the VirtualCenter Management Server

⁵EX1074 is a copy of “VMware Infrastructure Architecture Overview,” published in 2006 by VMware, Inc. This copy was originally obtained from https://www.vmware.com/pdf/vi_architecture_wp.pdf. However, the document has been removed from this URL. An archived version of the document can be found using the internet archive Wayback Machine at the following URL: https://web.archive.org/web/20181010140113/https://www.vmware.com/pdf/vi_architecture_wp.pdf, which was archived on October 10, 2018. The archived document contains the same title, company, and copyright date as shown in EX1074.

through the VI API.”); EX1048 (Price), ¶21 (“Virtual machine manager 130 interfaces can include interfaces and application programming interfaces (APIs) that can provide operations and accessing including guest management, offline registry access, virtual disk access, and other features of virtual machines that maybe running or accessible through a particular virtualization host environment. As an example, a virtual machine manager associated with VMware™ virtualization tools can include such interfaces as the VIX API and VDDKAPI, among others. Further, virtual machine manager-provided interfaces can be leveraged ... to allow outside security tools access to firewalled and other protected virtualized appliances and resources.”). VMs used virtual memory addresses to map the location of data as it appeared inside a virtual machine to a physical location on a disk. *E.g.*, EX1009 (Wolf), 2 (“The virtualization layer is responsible for mapping virtualized hardware to the host’s physical resources.”). A single file or piece of data would have both a virtual memory address and a physical location on disk, mapped together by the virtualization layer software, as illustrated in the following example:

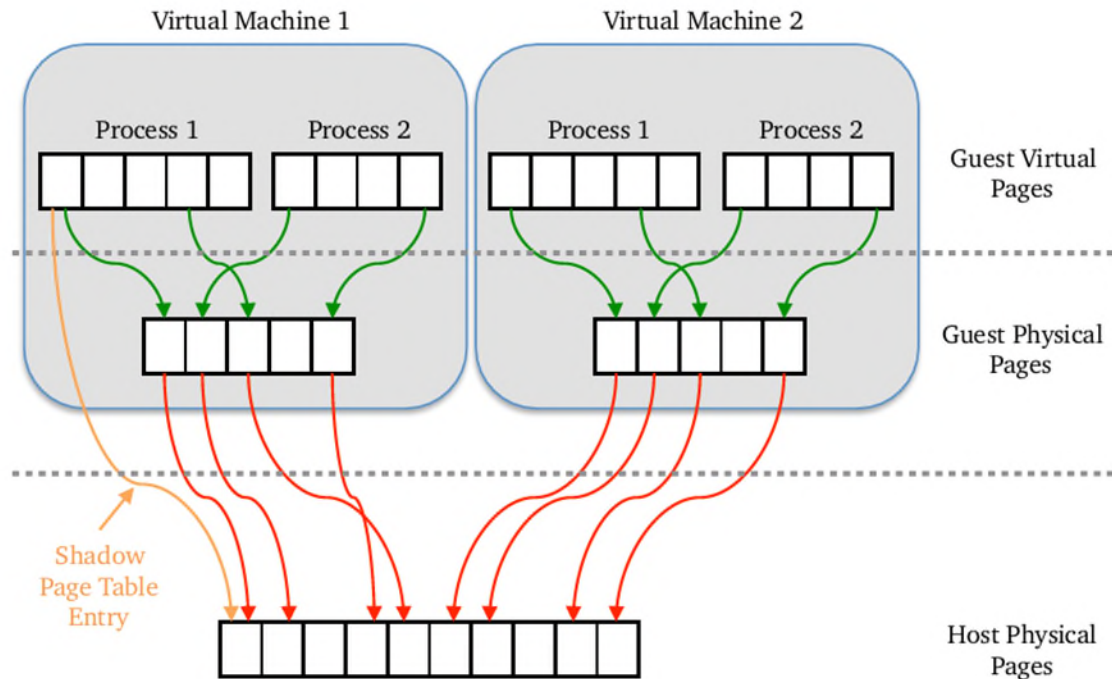


Figure 2.1: Depiction of two-level page mapping for two virtual machine guests, their processes on the same host, and shadow page table mappings. Figure borrowed with modification from VMware Performance Evaluation of Intel EPT Hardware Assist. http://www.vmware.com/pdf/Perf_ESX_Intel-EPT-eval.pdf

EX1014 (Klempner), 22.⁶ To a guest VM, files appeared to be stored in a contiguous “logical volume” or “virtual disk drive,” but the file data could be

⁶EX1014 is a copy of “Efficient Hypervisor Based Malware Detection,” by P. Klempner, published in May 2015 by Carnegie Mellon University. This copy was obtained from

physically stored in multiple discontinuous physical drives. See EX1015 (VMware vSphere), 124 (“Each datastore is a logical container, analogous to a file system on a logical volume, where the host places virtual disk files and other virtual machine files. Datastores hide specifics of the physical storage device and provide a uniform model for storing virtual machine files.”).⁷

31. Software containers were another form of virtualization that were well understood and widely used at the time. A container was an isolated, lightweight

https://kilthub.cmu.edu/articles/thesis/Efficient_Hypervisor_Based_Malware_Detection/6716180.

⁷ EX1015 is a copy of the vSphere Web Services SDK Programming Guide, published in April 2018 by VMware, Inc. This copy was obtained from <https://vdc-download.vmware.com/vmwb-repository/dcr-public/cdbbd51c-4824-4a1b-ad43-45df55a76a76/8cb3ed93-cac2-46aa-b329-db5a096af5bc/vsphere-web-services-sdk-67-programming-guide.pdf>. To avoid overburdening the record, EX1015 includes only the front matter, chapter 2, chapter 4, chapter 5, chapter 8, chapter 9, chapter 10, and chapter 11, since the document is quite large and I only cite to chapters 2, 4, 5, 8, 9, 10, and 11.

silo for running an application on the host operating system. Software containers virtualized an operating system to run reusable software packages in various virtual environments. EX1016 (NIST Application Container), ii (“Application container technologies, also known as containers, are a form of operating system virtualization combined with application software packaging. Containers provide a portable, reusable, and automatable way to package and run applications. This publication explains the potential security concerns associated with the use of containers and provides recommendations for addressing these concerns.”);⁸ EX1017 (Rosendahl), 1:16-20 (“In the container model, a kernel of an operating system (e.g., Linux) allows for multiple isolated user-space instances, or ‘containers,’ executing simultaneously.”); *see also id.*, 1:20-35 (discussing operation and benefits of containers).

⁸ EX1016 is a copy of September 2017 publication by the National Institute of Standards and Technology (“NIST”), which is an agency within the U.S. Department of Commerce supporting innovation and competitiveness. This copy was obtained from <https://doi.org/10.6028/NIST.SP.800-190>.

B. Snapshots

32. Virtual machines provided various features to manage and backup resources. One such feature in most virtual machines was a “snapshot.” Snapshots were well known and had been used to create a point in time copy of a virtual machine since at least 2005, well before the ’798 patent. EX1009 (Wolf), 257 (“With online snapshots, you can create a point-in-time copy of a running virtual machine. With an online snapshot saved, you can revert to the snapshot later. This gives you the ability to create an image of a VM, make any changes you want, and with the click of a button roll the VM back to the time of the last snapshot.”); *see also* EX1018 (NIST Full Virtualization Technologies), 2-6 (“A snapshot is a record of the state of a running image, generally captured as the differences between an image and the current state. For example, a snapshot would record changes within virtual storage, virtual memory, network connections, and other state-related data. Snapshots allow the guest OS to be suspended and subsequently resumed without having to shut down or reboot the guest OS. Many, but not all, virtualization systems

can take snapshots.”).⁹ Further, snapshots had been used to create a copy of a VM on different hosts. EX1018 (NIST Full Virtualization Technologies), 2-6 (“On some hypervisors, snapshots of the guest OS can even be resumed on a different host.”). Most virtualization technologies offered utility functions that included the generation, storage, indexing, and import of snapshots. EX1019 (Rawat), Abstract; *see also* EX1020 (Sancheti), Abstract, 21:42-22:58 (describing use of API calls to interact with snapshots).

33. It was well known that snapshots could be taken of VMs while they were in a variety of different states (e.g., powered on, suspended, powered off, etc.). EX1015 (VMware vSphere), 164 (“The snapshot includes the state of the data on all virtual machine disks and the virtual machine power state (on, off, or suspended). You can take a snapshot when a virtual machine is powered on, powered off, or suspended.”); EX1064 (Lutas), ¶31 (“Some examples of forensic tools include snapshot-generating tools, programmed to obtain a memory snapshot of a client system or of a virtual machine executing on the respective client system.”); *see also*

⁹ EX1018 is a copy of a January 2011 NIST publication obtained from <https://doi.org/10.6028/NIST.SP.800-125>.

EX1052 (VirtualCenter)¹⁰, 203 (describing the different states of a VM). However, snapshots of VMs in different states commonly captured different data. In particular, additional run-time information was typically captured when a snapshot was taken of a running or suspended VM as compared to a VM that was fully off-line. EX1051

¹⁰ EX1052 is a copy of “VMware VirtualCenter User’s Manual version 1.4,” published by VMware, Inc. on July 7, 2006. This copy was originally obtained from https://www.vmware.com/pdf/vc_users14.pdf. To avoid overburdening the record, EX1052 includes only the front matter, chapter 8, and chapter 17, since the document is quite large and I only cite to chapters 8 and 17. However, the document has been removed from this URL. Nevertheless, the same user’s manual can be found in its entirety using the internet archive Wayback Machine at the following URL:

https://web.archive.org/web/20160913032428/https://www.vmware.com/pdf/vc_users14.pdf, which was archived on September 13, 2016. The archived user’s manual contains the same title, company, version number, revision number, and copyright dates as shown in EX1052.

(Workstation Pro)¹¹, 119 (“A snapshot includes the contents of the virtual machine memory....”); EX1049 (Huseinovic)¹², 940 (“Physical memory can contain some important data....If the computer is set to off, the data are [sic] lost and cannot be retrieved for later analysis.”).

34. That is, when a snapshot was taken of a running or suspended VM, the snapshot typically included page files of a couple different types. These pages files contained run-time information about the state of the VM at the time the snapshot

¹¹EX1051 is a copy of “Using VMware Workstation Pro: VMware Workstation Pro 14.0,” published by VMware, Inc. on December 21, 2017. This copy was obtained from <https://docs.vmware.com/en/VMware-Workstation-Pro/14.0/workstation-pro-14-user-guide.pdf>. To avoid overburdening the record, EX1051 includes only the front matter, chapter 3, chapter 4, and chapter 12, since the document is quite large and I only cite to chapters 3, 4, and 12.

¹²EX1049 is a copy of “Virtual Machine Memory Forensics,” by A. Huseinović and S. Ribić, published by IEEE in 2013 as a part of the 21st Telecommunications Forum Telfor (TELFOR). This copy was obtained from <https://ieeexplore.ieee.org/abstract/document/6716386>.

was taken, such as a list of system processes and their current state (e.g., suspended, running, terminated, etc.). Two types of these page files included (1) page files located on a disk used for memory overflow and (2) page files that represented the state of dynamic machine memory (e.g., random access memory or “RAM”).

35. First, the physical memory overflow page file was a well-known type of page file (sometimes referred to as a swap file) that was routinely used in popular operating systems (“OSs”) such as Windows™ and Linux. These OSs maintained the page file on a disk and used it as a memory overflow (sometimes referred to as a memory extension) when the actively running processes were demanding more RAM than was currently available. EX1051 (Workstation Pro), 297 (referring to a disk containing “the Linux swap file or the Windows page file”); EX1069 (Vorbach), 18:23-32 (referring to data being “loaded/off-loaded from/to the page file (MS Windows) or swap partition (e.g. Unix, Linux, etc.)”).

36. Second, page files representing the state of machine memory (e.g., RAM) were similarly well known and routinely included in a VM’s snapshot if that snapshot was taken when the VM was not fully powered off. These types of snapshots commonly stored the current state of the VM’s machine memory in a vmem file. As a result, the vmem file typically included runtime information about the VM, such as the currently running processes and applications. EX1051

(Workstation Pro), 77 (“Each snapshot of a virtual machine that is powered on has an associated .vmem file, which contains the guest operating system main memory, saved as part of the snapshot.”); EX1049 (Huseinovic), 940 (“File vmem is a virtual machine paging file.”), 941 (the .vmem file is “created by taking [sic] snapshot” and contains a list of processes running on the VM); EX1051 (Workstation Pro), 77 (“Each snapshot of a virtual machine that is powered on has an associated .vmem file, which contains the guest operating system main memory, saved as part of the snapshot.”); EX1064 (Lutas), ¶¶23 (“[A] page represents the smallest unit of virtual memory that can be individually mapped to a physical memory of a host system.”), 31 (“A snapshot of an OS kernel may include, among others ... in memory kernel threads and their corresponding stacks, the OS’s kernel data structures—such as the list of loaded modules, the list of processes, etc. An exemplary snapshot of an application comprises a copy of a memory image of the application, including its code and data sections, the in memory stacks used by the applications threads, the heap memory pages of the respective application, etc.”).

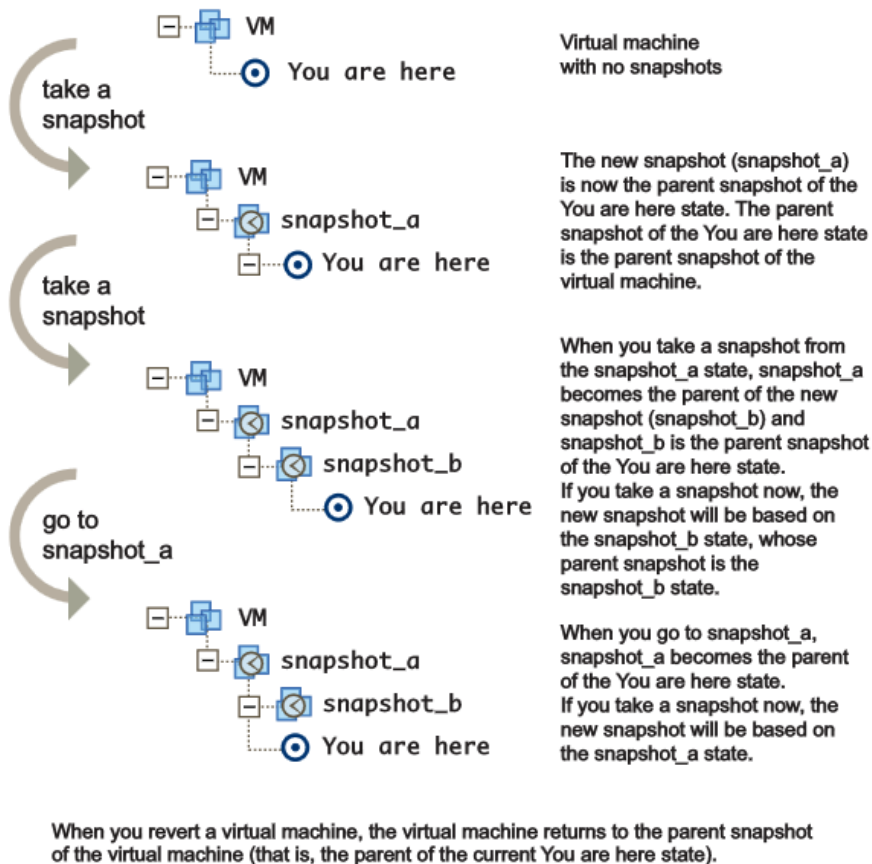
37. It was also well known that snapshots may contain important and sensitive data that is important to keep secure. EX1018 (NIST Full Virtualization Technologies), 2-6 (“Note that one of the biggest security issues with images and snapshots is that they contain sensitive data (such as passwords, personal data, and

so on) just like a physical hard drive. Because it is easier to move around an image or snapshot than a hard drive, it is more important to think about the security of the data in that image or snapshot. Snapshots can be more risky than images because snapshots contain the contents of RAM memory at the time that the snapshot was taken, and this might include sensitive information that was not even stored on the drive itself.”). Because snapshots provided insight into the VM they were taken from (including the machine’s memory), they could be used to provide insight into vulnerabilities of the virtual machine.

38. A snapshot could be comprised of one or more files, depending on the mechanism that was used to generate the snapshot. EX1071 (Bezbaruah), 6:35-39 (“Traditionally, a point in time snapshot of a VM is represented by VM checkpoints.”); EX1072 (Derbeko), 4:1-13 (snapshot may include “deltas”). As with computer data more generally, a snapshot’s “location” could refer to the types of locations I discussed above in paragraph 30. *See, e.g.*, EX1009 (Wolf), 257 (“The snapshot information will be saved in the same location as the VM’s virtual disk files.”); EX1015 (VMWare vSphere), 56 (“When you manage the virtual infrastructure, you access objects and their properties and methods based on their location in the inventory.”). It was common for a hypervisor to have a default (predetermined) location where VM snapshot files were stored. Locating snapshot

files was often facilitated by utilities provided by the virtualization hypervisor via a console, Graphical User Interface (“GUI”) or Application Programming Interface (“API”). For example, VMware’s vSphere Web Services SDK Programming Guide, released in 2018, mentions on page 164: “When you create a snapshot, the system creates a delta disk file for that snapshot in the datastore and writes any changes to that delta disk. You can later revert to the previous state of the virtual machine. The VirtualMachine object has methods for creating snapshots, reverting to any snapshot in the tree, and removing snapshots.” EX1015 (VMware vSphere), 164-66. Moreover, the VirtualMachine.CreateSnapshot_Task method created a new snapshot of a virtual machine. *Id.* The method allowed a user to specify a name for the snapshot. *Id.* The following figure illustrates the type of GUI representation that a user could interact with when managing snapshots in VMware products:

Figure 11-1. Virtual Machine Snapshots



Id.; see also EX1009 (Wolf), 32 (“All resources can be remotely managed...”), 221 (“VirtualCenter uses a central console so you can access and manage all your VMs from one location.”), 221 (“The Scripting API has scripts to start all VMs, list all VMs, create and commit a redo log, and enumerate a VM’s IP.”), 257-60 (snapshot generated via VMware management UI).

39. For a service seeking to access an existing snapshot, the snapshot location was routinely determined both physically, where the files are located, and logically, that is, what computing asset this snapshot of the virtual disk belonged to.

Both of these could be ascertained using the metadata that accompanied the virtual machine snapshot when the snapshot was generated. EX1015 (VMware vSphere), 164-66.

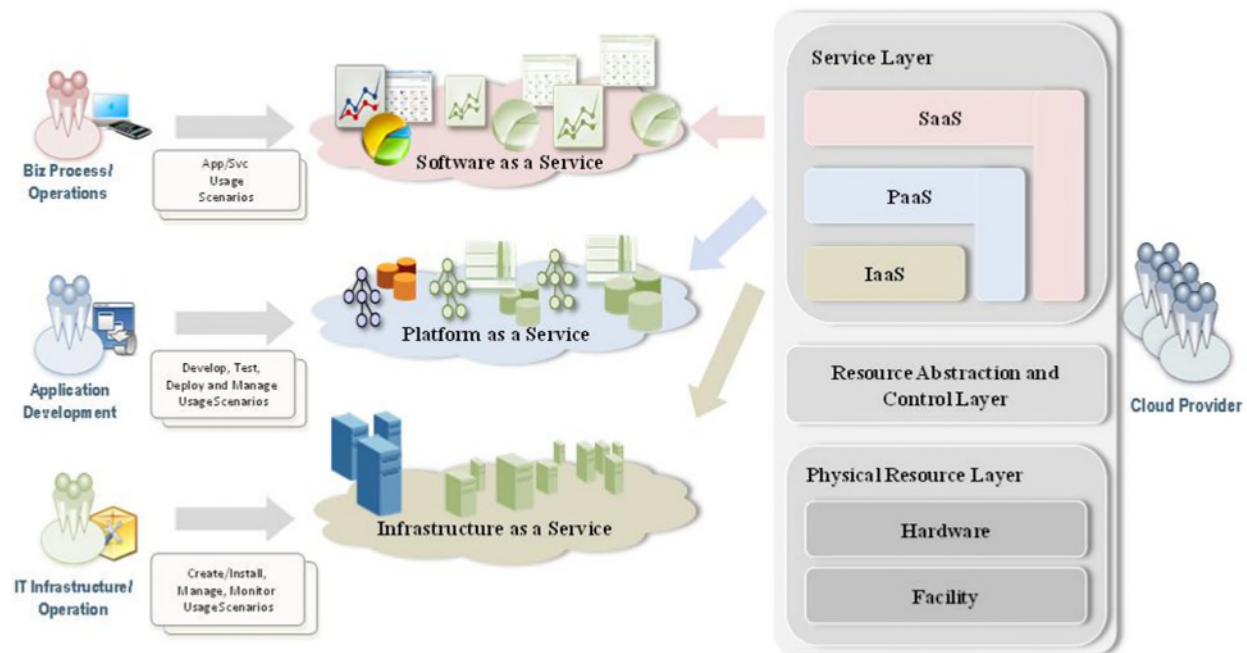
C. Cloud Computing

40. Cloud computing typically provided on-demand availability of computer system resources (storage and compute power) from a shared pool and over a remote network. EX1021 (NIST Cloud Computing), 1 (“Cloud Computing enables ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”).¹³

41. Cloud computing services were commonly built on a three-layered framework: a service layer, a resource abstraction layer, and a physical resource layer. This three-layered framework enabled three main types of service model:

¹³EX1021 is a copy of a July 2013 NIST publication obtained from <http://dx.doi.org/10.6028/NIST.SP.500-291r2>.

SaaS (software as a service), PaaS (platform as a service), and IaaS (infrastructure as a service). The figure below illustrates a cloud computing environment:



EX1021 (NIST Cloud Computing), 18.

42. “The top layer is the service layer, where a cloud provider defines and provisions each of the three service models. This is where cloud consumers consume cloud services through the respective cloud interfaces.” EX1021 (NIST Cloud Computing), 19. In the SaaS model, applications were accessible from various client devices through the service layer, either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. Cloud providers also offered VMs as cloud instances that a user could access through APIs. See, e.g., EX1022

(VMware Public Cloud), 29;¹⁴ EX1015 (VMware vSphere), 55-58, 62-66; EX1021 (NIST Cloud Computing), 94-95. Through APIs, GUIs, or other consoles, a user

¹⁴ EX1022 is a copy of “Public Cloud Service Definition,” by A. Roberts, published in January 2018 by VMware, Inc. This copy was originally obtained from <https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/vcat/vmware-public-cloud-service-definition.pdf>. However, the document has been removed from this URL. Nevertheless, an earlier version of EX1022 can be found using the internet archive Wayback Machine at: <https://web.archive.org/web/20161120131550/http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/vcat/vmware-public-cloud-service-definition.pdf>, which was archived on November 20, 2016, and contains substantively the same cited information. Furthermore, an archived copy of EX1022 can be found using the internet archive Wayback Machine at: <https://web.archive.org/web/20220120040737/https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/vcat/vmware-public-cloud-service-definition.pdf>, which was archived on January 20, 2022. This archived copy has the

could control and manage their cloud resources, for example, by sending queries and commands to a hypervisor. EX1015 (VMware vSphere), 164-66, 118 (Fig 8-2), 138 (Fig. 9-1).

43. “The middle layer is the resource abstraction and control layer.... The layer typically includes software elements such as hypervisors, virtual machines, virtual data storage, and other resource abstraction and management components needed to ensure efficient, secure, and reliable usage.” EX1021 (NIST Cloud Computing), 19. With the interfaces provided by the service layer, a user could interact with the abstraction layer to, for example, spin up virtual machines, allocate virtual storage, and perform data backups. Virtualization technology (including virtual machines and virtual disk storage) was thus central to cloud computing.

44. “The lowest layer in the framework is the physical resource layer, which includes all the physical computing resources.” EX1021 (NIST Cloud Computing), 19. The physical layer’s resources (including servers, physical disk

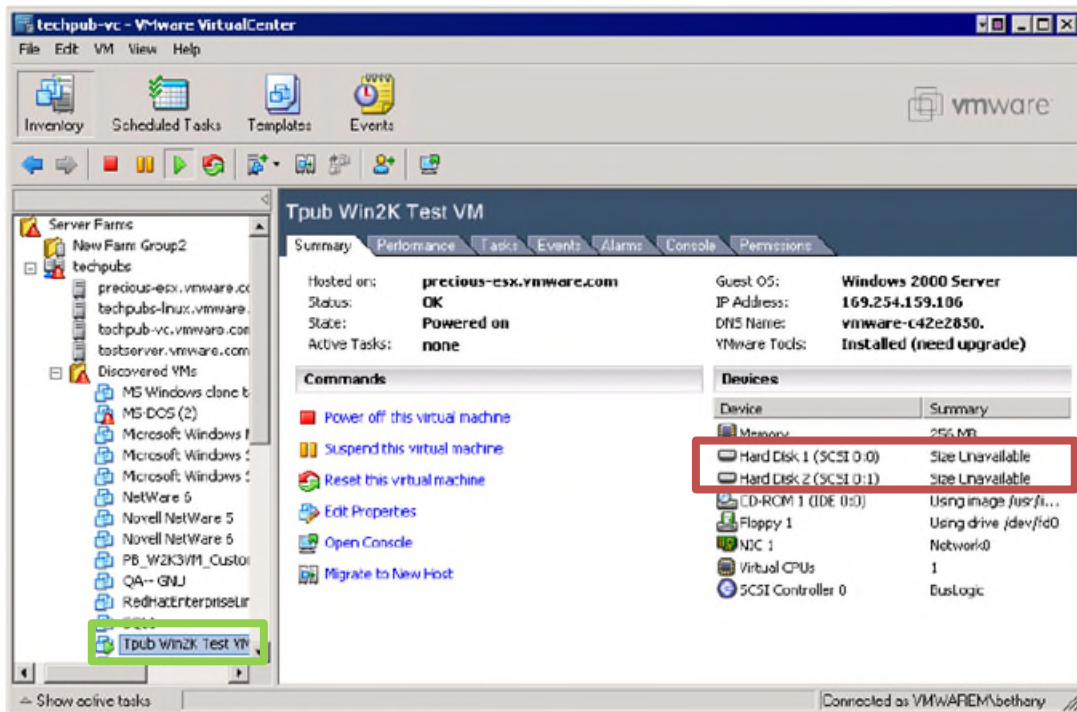
same version number, publication date, author, and copyright date as shown in EX1022.

storage, and network components) were frequently concentrated in one or more “data centers.” EX1013 (Wood), 229 (“Data centers—server farms that run networked applications—have become popular in a variety of domains such as web hosting, enterprise systems, and e-commerce sites.”).

45. Furthermore, off-the-shelf virtualization tools (e.g., VMWare® VirtualCenter) routinely used APIs to query for information about a target VM, such as the identity and location of the virtual disk(s) associated with the target VM. These API tools would often provide a graphical user interface (“GUI”) providing various virtual-resource-management features such as allowing users to view their VMs and associated information. For example, it was commonplace for GUIs to allow users to select a particular VM, at which point more detailed information about the VM (e.g., the names and locations of associated virtual disks) would be presented to the user based on API calls. The annotated figure below depicts an example of this from the well-known VirtualCenter tool provided by VMware®. In this example, the user has selected a particular VM (outlined in green below), and information about the VM (retrieved via API calls) is displayed in the adjacent window:

Viewing the Virtual Machine Summary Tab

The virtual machine **Summary** tab displays information that applies to a virtual machine and its resources.



Inventory > Virtual Machine > Summary Tab

46. EX1052 (VirtualCenter), 445-46 (green and red annotations added).

The portion outlined in red above shows the identification of two virtual disks allocated to the VM (“Hard Disk 1” and “Hard Disk 2”) which are displayed along with their respective locations (“SCSI 0:0” and “SCSI 0:1”). “SCSI” refers to the “Small Computer System Interface,” which was a well-known standard for communicating with peripheral devices that was commonly used for local disks, and “0:0” and “0:1” refer to the SCSI address (here a bus number followed by a device ID). Similar standards such as “iSCSI” were routinely used for remote disks. *E.g.*,

EX1053 (Prahlad), ¶36 (“The computing device 125 is connected to the primary storage data store 135 via the SAN 130, which may be any type of SAN (e.g., a Fibre Channel SAN, an iSCSI SAN, or any other type of SAN). The primary storage data store 135 stores the virtual disks 140 (shown individually as virtual disks 140a and 140b) of the virtual machines....”); EX1048 (Price), ¶¶21 (“Virtual machine manager 130 interfaces can include interfaces and application programming interfaces (APIs) that can provide ... virtual disk access....”), 42 (“[S]ecurity server 305 can use a virtual disk API of the virtual machine manager 330 to mount a virtual image of an offline virtual machine and read files of the offline virtual machine ... The security server 305 can thereby collect registry, file, and policy setting values from the images and files obtained over the virtual machine manager 330 API(s) and then analyze these values for vulnerabilities....”). Furthermore, this type of API-based communication was not limited to UIs. Common cloud/virtualization platforms included APIs that were routinely used by GUIs and other types of systems for querying information about virtual resources. For example, in the vSphere platform from VMware®, the “find_entity_views()” API call would provide the inventory of the user’s VMs, including allocated virtual disks, and

“get_inventory_path()” would provide locations of the identified disk(s). EX1055 (Virtual Disk),¹⁵ 55-56, 13, 23, 32-33, 53-54, 68-69.

47. These API calls, whether initiated through the use of a GUI or another programming interface, were routinely used as building blocks of more complex tasks, such as displaying summary information to a user about a target VM (EX1052 (VirtualCenter), 445-46) or performing a virus scan of the virtual disks of a target VM (EX1055 (Virtual Disk), 55-56). It was also well known to use these types of APIs as part of capturing a snapshot. For example, a VM’s virtual disks were routinely identified using API calls prior to capturing a snapshot. EX1020 (Sancheti), 21:9-41 (“At block 410, the management component 254 discovers and

¹⁵EX1055 is a copy of “VMWare Virtual Disk Programming Guide,” published by VMware, Inc. on March 25, 2016. This copy was obtained from https://vdc-download.vmware.com/vmwb-repository/dcr-public/ae1a858c-2f59-4a75-b333-f2ddfa631fbd/4ae542fa-c7ec-4713-a3a0-6fef938b03d1/vddk60_programming.pdf. To avoid overburdening the record, EX1055 includes only the front matter, chapter 1, chapter 3, chapter 4, chapter 6, and chapter 7, since the document is quite large and I only cite to chapters 1, 3, 4, 6, and 7.

identifies the virtual disks and the PRDM-mapped LUNs associated with the virtual machine, for example by using APIs to query the virtual machine host 105 on which the virtual machine resides or a virtual machine manager 150.”), Fig. 4; EX1051 (Workstation Pro), 47, 125 (describing the selection of which virtual disks to include in the snapshot). The locations of identified virtual disks were also routinely queried using API calls. EX1020 (Sancheti), 21:9-22:18 (“The virtual machine tracking data structure may also include a mount point column 710 that indicates the drive location or drive identifier (e.g. C:/) where each virtual disk or PRDM-mapped LUN is mounted within the file system of the virtual machine....”), Fig. 4; EX1053 (Prahlad), ¶¶87-92 (describing the process of determining the files, volumes, or disks of the target VM to be included in the snapshot), Fig. 7 (same); *see also* EX1051 (Workstation Pro), 119 (“A snapshot includes the contents of the virtual machine memory, virtual machine settings, and the state of all the virtual disks.”).

D. Cyber Security

48. Cyber security has long been an important topic in IT applications. Below, I will first describe general threats to computing resources, how these threats apply to virtual machines, and how they appear in the cloud. Next, I will discuss how these threats and vulnerabilities were often detected and the common tools used to do so. Finally, I will discuss common responses to threats and vulnerabilities once they are detected.

1. Types of Security Risks

49. It was well known that computer systems faced various types of security risks, including threats that took advantage of vulnerabilities in the computer system. EX1023 (NIST Security Patches), 2 (“Vulnerabilities are weaknesses in software that can be exploited by a malicious entity to gain greater access and/ or permission than it is authorized to have on a computer.”).¹⁶ For example, computer systems were often targeted with “malicious code and malicious software that are covertly inserted into an information technology (IT) system to compromise the confidentiality, integrity, or availability of the data, applications, or operating system, or to annoy or disrupt the system’s owner,” commonly referred to as malware. EX1024 (NIST Preventing Malware), 1.¹⁷ Old, outdated software often contained bugs that, if not patched, could be exploited. EX1023 (NIST Security

¹⁶EX1023 is a copy of an October 2002 NIST publication obtained from https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=150702.

¹⁷EX1024 is a copy of a December 2005 NIST publication obtained from <https://csrc.nist.gov/csrc/media/publications/shared/documents/itl-bulletin/itlbul2005-12.pdf>.

Patches), 1 (“Despite extensive testing, all operating systems and applications are released with bugs (errors in the software) that affect security, performance, and stability.... Many security related bugs are generally discovered only after a large number of users start using the software, and hackers and independent testers start attempting to compromise it. Once a bug is discovered, the software manufacturer often releases a piece of software to correct the bug. This software is often called a patch, hotfix, or service pack.”).

50. Many vulnerabilities were catalogued by individuals and organizations in the field and published as a set of Common Vulnerabilities and Exposures (“CVE”) to facilitate identification and mitigation of these known threats. EX1023 (NIST Security Patches), 1-3 (“Technical vulnerability experts from 31 industry, academia, and government organizations vote on the common names. CVE provides the computer security community with: a comprehensive list of publicly known vulnerabilities, an analysis of the authenticity of newly published vulnerabilities, and a unique name to be used for each vulnerability.”). Cyber security assessments were also concerned with vulnerabilities that were unknown at the time. For example, computers were often scanned to detect potential problems related to unknown software vulnerabilities so that these vulnerabilities could be better understood and resolved. EX1025 (NIST Incident Handling), 12 (“More information may need to

be provided in some cases—for example, if a server appears to have been compromised through an unknown software vulnerability.”).¹⁸ The European Union Agency for Network and Information Security (ENISA) “Threat Landscape Report” for 2018 shows that the top threats and vulnerabilities were malware and web-based threats, as illustrated below:



ENISA Threat Landscape Report 2018
 ETL 2018 | 1.0 | External | January 2019

Top Threats 2017	Assessed Trends 2017	Top Threats 2018	Assessed Trends 2018	Change in ranking
1. Malware	➡	1. Malware	➡	➡
2. Web Based Attacks	⬆	2. Web Based Attacks	⬆	➡
3. Web Application Attacks	⬆	3. Web Application Attacks	➡	➡
4. Phishing	⬆	4. Phishing	⬆	➡
5. Spam	⬆	5. Denial of Service	⬆	⬆
6. Denial of Service	⬆	6. Spam	➡	⬇
7. Ransomware	⬆	7. Botnets	⬆	⬆
8. Botnets	⬆	8. Data Breaches	⬆	⬆
9. Insider threat	➡	9. Insider Threat	⬆	➡
10. Physical manipulation/ damage/ theft/loss	➡	10. Physical manipulation/ damage/ theft/loss	➡	➡
11. Data Breaches	⬆	11. Information Leakage	⬆	⬆
12. Identity Theft	⬆	12. Identity Theft	⬆	➡
13. Information Leakage	⬆	13. Cryptojacking	⬆	NEW
14. Exploit Kits	⬆	14. Ransomware	⬆	⬇
15. Cyber Espionage	⬆	15. Cyber Espionage	⬆	➡

Legend: Trends: ⬆ Declining, ➡ Stable, ⬆ Increasing
 Ranking: ⬆ Going up, ➡ Same, ⬇ Going down

Table 1- Overview and comparison of the current threat landscape 2018 with the one of 2017

¹⁸EX1025 is a copy of an August 2012 NIST publication obtained from <http://dx.doi.org/10.6028/NIST.SP.800-61r2>.

EX1026 (ENISA Report, 2018), 9.¹⁹

51. Another type of common security risk in computer systems involved the potential exposure of certain types of data. It was well-known that it was dangerous for certain types of sensitive data—for example, financial information, personally identifiable information, passwords, private encryption keys, etc.—to be left unencrypted on a computer system or otherwise left vulnerable to exposure. *See, e.g.,* EX1036 (Wysopal), 12:36-45 (“Each of the FIPS factors can be assigned a rating (e.g., as n/a, minimal, moderate or serious), and in some embodiments certain factors are weighted more than others according to relative importance, (e.g., as defined by a user or industry standards). For example, a virtual machine that includes an application that processes healthcare data including personally

¹⁹EX1026 is a copy of the “ENISA Threat Landscape Report 2018,” published in January 2019 by the European Network and Information Security Agency (“ENISA”). This copy was obtained from <https://www.enisa.europa.eu/publications/enisa-threat-landscape-report-2018>. EX1026 confirms my recollection and understanding of the landscape of known and unknown cyber threats prior to 2019.

identifiable information may accord a rating of ‘serious’ to factors such as damage to reputation, liability, unauthorized release of sensitive information and criminal liability, but ‘n/a’ for personal safety.”); EX1056 (Chari), ¶¶38 (“Sensitive data is information that a user of a virtual machine does not want to be made public or made available to other users after the user has finished using the virtual machine. Types of sensitive data that may be contained within a virtual machine may be, for example, passwords, credit card numbers, social security numbers, bank account numbers, cryptographic keys, user-generated data, user history data, and paths to locations of the sensitive data.”), 60-61 (“Vulnerabilities within the cloud computing environments may include virtual machine images that are not completely sanitized of the sensitive data after use.”); EX1057 (CWE200),²⁰ 1-2; EX1058 (CWE311),²¹

²⁰EX1057 is an archived copy of “CWE-200: Information Exposure,” archived on January 26, 2019, by the Internet Archive’s “Wayback Machine” at <https://web.archive.org/web/20190126150012/https://cwe.mitre.org/data/definitions/200.html>. The acronym CWE refers to “common weakness enumeration.”

²¹EX1058 is an archived copy of “CWE-311: Missing Encryption of Sensitive Data,” archived on January 17, 2019, by the Internet Archive’s “Wayback

1-4; EX1059 (CWE359),²² 1-3. Such data was often targeted by malware or malicious intruders.

52. Virtual machines often had the same vulnerabilities as traditional physical machines, as well as additional vulnerabilities related to the virtualization itself. EX1018 (NIST Full Virtualization Technologies), 3-1 (“Migrating computing resources to a virtualized environment has little or no effect on most of the resources’ vulnerabilities and threats. For example, if a service has certain vulnerabilities and that service is moved from a non-virtualized server to a virtualized server, the service is still just as vulnerable to exploitation. However, the use of virtualization may help

Machine” at

<https://web.archive.org/web/20190117150049/https://cwe.mitre.org/data/definitions/311.html>.

²²EX1059 is an archived copy of “CWE-359: Exposure of Private Information (‘Privacy Violation’),” archived on January 13, 2019, by the Internet Archive’s “Wayback Machine” at

<https://web.archive.org/web/20190113052451/https://cwe.mitre.org/data/definitions/359.html>.

reduce the impact of such exploitation—but virtualization may also provide additional attack vectors, thus increasing the likelihood of successful attack.”). As I discussed above, virtualization often added complexity in the form of a “host computer” and a “hypervisor” to manage the virtual machine, with each component potentially having its own vulnerabilities. *Id.*, ES-2 (“The security of the entire virtual infrastructure relies on the security of the virtualization management system that controls the hypervisor and allows the operator to start guest OSs, create new guest OS images, and perform other actions.”), 4-1 (same), 4-3 (“Because hosted virtualization systems are run under host OSs, the security of every guest OS relies on the security of the host OS.”).

53. It was also well known at the time that cloud computing systems, including the data centers and virtual machines used in cloud systems, were common targets of cyber attacks. For example, Verizon’s Data Breach Investigations Report for 2019 shows that servers were the number one asset to be targeted by hacking and malware, as illustrated below:

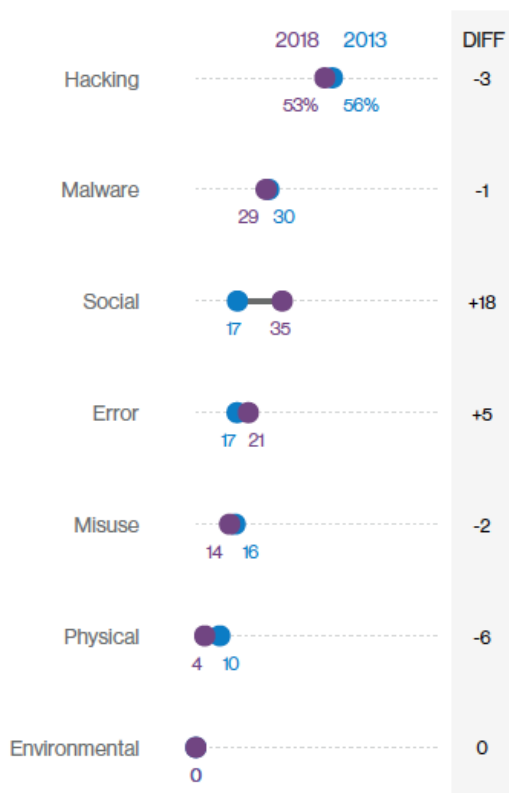


Figure 9. Threat actions in data breaches over time
n=2,501 (2013), n=1,638 (2018)

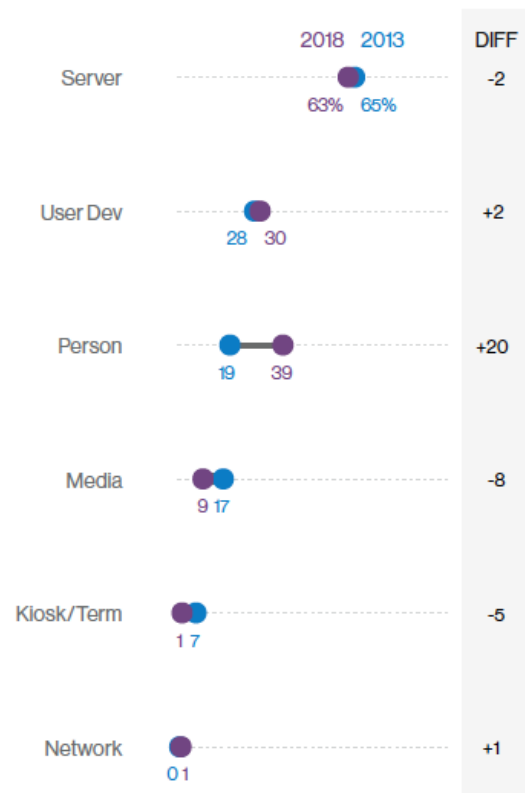


Figure 10. Asset categories in data breaches over time
n=2,294 (2013), n=1,513 (2018)

EX1027 (Verizon DBIR), 8.²³ Cloud computing systems were known to have many of the same vulnerabilities as non-cloud systems, while also presenting additional

²³EX1027 is a copy of the 2019 Data Breach Investigations Report, summarizing data from 2018, published by Verizon in May 2019. This copy was obtained from <https://www.verizon.com/business/resources/reports/2019-data->

risks and vulnerabilities. EX1028 (Di Pietro), 171-72.²⁴ For example, the National Institute of Standards and Technology noted numerous security challenges facing cloud systems, including:

- Compromises to the confidentiality and integrity of data in transit to and from a cloud provider and at rest;
- Attacks which take advantage of the homogeneity and power of cloud computing systems to rapidly scale and increase the magnitude of the attack;
- A consumer's unauthorized access (through improper authentication or authorization, or exploit of vulnerabilities introduced maliciously or unintentionally) to software, data, and resources provisioned to, and owned by another authorized cloud consumer;

breach-investigations-report.pdf and confirms my recollection and understanding of the landscape of cyber threats facing cloud-computing systems prior to 2019.

²⁴EX1028 is a copy of “Virtualization Technologies and Cloud Security: Advantages, Issues, and Perspectives” by R. Di Pietro and F. Lombardi, published as a chapter in the book “From Database to Cyber Security” in November 2018. This copy was obtained from https://doi.org/10.1007/978-3-030-04834-1_9.

- Increased levels of network-based attacks that exploit software not designed for an Internet-based model and vulnerabilities existing in resources formerly accessed through private networks;
- ...
- Attacks that exploit inconsistencies in global privacy policies and regulations....

EX1021 (NIST Cloud Computing), 45-46. Others noted that cloud virtual machines frequently encountered malware injection attacks. *See, e.g.*, EX1029 (Singh), 321 (“A first considerable attack attempt aims at injecting a malicious service implementation or virtual machine into the Cloud system. Such [c]loud malware could serve any particular purpose the adversary is interested in, ranging from eavesdropping via subtle data modifications to full functionality changes or blockings.”).

2. Detecting Security Risks

54. Users and administrators routinely sought to identify the types of security risks described above using various security analysis techniques. One common approach was to use software to scan computer systems for security vulnerabilities. *See* EX1030 (Bairwa), 116 (“Vulnerability assessment means identifying the vulnerabilities in the system before they could be used by anyone else with bad intentions of harming the network.... Vulnerability scanners are the ones that find out the vulnerabilities in the system which if accessed by a malicious

user or hacker can put the whole network system at risk.”). Such scanners were routinely used to evaluate operating systems, applications, files, configurations, and other aspects of computer systems (both physical and virtual) to identify potential vulnerabilities. See EX1031 (NIST IT Asset Mgmt), 22;²⁵ EX1032 (Li), ¶¶27, 43; EX1033 (Garfinkel), 2;²⁶ EX1030 (Bairwa), 116; EX1095 (Burdett), ¶43 (“The threat management facility 100 may provide configuration management, which may be similar to policy management, but may specifically examine the configuration set of applications, operating systems, hardware, and the like, and manage changes to their configurations. Assessment of a configuration may be made against a standard

²⁵EX1031 is a copy of a September 2018 NIST publication obtained from <http://doi.org/10.6028/NIST.SP.1800-5>.

²⁶EX1033 is a copy of “A Virtual Machine Introspection Based Architecture for Intrusion Detection,” by T. Garfinkel and M. Rosenblum, published in February 2003 as a part of the Network and Distributed System Security (NDSS) 2003 Symposium. This copy was obtained from <https://www.ndss-symposium.org/ndss2003/virtual-machine-introspection-based-architecture-intrusion-detection/>.

configuration policy, detection of configuration changes, remediation of improper configuration, application of new configurations, and the like.”).

55. Security scanners also routinely used static and/or dynamic analysis tools to detect sensitive data that might be at risk of exposure. For example, Chari describes scanning, labeling, and sanitizing sensitive data in VM images. EX1056 (Chari), ¶¶38 (“Labeler 222 is a software module that labels sensitive data contained within virtual machines.”), 64 (“Illustrative embodiments utilize a markup-based language for specification of sensitive data contained with virtual machine images and sanitization directives for sanitizing the sensitive data contained within the virtual machine images.”), 65 (discussing labeler and sanitizer for sensitive data), 66 (“Thus, illustrative embodiments focus on scanning and sanitization of virtual machine images in a cloud computing environment in order to remove sensitive data contained within the virtual machine images, which are not malware or viruses.”). The presence of sensitive data was routinely factored into security assessments. *E.g.*, EX1075 (Mohanty), 3:43-67 (alerts prioritized based on factors such as whether a VM contains sensitive data) 5:16-58 (same).

56. In the virtualization context, it was well known that security scans could be performed in a variety of different ways and on different types of data. For example, scanners were sometimes installed on the virtual machine as an agent that

would then run as an application on the targeted system, performing its scans while the system is in operation. *See, e.g.*, EX1034 (Tang), 5643 (“In the internal approach, antivirus software is installed in the VM and it often uses signature-matching technology to search viruses in files.”). Although directly inspecting the state of VMs with an agent provided high visibility, this visibility frequently came at the cost of weaker isolation, meaning that an attacker who successfully infected a guest VM’s operating system could overpower the agent. EX1033 (Garfinkel), 1; EX1037 (Bauman), 10:5 (discussing the advantages of agentless VM monitoring); *see also* EX1095 (Burdett), ¶¶5-6 (noting desirability of performing security analysis “without use of an agent installed on the template image used in cloud-based environments”).

57. Another common approach involved external scanning software that was not run as an application on the scanned system. As far back as 2003, the technique of Virtual Machine Introspection (VMI), also known as “external” or “out-of-VM” security, became an increasingly common approach involving scanning software located in the hypervisor that was not run as an application or agent on the scanned guest VM. *E.g.*, EX1033 (Garfinkel), 1; EX1034 (Tang), 5643 (“The external approach is inspired by the feature that the VMM has supervisory privilege on guest VMs. Antivirus software is totally moved out of the VM, and gets

necessary information from the guest OS using VMI techniques....”); *see also* EX1034 (Tang), 5643-644 (discussing hybrid approach); EX1035 (Lengyel), 389 (distinguishing its techniques from “prior agentless VMI approaches”).²⁷ For example, scanners often evaluated a virtual machine and its data in native format, such as an image file. EX1036 (Wysopal), 3:56-65 (“The analysis may also include scanning the executing image file (and, in some cases, executing applications within the virtual machine) using a network vulnerability scanner and/or a host vulnerability scanner.”); *see also* EX1018 (NIST Full Virtualization Technologies), 3-3 (“Image files can also be scanned to detect rootkits and other malware that, when running, conceal themselves from security software present within the guest OS.”). By 2015, VMI had developed into a well-established field, with a robust literature and a wide array of techniques. *See, e.g.*, EX1037 (Bauman), 10:9-10 (surveying 64

²⁷EX1035 is a copy of “Scalability, Fidelity and Stealth in the DRAKVUF Dynamic Malware Analysis System,” by T. Lengyel et al., published in December 2014 as a part of ACSAC '14: Proceedings of the 30th Annual Computer Security Applications Conference. I obtained this copy from <https://dl.acm.org/doi/pdf/10.1145/2664243.2664252>.

different out-of-VM approaches); EX1038 (Suneja), 133-34 (creating a taxonomy to summarize the field of VMI). VMI has been used to support a wide range of use cases, including kernel integrity and security monitoring, intrusion detection, anti-malware, firewall, virus scanning, cloud-wide information flow tracking, and policy enforcement. EX1038 (Suneja), 133.²⁸ It was also known that the efficiency of VMI security scanning techniques could be improved by scanning snapshots of guest VMs rather than scanning the live running guest VM instances, as illustrated by the diagrams below from a 2015 Ph.D. thesis:

²⁸EX1038 is a copy of “Exploring VM Introspection: Techniques and Trade-offs,” by S. Suneja et al., published in March 2015 as a part of VEE '15: Proceedings of the 11th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments. This copy was obtained from <https://dl.acm.org/doi/abs/10.1145/2817817.2731196>.

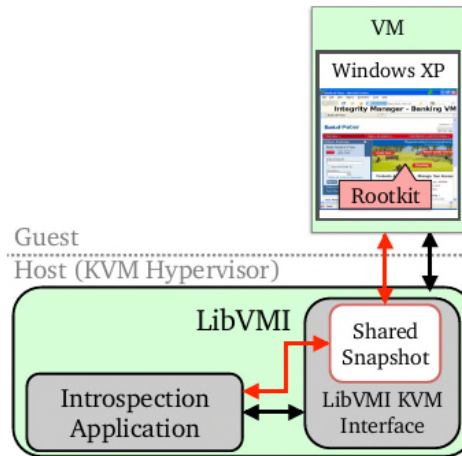


Figure 4.1: This block diagram illustrates how the shared snapshot interface is provides the Intro-
spection Application with a view into the memory of the VM Guest.

EX1014 (Klemperer), 25.

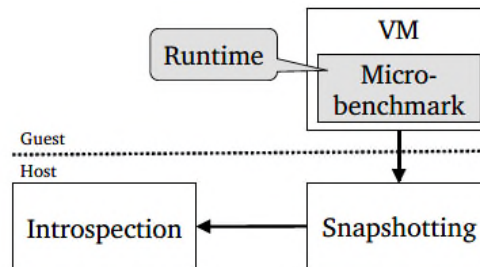
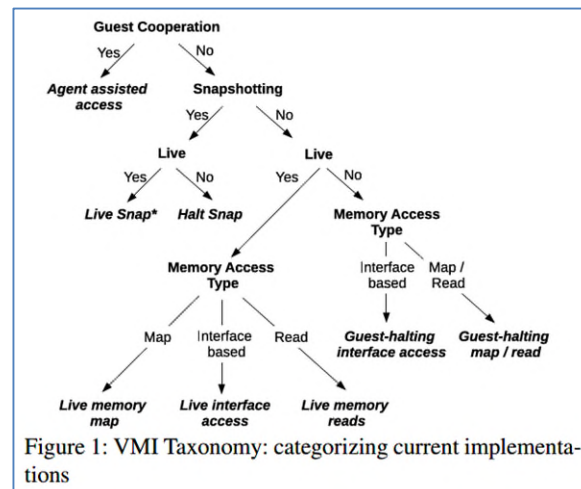


Figure 6.1: Microbenchmarks are used to quickly evaluate the effect of snapshotting over various
snapshotting regimes, guest loads, and introspection loads.

EX1014 (Klemperer), 55; *see also* EX1039 (Payne), 9 (“LibVMI programs can also use a static memory snapshot as a data source.”);²⁹ EX1038 (Suneja), 134-35 (discussing VMI methods that “do not require guest cooperation and distinguish themselves for producing a full copy of the guest’s memory image, while also pausing the guest to obtain a consistent snapshot”). The diagram below illustrates a subset of VMI methods that require no “agent assisted access” and use “snapshotting” to inspect virtual machines:



²⁹EX1039 is a copy of “Simplifying Virtual Machine Introspection Using LibVMI,” by B. Payne, published in September 2012 by the Sandia National Laboratory. It was obtained from <https://www.osti.gov/servlets/purl/1055635>.

EX1038 (Suneja), 134; *see also* EX1040 (Ammons), Abstract, ¶43 (describing use of snapshots to perform security analysis of virtual machines).

58. VMI approaches frequently employed the technique of parsing the scanned data to reformat it in a way that facilitated subsequent analysis. Data parsing was a common, well-understood technique in many computer contexts. It generally involves taking in unstructured data and converting it into a different format that could be processed more easily. *See* EX1041 (Kornfeld), 3:1-9 (“parsing algorithm” applied to data with “inconsistent internal structures” in order to “identify and organize record data associated with a computer.”). Its application in the scanning context was similarly well known. *See, e.g.,* EX1042 (Rubin), 10:45-55 (“In accordance with a preferred embodiment of the present invention, parser 220 controls the process of scanning incoming content.”), 12:54-60 (“Preferably, immediately after parser 220 performs a reduce operation, it calls analyzer 230 to check for exploits. Analyzer 230 searches for specific patterns of content that indicate an exploit. Preferably, parser 220 passes to analyzer 230 a newly created parsing node. Analyzer 230 uses a set of analyzer rules to perform its analysis.”); EX1035 (Lengyel), 389 (“The location of the kernel functions are determined by extracting information from the debug data provided for the kernel. The use of debugging information has been an established method in the forensics community

and it is the most convenient avenue to gain insight into the state of the operating system. In DRAKVUF we make use of the Rekall forensics tool [28] to parse the debug data provided by Microsoft to establish a map of internal kernel functions.”); EX1039 (Payne), 14 (“To support 64-bit guests, minor changes were needed in the PE parsing code that converts a Windows kernel symbol to a virtual address. On the other hand, major changes were required to support translating a 64-bit virtual address to a physical address. This required implementing the 64-bit address translation algorithms as described in the Intel Architecture Manuals.”); EX1043 (Jiang), 12:10 (“[W]e can further parse the raw memory image and traverse the doubly linked list to reconstruct detailed semantic information about each running process (e.g., its page table and memory layout in the mm struct data structure). From the same memory image, we can also cast and reconstruct a number of other important kernel data structures (e.g., the system call table, the interrupt descriptor table, and the kernel module list) and identify the areas that contain core kernel instructions or instructions in the loadable kernel modules.”); EX1048 (Price), ¶32 (“[A] machine image reader 282 can be provided for parsing data in retrieved virtual machine images to determine attributes of a virtual machine that can be scanned, checked, and verified in accordance security tasks performed by the security tool 205.”). Scanners typically evaluated the computer system to identify different types

of vulnerabilities. For example, scanners often checked the configuration files of operating systems and installed applications, looking for out-of-date software that needed to be patched. EX1044 (Giakouminakis), 3:50-54 (“For example, a vulnerability in the software resources can include, for example, software that is out of date, software that has known security weakness, configurations of software that have known security weaknesses, known bugs of software, etc.”); EX1079 (Carroll), 31-36 (discussing use of known logs and configuration files used in forensic analysis).

59. Scanners also routinely looked for signatures of known malicious code. EX1044 (Giakouminakis), 5:28-32 (“In embodiments, when the vulnerability is identified, the security tool 102 can identify whether the identified vulnerability is associated with known malware and/or is associated with exploit kits used for the creation of malware.”). Matching system data against public repositories was a common way of detecting security issues such as applications (or versions of applications) known to be vulnerable or files infected with malware. EX1076 (Milner), 2:24-67 (“To check the security, the security tool can be configured to compare the components of the software package to the black list. [...] A black list can include a list of archival files that are known to be insecure, such as known insecure packages/subpackages referenced in a Common Vulnerabilities and

Exposures (CVE) list.”); EX1077 (Khalid), 2:45-51 (“Moreover, an object engine may be configured to observe and/or extract characteristics of a suspicious object and compare such observed characteristics with those stored on one or more cyber-security databases, such as, for example, a Common Vulnerabilities and Exposures (CVEs) database.”), 8:1-30 (discussing CVE databases and other cyber-security databases). Various types of matching techniques were well known and routinely used. *E.g.*, EX1047 (Bourquin), 1-5, 7 (discussing known binary comparison tools such as BinDiff for identifying malware);³⁰ EX1068 (Epstein), 14:58-15:18

³⁰ EX1047 is a copy of the 2013 conference paper, “Binslayer: Accurate Comparison of Binary Executables,” by Bourquin et al. This copy was obtained from the following URL: <https://kar.kent.ac.uk/37627/1/a4-bourquin.pdf>.

Furthermore, the same conference paper can be found using the internet archive Wayback Machine at the following URL:

<https://web.archive.org/web/20181103234309/https://kar.kent.ac.uk/37627/1/a4-bourquin.pdf>, which was archived on November 3, 2018. The archived copy contains the same title, authors, conference title, conference date, and copyright

(discussing “byte-by-byte comparison”); EX1084 (Czarny), 3:56-4:24 (discussing matching hash of software product with hash known vulnerability), 11:27-47 (discussing performing binary comparison with binary of known vulnerability).

60. Log files were another common target of security scanners. For example, a scanner might look for abnormal activity in a log file to detect anomalies in virtual machine operation. EX1025 (NIST Incident Handling), 29 (“One way to gain this knowledge is through reviewing log entries and security alerts. This may be tedious if filtering is not used to condense the logs to a reasonable size. As handlers become more familiar with the logs and alerts, they should be able to focus on unexplained entries, which are usually more important to investigate. Conducting frequent log reviews should keep the knowledge fresh, and the analyst should be able to notice trends and changes over time. The reviews also give the analyst an indication of the reliability of each source.”).

61. Scanners were also commonly used to identify and classify important assets to help prioritize risks. EX1044 (Giakouminakis), 6:9-18 (“Assets can be

date as shown in EX1047. The archived copy having a cover page is the only difference between the archived version and EX1047.

assigned a classification that will automatically assign specific values that drive the risk associated with vulnerabilities detected on that asset. The classification can describe the importance of the asset (e.g., importance of confidentiality, integrity, or availability) to the entity controlling the asset, the cost to mitigate vulnerabilities, and/or the cost to deal with consequences of an attack on the asset. In embodiments, in determining the risk level, the security tool 102 can be configured to utilize an additional factor that represents the classification of the asset.”); EX1075 (Mohanty), 1:5-22 (“Consequently, in virtualized environments, any risks, threat exposures or known vulnerabilities are constantly changing. A security operations team needs a strong and continuous prioritization system to track critical vulnerabilities and take actions as changes occur.”). Security risks were routinely prioritized based on multiple factors. *E.g.*, EX1075 (Mohanty), 1:23-42 (discussing “common vulnerability scoring system” or “CVSS”), 4:24-5:15 (vulnerability-assessment products “known as scanners scan virtual machines and report vulnerabilities with additional information such as Severity, Base CVSS Score 222, Temporal CVSS Score 222, and CVE ID 220”), 5:16-58 (discussing consideration of “workload context”), 5:59-6:44 (prioritization score based on vulnerability/threat/workload scores and other factors), Figs. 2-4; EX1085 (Hovor), ¶¶48 (“The system may determine priority information for an asset, type of data store on an asset, or both,

using the location of the asset in the network topology.”), 75-77 (asset score based in part on asset scores of other connected assets).

62. Security scans were routinely performed on VMs while target resources were in various states, including while they were inactive (e.g., powered off or suspended) or actively running. EX1065 (Thakar), 2:29-33 (“Scanning the echo for vulnerabilities rather than scanning the actual machine may free up resources at the machine and may allow scanning even if the machine is currently offline, e.g. not connected to a network.”); EX1073 (Ammons Mirage),³¹ 1 (offline image analysis includes image browsing, content introspection, and manipulation capabilities); EX1048 (Price), Abstract (“For at least the virtual machines determined to be offline, a machine image is collected for each offline virtual machine. Security of the offline virtual machines is assessed from the collected images.”), ¶¶14 (the security

³¹EX1073 is a copy of “Virtual machine images as structured data: the Mirage image library,” by G. Ammons et al., published by HotCloud’11: Proceedings of the 3rd USENIX conference on Hot topics in cloud computing on June 14, 2011. This copy was obtained from https://www.usenix.org/legacy/events/hotcloud11/tech/final_files/Ammons.pdf.

assessment of the offline VM may include reading each image file to identify security characteristics or simulating operation of each offline VM based on the corresponding image data), 29, 32, 42, 45, Figs. 3A-C, 4, 6 (illustrating sending the offline VM images to the security server), claim 17. This was done for a variety of reasons. The target VMs themselves might be any of these states already, depending on the circumstances. Not all VMs in production environments would be actively running at any given time, and in fact many VMs would be inactive for extended periods of time. EX1066 (Zhou),³² 2 (“A significant portion of the VM images are mostly offline and infrequently booted.”); EX1067 (Jeswani),³³ 390 (“In a recent

³²EX1066 is a copy of “Always up-to-date: scalable offline patching of vm images in a compute cloud,” by W. Zhou et al., published by ACSAC ’10: Proceedings of the 26th Annual Computer Security Applications Conference on December 6, 2016. This copy was obtained from <https://dl.acm.org/doi/abs/10.1145/1920261.1920317>.

³³EX1067 is a copy of “ImageElves: Rapid and reliable system updates in the cloud,” by D. Jeswani et al., published by the 2013 IEEE 33rd International Conference on Distributed Computing Systems in July 2013 and added to IEEE

study, it was noted that more than 58% of virtual machine images in a university data center had not been used for over 1.5 months.”). For VMs that were already inactive when a security scan was initiated, the assessment process would often be performed without activating the VM, since this was known to cause delays and incur costs. EX1066 (Zhou), 2 (“Note that it takes on the order of minutes just to power up and shut down a VM image. With the large number of dormant VM images that are infrequently used, these approaches add significant extra costs either for customers or for cloud service providers.”).

63. The parameters of the security assessment also impacted the state of the VM during the assessment process in some cases. For example, certain types of scans evaluated the VM’s activity/connections or data stored in the VM’s machine memory, in which case it was reasonable to perform the scan on an active VM. EX1007 (Veselov), 15:60-16:3 (monitoring/mirroring target VM’s activity and connections during assessment in certain embodiments). In other situations—for example, if it were known or suspected that a system was compromised by malicious

Xplore on December 12, 2013. This copy was obtained from
<https://ieeexplore.ieee.org/abstract/document/6681608>.

code, or if a software publisher notified users about a vulnerability discovered in an older version of a program—it was preferable to keep the VM from actively running during the assessment in order to avoid or mitigate damage. *See, e.g.*, EX1081 (Budko), ¶79 (“A detective control may require vulnerability scanning of the VMs at rest to detect vulnerabilities that were recently exposed in the operating system or applications in the VM. A corrective control may require that the VO be quarantined and remediated before being allowed to be activated in a particular environment.”).

3. Responding to Security Risks

64. Regardless of the type of security risk detected, the issue was typically communicated to one or more users, system administrators, or other parties. This was commonly achieved through alerts that would provide information about the vulnerabilities detected by the scanner. EX1025 (NIST Incident Handling), 21 (“Detection of security breaches is thus necessary to alert the organization whenever incidents occur.”).

65. The reporting/alerting process typically involved some sort of prioritization of the threats, which was important for multiple reasons. First, it was understood that prioritization was important as a practical matter given the large number of threats and vulnerabilities that were commonly identified and the different concomitant risk levels. Scans of a given computer system often identified numerous security issues. EX1042 (Rubin), 2:10-27 (“The content scanners of the

present invention are referred to as adaptive rule-based (ARB) scanners. An ARB scanner is able to adapt itself dynamically to scan a specific type of content, such as inter alia JavaScript, VBScript, URI, URL and HTML. ARB scanners differ from prior art scanners that are hard-coded for one particular type of content. In distinction, ARB scanners are data-driven, and can be enabled to scan any specific type of content by providing appropriate rule files, without the need to modify source code.”). This was especially true for cloud computing systems hosted in data centers given the large scale of such systems and their exposed interfaces. EX1045 (Canzanese), ¶7 (discussing enterprise networks using cloud services “Gigabytes of log security and operations log data can be generated in a short time. These logs contain security events with varying levels of threat. Firstly, it is difficult for an analyst to go through these logs and identify the alerts that need immediate attention. Secondly, it is difficult to identify different computer network entities related to a particular alert.”).

66. Moreover, it was well known that different types of threats and vulnerabilities presented significantly different levels of risk. For example, certain threats might have a more significant impact on a given system’s core functionality. EX1025 (NIST Incident Handling), 32 (“Incident handlers should consider how the incident will impact the existing functionality of the affected systems. Incident

handlers should consider not only the current functional impact of the incident, but also the likely future functional impact of the incident if it is not immediately contained.”). Certain types of information targeted by various threats might also be more sensitive than others. *Id.*, 32 (“Incidents may affect the confidentiality, integrity, and availability of the organization’s information. For example, a malicious agent may exfiltrate sensitive information. Incident handlers should consider how this information exfiltration will impact the organization’s overall mission.”). Additionally, the computer system may not be able to recover as quickly or as completely from certain threats, making expeditious resolution of those issues more important. *Id.*, 32 (“The size of the incident and the type of resources it affects will determine the amount of time and resources that must be spent on recovering from that incident. [...] Incident handlers should consider the effort necessary to actually recover from an incident and carefully weigh that against the value the recovery effort will create and any requirements related to incident handling.”).

67. Given the large number of potential issues and their varying importance, prioritizing certain threats/alerts was viewed as an important part of most security assessments. EX1025 (NIST Incident Handling), 32 (“Prioritizing the handling of the incident is perhaps the most critical decision point in the incident handling process.”); EX1046 (Crabtree), 11:52-12:14 (“[A]s cybersecurity and

malware advance, the need for a changing and advancing ranking of threats is obvious and immediate—what was considered a horrible computer virus 15 years ago may be easily detectable today, and similarly, what is considered incredibly high-risk today may be of no consequence to antivirus software 15 years from now, and therefore be categorized as a low threat to most devices.”). For example, it was recognized that prioritizing alerts was preferable to a simple notification process that alerted users about threats in the order they were identified. EX1025 (NIST Incident Handling), 32 (“Incidents should not be handled on a first-come, first-served basis as a result of resource limitations. Instead, handling should be prioritized based on the relevant factors....”); EX1075 (Mohanty), 3:9-33 (security teams “need solutions that help them distinguish the critical vulnerabilities from the noise or false-positives”), 3:33-42 (identifying importance of vulnerability “solves a critical problem in virtualization space, by identifying, correlating, calculating and determining the prioritization of vulnerabilities that pose serious risk to an organization that has operating assets in a virtualized environment”); EX1048 (Price), ¶43 (“[I]nformation can also be displayed pertaining to and detailing the vulnerabilities detected in the example of FIG. 5. For example, a particular issue can be identified 510 together with an indication of the seriousness or priority of the issue.”), Fig. 5.

68. Alerts were commonly prioritized in a variety of ways. *E.g.*, EX1075 (Mohanty), 6:45-9:26 (discussing multi-factor calculation of prioritization score), Fig. 3. For example, some systems prioritized alerts based on the probability that an asset had a particular vulnerability. EX1008 (Basavapatna), Abstract (“A vulnerability detection score is determined that indicates an estimated probability that a particular asset possess the particular vulnerability.”). Another well-known approach was to prioritize by the potential impact or harm that would occur in the event the associated vulnerability was exploited. See EX1044 (Giakouminakis), 11:51-67 (“Prioritization of risk is more than just comparing sums of numbers. Vulnerability A on an asset that is computed to be 10,000 times riskier than another Vulnerability B does not necessarily pose equal risk to the Vulnerability found on 10,000 equivalent assets. ... Risk values are not simple scalars and should be aggregated with risk tolerance in mind. In embodiments, the security tool 102 can be configured to determine a priority of remediation of vulnerabilities based on any combination of the risk level, the overall risk, and/or risk aggregation in relation to risk thresholds.”). It was also well-known that the activity/dormancy of a given application affected the priority of vulnerabilities associated with that application. For example, as Hibbert explains:

But what if no one uses Internet Explorer, and you have standardized on another browser like FireFox or Chrome? Yes, the system is

technically vulnerable but **the offending application is not used and therefore a lower risk** even if the user logs in as an administrator.

EX1063 (Hibbert), ¶80 (emphasis added); *see also id.*, ¶¶100-112 (evaluating factors such as “Application Name” and “Execution Time”). Hibbert goes on to explain that the scanning system may receive information including:

Indication of which applications are being run within the environment that have known vulnerabilities that can put the organization at risk;

Identification of the most frequently run vulnerable applications to help prioritize remediation of these applications through patch deployment or other means; and

Identification of when critical servers or sensitive accounts are utilizing vulnerable applications to help prioritize remediation of these applications through patch deployment or other means.

Id., ¶113.

69. It was also commonplace for cyber security systems to facilitate mitigation of the identified threats and vulnerabilities. *See, e.g.*, EX1025 (NIST Incident Handling), 21 (“In keeping with the severity of the incident, the organization can mitigate the impact of the incident by containing it and ultimately recovering from it. During this phase, activity often cycles back to detection and analysis—for example, to see if additional hosts are infected by malware while eradicating a malware incident. After the incident is adequately handled, the

organization issues a report that details the cause and cost of the incident and the steps the organization should take to prevent future incidents.”); EX1095 (Burdett), ¶¶54-55 (“When a threat or policy violation is detected by the threat management facility 100, the threat management facility 100 may perform or initiate a remedial action facility 128.”).

70. A wide variety of remedial actions were known and commonly used, and the type of remedial action typically depended on the nature of the detected threat or vulnerability. Common remediation techniques included, for example, installing software patches, granting only necessary privileges, and setting up firewalls, and quarantining systems exposes to potential vulnerabilities. EX1025 (NIST Incident Handling), 24 (“All hosts should be hardened appropriately using standard configurations.”); EX1032 (Li), ¶33 (“Policy manager 128 may interpret an entry in system evaluation database 126 that the candidate system (for example, DEF) has a particular firewall installed with given settings. The settings of the particular firewall may be tuned to defeat the particular vulnerability identified in XYZ.exe 202 through its use of ABC.dll.”); EX1046 (Crabtree), 12:15-53 (discussing various remediation techniques such as enabling Address Space Layout Randomization), EX1095 (Burdett), ¶¶54-55 (“Remedial action may take a plurality of forms, such as terminating or modifying an ongoing process or interaction,

sending a warning to a client or administration facility 134 of an ongoing process or interaction, executing a program or application to remediate against a threat or violation, record interactions for subsequent evaluation, or the like.”). Software patches were sometimes applied to actively running VMs, but it was also common for this type of remediation to be performed while the VM was offline, particular since many VMs remained offline for long periods of time. EX1067 (Jeswani), 390 (“[I]t was noted that more than 58% of virtual machine images in a university data center had not been used for over 1.5 months.”). Quarantining was often performed to avoid or mitigate damage to the system or network. EX1070 (Hutchins), Abstract (“Thus, the action in response to detecting the malicious code may include placing the first virtual machine in quarantine”), ¶53 (“If the virus scan finds one or more viruses, the original virtual machine can be quarantined.”), claim 2 (“[W]herein taking action in response to detecting the malicious code includes placing the first virtual machine in quarantine.”); EX1081 (Budko), ¶79 (“A corrective control may require that the VO be quarantined and remediated before being allowed to be activated in a particular environment.”); EX1048 (Price), ¶14 (“The security assessment of the offline virtual machines can include remedying at least one of a security vulnerability or policy violation detected for a particular one of the offline

virtual machines before the particular virtual machine resumes online operation.”), claim 17 (same).

IX. OVERVIEW OF THE '798 PATENT

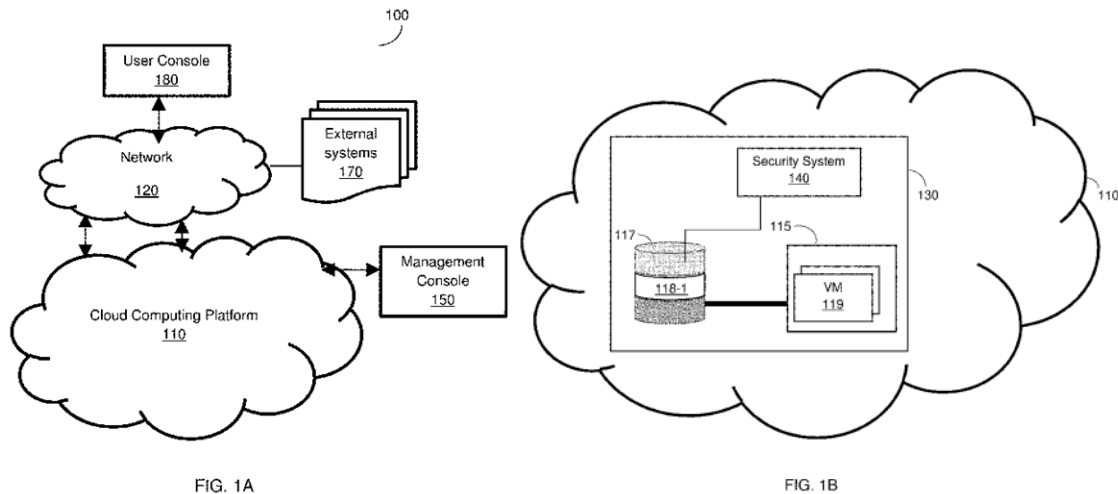
71. The '798 patent is entitled “Techniques For Securing Virtual Machines.” EX1001, Face. The '798 patent discloses techniques “for securing virtual cloud assets in a cloud computing environment against cyber threats.” *Id.*, Abstract. As part of these techniques, the system may first either take a snapshot of a virtual cloud asset or request a snapshot be taken if one does not exist. *Id.*, 5:1-8. Moreover, a snapshot may also be taken on “predefined schedule or upon predefined events (e.g., a network event or abnormal event).” *Id.*, 5:8-11. Such events may include “changes to infrastructure such as instantiation of an additional container on existing VM, image change on a VM, new VM created, unexpected shutdowns, access requests from unauthorized users, and the like.” *Id.*, 7:19-25.

72. Once the snapshot has been taken, the snapshot may be “accessed, or otherwise copied” and “parsed” so that it may be “analyzed to detect cyber threats and potential vulnerabilities.” *Id.*, 5:19-24 (“The snapshot is parsed and analyzed by the security system 140 to detect vulnerabilities.”), 7:32-38. The analysis may include checking the “configuration files of the applications and operating system of the VM” and “match installed application lists, with their respective versions, to a known list of vulnerable applications” by, for example, “computing a cryptographic

hash against database of files in vulnerable applications.” *Id.*, 5:32-53. The security system may also read process identification number (PID) files and then use them to deduce “which processes are running, and hence the priority of vulnerabilities detected in processes existing on the disk.” *Id.*, 6:12-18.

73. Next, the ’798 patent discloses that “the detected cyber threats and/or vulnerabilities are reported,” for example, as filtered and prioritized alerts. *Id.*, 7:50-52. For example, the detected cyber threats may be “reported to a user console 180 and/or a security information and event management (SIEM) system” after being “filtered or prioritized based in part on their determined risk” to reduce “the number of alerts reported to the user.” *Id.*, 6:38-45. In addition to reporting the cyber threats, the security system may “cause an execution of one or more mitigation actions,” such as “blocking traffic from untrusted networks, halting the operation of the VM, quarantining an infected VM, and the like.” *Id.*, 6:59-63.

74. Figures 1A and 1B of the ’798 patent show network diagrams used to describe the various embodiments.



Id., Figs. 1A-B. “The arrangement of the example cloud computing platform 110 is shown in FIG. 1B. As illustrated, the platform 110 includes a server 115 and a storage 117, serving as the storage space for the server 115. The server 115 is a physical device hosting at least one virtual machine (VM) 119. The VM 119 is a protected VM, which may be any virtual cloud asset including, but not limited to, a software container, a micro-service, a serverless function, and the like.” *Id.*, 3:44-51. The specification also describes using commercially available platforms. *Id.*, 3:64-67 (“In some example embodiment, the client environment 130 may be part of a virtualized environment or on-premises virtualization environment, such as a VMware® based solution.”).

A. Challenged Claims

75. The '798 patent has 34 claims, with claims 1 and 18 being independent. I understand that each of claims 1-34 is being challenged in the *inter partes* review proceeding related to this declaration.

76. Below, I reproduce the text of independent claim 1 with numbering added to certain claim elements for ease of reference.

A method for securing virtual cloud assets in a cloud computing environment against cyber threats, comprising:

[1.1] taking a snapshot of at least one virtual disk of a protected virtual cloud asset, wherein the virtual cloud asset is instantiated in the cloud computing environment;

[1.2] accessing the snapshot of the at least one virtual disk;

[1.3] analyzing the snapshot of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset; and

[1.4] alerting detected potential cyber threats based on filtering or prioritizing using a determined risk of each detected threat.

77. Furthermore, independent claim 18 is directed to a system that essentially mirrors claim 1. The dependent claims add other conventional aspects of cybersecurity and cloud computing, including prioritizing threats based on a risk to an asset (claims 2 and 19), parsing/scanning a copy of a snapshot to detect threats that are known and unknown vulnerabilities (claims 3 and 20), checking

configuration files/installed OS, verifying access times, analyzing system logs to deduce executed applications/modules, or analyzing memory in a snapshot to determine executed applications/modules (claims 4 and 21), instantiating a copy of a VM from a snapshot and monitoring activity of the instance (claims 5 and 22), reading process identification number files and checking access times (claims 6 and 23), matching a list of installed applications with a list of vulnerable applications, matching application file against a database of vulnerable applications, matching application sub-modules against a database of sub-modules, or reading packet manager installation logs (claims 7 and 24), using VMs, software containers, or a micro-services as a cloud asset (claims 8 and 25), taking a snapshot based on a schedule (claims 9 and 26), taking a snapshot based on a request (claims 10 and 27), taking a snapshot based on network/abnormal events (claims 11 and 28), mitigating a threat (claims 12 and 29), detecting threats that do not represent vulnerabilities (claims 13 and 30), detecting vulnerable applications by comparing a hash of an application with a database of known applications (claims 14 and 31), looking for threats inside containers (claims 15 and 32), looking for personal identifiable information or private keys (claims 16 and 33), using a server or storage system as a cloud asset (claims 17 and 34).

B. Prosecution History of the '798 Patent

78. I understand that the '345 application (i.e., the patent application that led to the '798 patent) never received a rejection under §102 or §103. I understand that the first office action rejected the claims based on nonstatutory double patenting over the parent application, which is now U.S. Patent No. 11,431,735 (the "'735 patent") along with five other related patents. EX1004, 135-37. I understand that Applicant filed six terminal disclaimers to secure allowance over the related patents, including the '735 patent which has now had all but two dependent claims disclaimed. *Id.*, 104-06, 110-17, 122-23; *Wiz, Inc. v. Orca Security Ltd.*, IPR2024-00220, Paper 7 (May. 09, 2024), 3. I understand that in the notice of allowance the Examiner identified three references as the closest prior art, but only broadly indicated the supposedly lacking teachings by listing most of the independent claim language without explanation. EX1004, 59-60.

X. CLAIM CONSTRUCTION

79. I understand that, in this proceeding, the claims are construed according to their ordinary and customary meaning, in view of the specification and prosecution history, to a POSA as of the effective filing date of the underlying patent application. I understand that this is the same claim construction approach used in district court litigation of related patents.

A. “[Analyzing/Analyze] the Snapshot”

80. Each independent claim recites “analyzing the snapshot” (or a system configured to “analyze the snapshot”).

81. In my opinion, a POSA would have understood the ordinary and customary meaning of “[analyze/analyzing] the snapshot” as at least encompassing direct analysis of the snapshot data (e.g., analyzing the snapshot as a data file without instantiating an assessment VM). This understanding is consistent with the ’798 specification. *See, e.g.*, EX1001, 5:19-20 (“The snapshot is parsed and analyzed by the security system 140 to detect vulnerabilities.”), 5:36-39 (“[T]he security system 140 may be configured to match the application files, either directly (using binary comparison) or by computing a cryptographic hash against database of files in vulnerable applications.”), 6:4-11 (“[T]he security system 140 is configured to analyze the machine memory, as reflected in the page file.”), 6:34-37 (security system “computing a cryptographic hash of the sensitive areas in the virtual disk and checking for differences over time”), 6:54-58 (system can analyze logs “derived from the snapshot”).

82. In addition, I have been asked to assume that “[analyze/analyzing] the snapshot” also encompasses analysis of a VM instantiated from the snapshot.

83. Accordingly, my discussion of Ground 1 below applies an understanding of “analyzing the snapshot” in claim 1, as well as “analyze the snapshot” in claim 18,

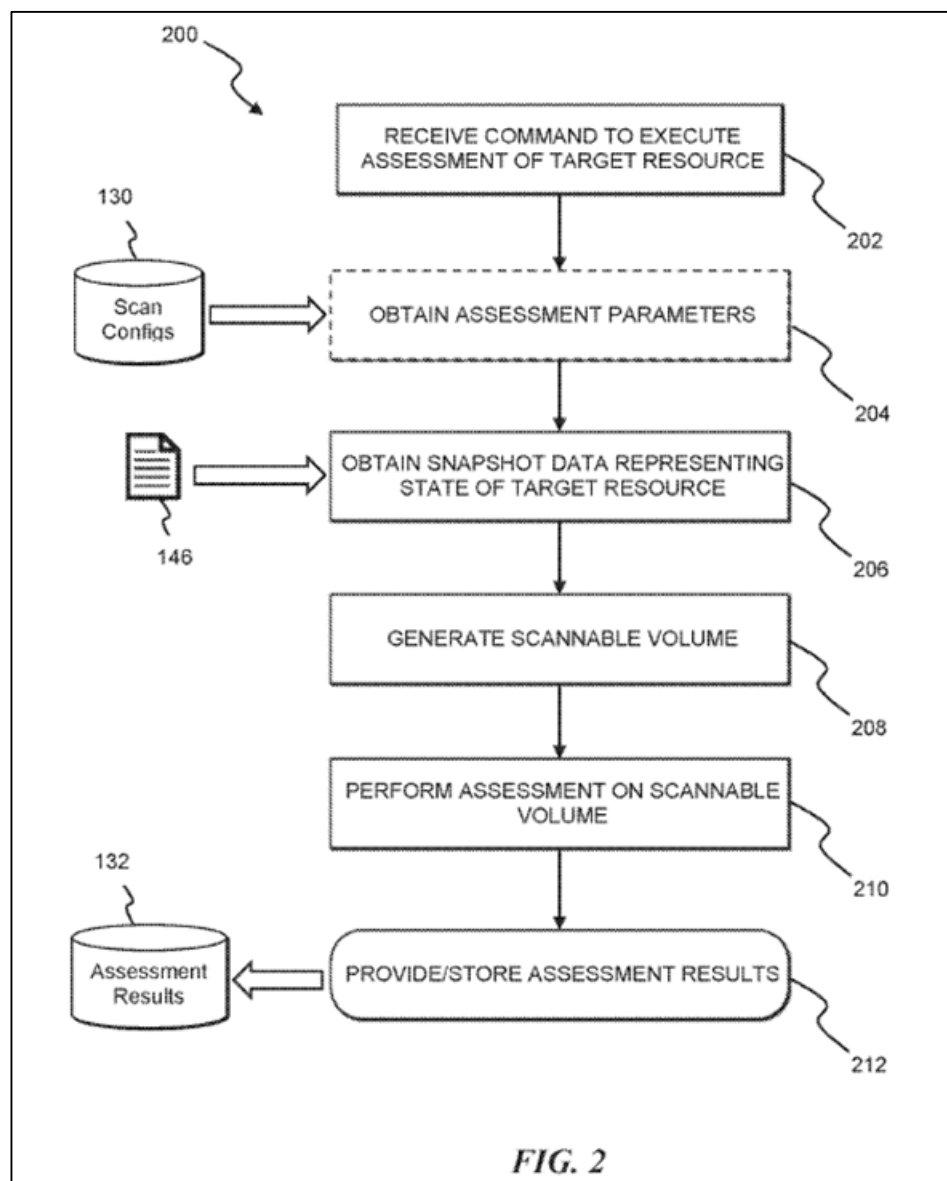
as encompassing both direct analysis of the snapshot data and analysis of a VM instantiated from a snapshot. *Infra*, §XII.C.5.

XI. OVERVIEW OF THE PRIOR ART

A. Veselov (U.S. Patent No. 11,216,563, EX1007)

84. Veselov describes systems for “performing a security assessment of a target computing resource, such as a virtual machine,” and teaches scanning techniques mirroring those of the ’798 patent. EX1007 (Veselov), Abstract.

85. Veselov describes systems for “performing a security assessment of a target computing resource, such as a virtual machine,” and teaches scanning techniques mirroring those of the ’798 patent. *Id.*, Abstract. The scanning service generates or requests a snapshot (or obtains access to the snapshot at its current location), scans the snapshot, and provides assessment results to a user. *Id.*, Abstract, 2:35-41, 3:20-4:18, 5:9-22, 8:16-51, 9:9-10:36, 11:5-33, 16:47-53. A high-level overview of an exemplary process is depicted below:

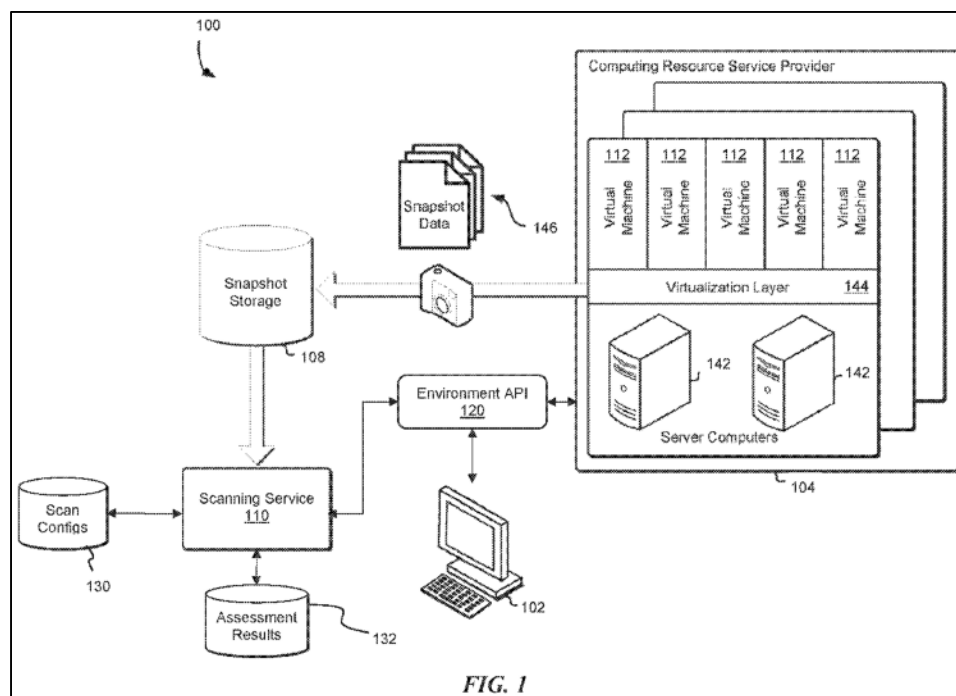


Id., Fig. 2; *see also id.*, 8:52-10:36 (describing Fig. 2), 16:15-17:9 (describing Fig. 4), 18:16-56 (describing Fig. 6), Figs. 4, 6.

86. The scanning service can directly analyze the snapshot data and/or analyze an assessment VM instantiated from the snapshot. *E.g.*, *id.*, Figs. 3A-B (scanning service analyzes assessment VM), 5A-B (scanning service analyzes

snapshot as data file). Both types of assessments identify a wide variety of security risks. *Id.*, 1:41-49 (known security assessment services identify “vulnerabilities, bad configurations, and the like”), 2:35-41 (“The present disclosure contemplates implementation of any suitable security assessment, including” static/dynamic analysis and analysis of CVEs, benchmarks, best practices, configurations, etc.).

87. Veselov notes that its techniques are particularly useful in cloud computing environments such as data centers. *Id.*, 2:22-3:19. It also applies its techniques to distributed, virtualized systems commonly used for cloud computing:



Id., Fig. 1; see also *id.*, 1:5-50 (discussing virtualization in data centers); *supra*, §§VIII.A, VIII.C (discussing cloud computing and virtualization). The scan can be performed in the target resource’s computing environment and in a test environment.

E.g., EX1007 (Veslov), Figs. 3A, 5A (analysis in test environment), 3B, 5B (analysis in target's environment).

88. Veslov describes both agent-based and agentless scans. In an agentless scan, for example, the system obtains (or obtains access to) a snapshot, which were routine techniques at the time that typically involved locating the snapshot and either bringing a snapshot into a test environment using that location or accessing the snapshot at its location in the target resource's host environment. *Id.*, 4:53-54 ("In some embodiments, the scanning service 110 may obtain and analyze snapshot data 146."), 6:38-43 ("The scanning service 110 may determine the corresponding virtualization layer 144 for the virtual machine 112 identified by the request and transmit a command to the virtualization layer 144 to provide snapshot data 146 (or to provide access to snapshot data 146) corresponding to the virtual machine 112."), 9:17-25 ("In various non-limiting examples, the scanning service may obtain the snapshot data by executing the snapshot itself, by requesting another service to capture the snapshot and provide it to the scanning service, by accessing a data store (e.g., the snapshot storage service) containing the snapshot, etc.; alternatively, the scanning system may obtain access to the snapshot data, such as at a storage location of the snapshot data, but may not obtain (i.e., make a copy

of) the snapshot data.”). Veselov further describes preparing the snapshot for scanning, performing the scan, and providing assessment results to a user:

89. Veselov describes a variety of analysis techniques. For example, it describes analysis using an agent installed on the analyzed resource (e.g., instantiating an assessment VM from the snapshot and installing an agent on that VM to monitor activity). *See, e.g., id.*, 14:22-17 (agent example), 16:45-53 (same), claim 2 (same), Fig. 3B (depicting security agent 330). Veselov also describes analysis techniques that do not rely on such agents, including agentless analysis of the assessment VM and analysis of the snapshot itself as a data file (without instantiating an assessment VM). *See, e.g., id.*, 3:61-67 (agentless analysis of snapshot data), 16:67-17:3 (“In the implementations of FIGS. 5A-B, rather than recreating and instantiating a duplicate VM, the scanning service 510 is configured to analyze the volume image 508 as a data file.”), Figs. 3A (depicting assessment VM without agent), 5A-B (agentless analysis of snapshot as data file); *see also id.*, 14:13-22 (appropriately authorized scanning service “may perform any suitable security assessment” on assessment VM and its underlying resources); *supra*, §VIII.D.2 (agent-based and agentless security assessments both well-known).

90. Following analysis, “if vulnerabilities are identified in the assessment results,” Veselov’s methods “compar[e] the assessment results to a remediation

framework to identify one or more actions the user can take to address the vulnerabilities, and providing the identified actions to the user; and the like.” *Id.*, 10:32-36.

91. Veselov describes several different embodiments and figures, but a POSA would have understood that the techniques and components described in the context of a particular embodiment/figure can also applied in the context of the other embodiments and figures. For example, discussing Figure 1, Veselov states that “embodiments of the present disclosure may operate within or upon a computing environment 100.” *Id.*, 4:19-29. Veselov similarly explains that the techniques discussed in the context of Figure 2 can be performed by the various systems described in other embodiments. *Id.*, 8:52-62 (“The process 200 may be performed by any suitable system such as a scanning service or component thereof as described above in connection with FIG. 1.”). Likewise, Veselov explains that the scanning service described in conjunction with Figures 3A-B may perform security assessments “in accordance with various embodiments,” and that Figures 5A-B illustrate “additional embodiments of the present security assessment systems” and that the depicted components “as well as their implementations and interactions, are the same as described above” for associated components of Figs. 3A-B. *Id.*, 10:37-41, 16:56-67. Veselov also explains that Figure 4 depicts security-assessment

techniques “in a computing environment like the example computing environments 300, 350 of FIGS. 3A-B” and that “FIG. 6 illustrates a method 600 of performing a security assessment in a computing environment such as the environments 500, 550 of FIGS. 5A-B.”). *Id.*, 16:15-18, 18:16-18. The teachings associated with Figure 7 similarly apply to other disclosed embodiments. *Id.*, 18:56-59 (“FIG. 7 illustrates aspects of an example environment 700 for implementing aspects in accordance with various embodiments.”). In other words, Veselov makes it clear that when a portion of the disclosure associated with a particular figure/embodiment describes certain techniques or components, those teachings are broadly applicable to the rest of Veselov’s disclosure. *See id.*, 24:54-25:2 (noting that phrases such as “in an embodiment” are “not necessarily intended to refer to the same embodiment or to any one particular embodiment described.”).

B. Basavapatna (U.S. Pub. No. 2013/0191919, EX1008)

92. Basavapatna is a publication of a patent application that relates to a “computer security assessment and, more particularly, to calculating risk metrics for assets in a system of computing assets.” EX1008 (Basavapatna), ¶1. The assessment involves scanning assets to detect, evaluate, and remediate various types of security problems. *Id.*, ¶¶20-25, 29-30, 59, Fig. 1. Basavapatna detects vulnerabilities and associated threats based on various types of data. *Id.*, ¶¶26-28, 33 (discussing threats and vulnerabilities). To detect these potential security problems, the system

processes (among other things) “asset configuration data,” “threat definition data,” “vulnerability definition data,” and “vulnerability detection data.” *Id.*, ¶34; *see also id.*, ¶¶70-71 (normalizing and reconciling data from different sources). Asset configuration data describes an assets’ hardware and software configurations, which informs the types of vulnerabilities and threats that are applicable to a given asset. *Id.*, ¶¶35-37. Threat definition data is obtained from threat information services (or “threat feeds”) and describes various threats, their severity, and the vulnerabilities they target. *Id.*, ¶¶38-49. Vulnerability definition data is received from vulnerability information services and describes various vulnerabilities, their severity, and associated threats. *Id.*, ¶¶34, 50-56. Vulnerability detection data allows the system to determine “that an asset has one or more vulnerabilities.” *Id.*, ¶¶57-62 (“Vulnerability detection data can also indicate the probability that a vulnerability is possessed by an asset.”). It is based on various tests such as “virus scans, vulnerability analyses, system configuration checks, policy compliance checks, network traffic checks (e.g., provided by a firewall, a network intrusion detection system, or a network intrusion prevention system), and tests performed by host-based intrusion prevention systems or host-based intrusion detection systems.” *Id.*, ¶59.

93. The system determines various types of risks associated with the detected vulnerabilities, including vulnerability-centric risks metric and threat-centric risk metrics. *Id.*, ¶¶72 (risk metrics assess risk created “by virtue of the asset’s exposure to a particular vulnerability and thereby also, in some instances, an associated threat known to exploit the particular vulnerability”), 114-15; *see also id.*, ¶¶84, 105 (risk metrics use vulnerability detection data). Basavapatna notes that some administrators were comfortable with assessments that focus on the threats that exploit the vulnerabilities (i.e., a threat-centric risk analysis) while others may find a vulnerability-centric risk assessment more useful. *Id.*, ¶¶72, 100. Accordingly, risk can be calculated “using either or both of threat-centric and vulnerability-centric techniques.” *Id.*, ¶72.

94. The vulnerability-centric risk metric is based on a composite vulnerability score and a countermeasure score. *Id.*, ¶112, Fig. 3B. The composite vulnerability score estimates “the potential risk a particular vulnerability poses to an asset” and is calculated based on a standardized vulnerability score (e.g., CVSS scores) and a vulnerability detection score (which indicates the likelihood that the asset has a particular vulnerability). *Id.*, ¶¶104-06.

95. The threat-centric risk metric is based on an exposure factor (which estimates the “risk that an asset will be affected by the threat”) and a criticality score

(which “represents an impact of losing an asset”). *Id.*, ¶¶82, 92-97, Fig. 3A. The exposure factor is derived from a threat factor, a vulnerability score (which “estimates whether the asset possesses a particular vulnerability”), and a countermeasure score. *Id.*, ¶¶82-85. The threat factor is derived from a threat severity score and an applicability score. *Id.*, ¶¶75-77; *see also id.*, ¶¶45 (severity score “is an estimate of how severe an attack by the threat would be for an asset, and may optionally also estimate how likely the threat is to affect assets”), 78-79 (applicability score may indicate “asset is running software that is of a type that can have a vulnerability that is exploited by the threat”).

96. Basavapatna also describes calculating various types of aggregate risk metrics. Risk metrics can be aggregated for vulnerabilities/threats (e.g., an aggregate of vulnerability-centric risk metrics for a particular vulnerability across multiple assets), for an asset as a whole (e.g., an aggregate of vulnerability-centric or threat-centric risk metrics for all vulnerabilities/threats applicable to the asset), or for a group of assets. *Id.*, ¶120-37.

97. Basavapatna teaches robust reporting, prioritization, and mitigation of detected security issues. After determining the risk metrics discussed above, the system may report vulnerabilities, threats, and assets along with their respective

risks. *Id.*, Abstract, ¶¶138-40. The results can be filtered and prioritized in various ways:

For example, the network monitor 102 can list all assets, vulnerabilities, or threats, sorted by aggregate risk metric, or can list a top number, e.g., top ten, of the assets, vulnerabilities, or threats. Ranking assets, vulnerabilities, and/or threats according to aggregate risk scores allows a user to quickly identify which assets are most at risk, or which vulnerabilities and threats are most dangerous for a system. The user can then remediate the most at-risk assets before remediating other less-at-risk assets, or can apply remediations across the system for the riskiest vulnerabilities and threats before applying remediations for other, less-risky ones.

Id., ¶138; *see also id.*, ¶¶139-40 (discussing examples of display and filtering), Fig. 6. The system can apply various types of mitigation techniques, including active countermeasures (e.g., “turning off vulnerable machines”) and passive countermeasures (e.g., “firewalls, host intrusion prevention systems, network intrusion prevention systems, network access control systems, intrusion detection systems”). *Id.*, ¶¶22-29.

C. Kapoor (US Patent No. 10,498,845, EX1111)

98. Kapoor is a publication of a patent application that relates to collecting “data associated with a process” to report to an external node “to detect anomalies in a network environment.” EX1111 (Kapoor), Abstract.

99. Kapoor teaches a cyber security method to address difficult to detect anomalies in virtualized cloud environments. *Id.*, 1:18-24 (computer resources are increasingly being virtualized and “such virtualization can also make it more difficult to detect and mitigate intruders (and other nefarious individuals)”). These cloud environments include various types of datacenters. *Id.*, 4:27-34 (“A datacenter can also comprise cloud-based resources, such as infrastructure as a service (IaaS), platform as a service (PaaS), and/or software as a service (SaaS) elements.”). Moreover, these datacenters comprise many nodes, which may include VMs. *Id.*, 13:30-54; *see also id.*, 14:20-44 (discussing processes running on VMs), 15:54-67 (discussing the use of VMs in datacenters). The security method models the activity of the datacenters to detect anomalous behavior. *Id.*, 4:15-21 (“Using techniques described herein, a baseline of datacenter activity can be modeled, and deviations from that baseline can be identified as anomalous.”).

100. As part of detecting anomalous behavior, Kapoor teaches collecting and transmitting information regarding one of the datacenter nodes, including the processes running on it. *Id.*, 6:41-7:67, 8:31-48. To determine what processes are running on a node, Kapoor teaches searching the PID file directory. *Id.*, 7:21-23 (“One way an agent can obtain a mapping between a given inode and a process identifier is to scan within the /proc/pid directory.”), 7:63-67 (tracking the “newest

process that it has reported information on (e.g., to data aggregator 114), and begin its search of /proc/ in a forward order starting from the PID of that process.”). Kapoor explains that the operating systems on the nodes use PIDs to “identify processes running at a given time.” *Id.*, 37:35-37. Once the node information is collected, which includes the node PIDs, it is then transmitted “to a data aggregator (e.g., data aggregator 114), which in turn provides it to platform 102.” *Id.*, 8:41-43, 8:64-65 (PID information is transmitted to platform), Fig. 3A (depicting the same).

101. Finally, Kapoor teaches that the platform performs the “process for detecting anomalies in a network environment.” *Id.*, 23:37-39, Fig. 18. The platform detects anomalies by analyzing how processes behave over time. For example, it may present a graph that “provide[s] information such as a list of all active processes and PIDs in the last hour, the number of historic total processes, [and] the average number of active processes per hour.” *Id.*, 24:8-21.

D. Czarny (U.S. Patent No. 9,749,349; EX1084)

102. Czarny is a publication of a patent application that describes identifying vulnerabilities by matching the scanned system’s software against a database of known vulnerable applications. EX1084 (Czarny), Abstract (“Computer security vulnerability assessment is performed with product binary data and product vulnerability data that correspond with product identification data.”).

103. Czarny discusses several types of databases that can be used for this matching analysis. *Id.*, 3:37-4:24. Czarny describes these databases as part of a database system that, in general, associates “binary data (related to software products, or specific versions of the software products, i.e. ‘product binary data’) with known security vulnerabilities (of the same software products, or specific versions thereof, i.e. ‘product vulnerability data’).” *Id.*, 3:56-62. Exemplary databases in this database system include a “binaries-to-vulnerabilities mapping database,” a “binaries-to-products mapping database,” a “products-to-vulnerabilities mapping database,” as well an “intermediate” database that combines information from other databases. *Id.*, 5:15-43, 8:17-55. For example, examples of the binaries-to-products mapping database and the binaries-to-vulnerabilities databases are shown below:

Fig. 2

106

Product	Version	Binary Files
ESET Endpoint Security	5.0.2214.4	Exe1_sha256, Exe2_sha256, Dll3_sha256, ...
ESET Endpoint Security	5.0.1055.2	Exe1_sha256, Exe4_sha256, Dll5_sha256, ...
ESET Endpoint Security	4.2.3330.1	Exe1_sha256, Exe2_sha256, Dll5_sha256, ...
ESET Endpoint Security	4.0.1211.2	Dll3_sha256, Dll5_sha256, Exe1_sha256, ...
JAVA	7.11	Dll9_sha256, Dll12_sha256, Dll13_sha256, ...
...	...	...

Fig. 5

105

Binary File	Known Vulnerabilities
Exe1_sha256	Vulner_1, Vulner_2, ...
Exe2_sha256	Vulner_4, ...
Exe4_sha256	Vulner_5, ...
Dll3_sha256	Vulner_3, ...
Dll5_sha256	Vulner_6, ...
...	...

Id., Figs. 2, 5. The binary data included in the databases may be obtained from user devices in the system, or it can be obtained from “an existing file information service or from copies of the files of the software products.” *Id.*, 5:53-6:24. The product data and vulnerability data included in the databases may be obtained from “from

various products and vulnerabilities information sources” such as CVE repositories. *Id.*, 6:25-38 (“There are a variety of sources 109 for known product vulnerability and severity information, including CVE (Common Vulnerabilities and Exposures), NVD (National Vulnerability Database) and OSVDB (Open Sourced Vulnerability Database)”).

104. Czarny describes several types of data that may be matched against the database to identify vulnerable applications, including complete binary-level application files and hashes of applications files. *Id.*, 4:7-15. For example, Czarny explains:

The binary data (for the product binary data or the target binary data) generally contains 1) **binary hashes of binary level files of the software products**, 2) binary hashes of strings of bits, bytes, words or characters extracted from the files of the software products, 3) the unprocessed strings of bits, bytes, words or characters that were extracted, 4) **the complete binary level files of the software products**, or 5) **any other appropriate binary-level representation of the software products**.

Id. (emphasis added). The matching process may be done at the individual bit, byte, or word level and may be performed using “string, binary, or file matching or comparing type of function.” *Id.*, 4:16-47; *see also id.*, 5:44-52, 9:17-10:42, 11:16-12:10, Figs. 1-5, 7-8, 10.

105. Czarny contrasts its matching technique with other matching processes. Czarny discusses conventional matching techniques that identify vulnerable software applications using application-identifying information such as the name/version of the application. *Id.*, 4:25-33. With the disclosed binary/hash-matching techniques, however, the “scanning and matching searches for a match between two files or two strings within two files at the low level of binary data, rather than matching a file name or other higher level meta data of two files.” *Id.*, 4:15-24. Czarny states that matching by application name/version is “less thorough, robust or flexible than the present system, because it could potentially miss some known vulnerabilities, since the actual binary level data in the files of the software product could be different from the official version of the software product.” *Id.*, 4:25-42. Czarny therefore explains that its own binary/hash-matching techniques “can be used in place of a conventional security vulnerability assessment system, or in combination therewith, to enable a more thorough, robust and flexible level of functionality.” *Id.*, 4:42-47.

106. Czarny also discloses determining and reporting a risk level for the asset. “Using any matches (results of the scanning) and the product vulnerability data of the offline vulnerability database of 110, the validation server 102 consolidates the information and generates a target device vulnerability report” that

can include binary data, file paths, and the known vulnerabilities. *Id.*, 7:31-45; *see also id.*, 12:11-18, Figs. 6, 10. The report can also include a “security vulnerability level” for the device. *Id.*, 7:46-67. This security vulnerability level “may be provided as a score value indicating a level of vulnerability of the target device 103.” *Id.*, 7:50-61 (scoring examples include zero/low/high or “pass/fail”). With this report, security administrators or access-control applications “can analyze the security vulnerability level of the target device 103 to determine whether to grant or deny access by the target device 103 to the secure environment 104.” *Id.*, 7:46-67.

E. Chari (U.S. Patent Pub. 2015/0033221; EX1056)

107. Chari describes security analysis of VM images/snapshots in cloud environments that involves detecting sensitive data. EX1056 (Chari), Abstract, ¶¶42, 52-62 (discussing cloud environment), 94 (snapshots), Figs. 1-2, 9, 10A-B; *see also id.*, ¶59 (leakage of sensitive data is a concern in cloud computing environments). Chari notes that VMs may contain sensitive data such as “passwords, credit card numbers, social security numbers, bank account numbers, cryptographic keys, user-generated data, user history data, and paths to locations of the sensitive data.” *Id.*, ¶38; *see also id.*, ¶¶30, 42 (“In other words, [a] specific instance of a virtual machine image 226 is associated with one particular user and may contain sensitive data that the particular user does not want to be accessible by other users.”), 61 (“Virtual machine images may contain substantial amounts of

sensitive data” such as “raw information in virtual disk blocks; sensitive details in system configuration files including network topology information and administrator accounts and passwords; sensitive details in system logs, such as administrator passwords; application data, such as raw data in database files; and application credentials, such as default passwords for applications”). To avoid unwanted exposure of such data, Chari analyzes the image to “identify and label sensitive data contained with a virtual machine” based on defined policies. *Id.*, ¶39; *see also id.*, ¶¶64-66, Figs. 5-7. Scripts discover the “presence or absence of particular sensitive data items within, for example, data caches or system logs” and generate “an appropriate sensitivity level label for each identified sensitive data item.” *Id.*, ¶¶76-80. Sensitivity levels are defined by policies. *Id.*, ¶77 (“For example, labeling policies 706 may define a password data item as having a high sensitivity level, whereas a system log data item may have a medium sensitivity level.”). The image may later be sanitized while the VM is online or offline to, for example, remove or encrypt certain sensitive data. *Id.*, ¶80, 83-87, Fig. 8. This process can be performed on different types of virtual resources, including instantiated VMs, disk images, or snapshots. *Id.*, ¶61 (sensitive data may be found at all layers of a cloud environment, including “raw information in virtual disk blocks; sensitive details in system configuration files including network topology

information and administrator accounts and passwords; sensitive details in system logs, such as administrator passwords; application data, such as raw data in database files; and application credentials, such as default passwords for applications”), ¶¶92-95 (“In addition, the virtual machine image manager labels and sanitizes the snapshot of the specific instance of the virtual machine image....”), Fig. 9, claim 9.

F. Roth (U.S. Patent No. 9,524,389; EX1116)

108. Roth describes techniques for triggering the generation of snapshots to facilitate security analysis of VMs. EX1116 (Roth), Title, Abstract (“A system and method for capturing forensic snapshots of virtual machines....”). Roth discusses the widespread use of virtual assets hosted by remote computing infrastructure and the security concerns that arise in such contexts, including the introduction of a “nonstandard configuration” and “installation of unauthorized executable code” that may compromise security. *Id.*, 1:12-36; *see also id.*, 2:47-62 (discussing virtual assets and infrastructure), 6:16-48 (describing exemplary environment), Figs. 1-4 (depicting physical and virtual resources). Roth also notes the importance of capturing the state of such systems for security-analysis purposes. *Id.*, 1:36-44. To facilitate that benefit, Roth discusses various techniques for triggering the taking of snapshots to capture VM state. *Id.*, Abstract, 2:20-47, 2:62-3:16, 7:1-21; *see also id.*, 3:17-4:14 (discussing initial snapshot and subsequent snapshot), 4:32-36 (“Described and suggested techniques improve the field of computing, specifically

the field of digital forensics, by capturing before and after snapshots of a computing system which can be preserved for forensic investigation at a later date.”), 4:46-5:15 (discussing VM and contents of base snapshot), claim 1. Roth explains that the triggering events may be configured by a user and can include things such as the VM reaching the end of its lifecycle, the VM reaching various work-related thresholds, detection of various errors, detection of unauthorized configurations, and **detection of an intrusion/security compromise of the VM**. *Id.*, 2:67-3:16, 3:50-67, 7:59-8:29, 17:35-43; *see also id.*, 6:58-67 (discussing VM termination in response to various events).

XII. GROUND 1: CLAIMS 1-5, 7-15, 17-22, 24-32, AND 34 WERE OBVIOUS OVER VESELOV AND BASAVAPATNA.

109. As I explain below, it is my opinion that claims 1-5, 7-15, 17-22, 24-32, and 34 were obvious over Veselov and Basavapatna. In the next few paragraphs, I summarize the reasons for this opinion at a high level, and in the sections that follow, I explain those reasons in more detail.

110. Veselov describes a security assessment performed on a snapshot of a virtual cloud asset such as a VM. As part of this assessment, a scanning service takes a snapshot and then obtains access to that snapshot. *Supra*, §XI.A (Veselov summary). It then performs various types of analyses on the snapshot data of a virtual disk to determine the existence of cyber threats—e.g., performing

configuration/CVE assessments—before reporting the assessment results. *Supra*, §XI.A (Veselov summary).

111. Basavapatna further discusses a set of scores representing the risk the various detected cyber threats pose to the system, countermeasures or remediation techniques to alleviate the risk posed, and alerts based on the determined risk. *Supra*, §XI.B (Basavapatna summary).

112. A POSA had reason to combine these teachings to arrive at the claims as a whole. *Infra*, §XII.A (reasons to combine). For example, a POSA had reason to perform snapshot-based security assessments, as taught by Veselov, while further applying Basavapatna’s teachings involving alerting based on a determined priority. *Infra*, §XII.A.

A. Reasons to Combine Veselov and Basavapatna

113. In my opinion, a POSA would have had reason to combine the teachings of Veselov and Basavapatna to arrive at claims 1-5, 7-15, 17-22, 24-32, and 34.

114. As an initial matter, Veselov and Basavapatna are analogous art to one another and to the ’798 patent. Veselov describes cyber security techniques involving analysis of computer resources to identify threats and provide alerts associated with the identified threats. *See supra*, §XI.A (Veselov summary); EX1007 (Veselov), Abstract, 4:5-8 (“The scanning system may store the assessment

results, or provide them to an owner or administrator of the virtual machine in order to evaluate the vulnerabilities of the target resources.”), 8:46-50 (“The scanning service 110 may perform the security assessment and obtain assessment results from the performance. The assessment results may be associated with the instance of the virtual machine 112 that was embodied in the snapshot data 146, and stored in a results data store 132 and/or delivered to the user (e.g., via the environment API 120). Basavapatna similarly describes cyber security techniques involving identifying vulnerabilities, assessing risk, and reporting assessment results. *See supra*, §XI.B (Basavapatna summary); EX1008 (Basavapatna), Abstract, ¶¶1 (“This disclosure relates in general to the field of computer security assessment and, more particularly, to calculating risk metrics for assets in a system of computing assets.”), 12. The references are therefore in the same field of endeavor as the ’798 patent and are pertinent to issues addressed in the ’798 patent. *See, e.g.*, EX1001, 1:12-14 (“This disclosure relates generally to cyber-security systems and, more specifically, to techniques for securing virtual machines.”).

115. Furthermore, securing cloud systems by receiving scan requests, locating/accessing/analyzing snapshots of virtual disks to identify vulnerabilities, reporting results, and mitigating detected security problems, as taught by Veselov, were well-known and predictable techniques. *Supra*, §§VIII.B (snapshots),

VIII.D.2-3 (security analysis and response); EX1014 (Klemperer), 24-25 (taking a snapshot of a VM); EX1034 (Tang), 5643 (matching vulnerabilities using signature database); EX1040 (Ammons), Abstract, ¶43, claim 6 (image/snapshot-based analysis); EX1076 (Milner), 2:24-67 (matching software packages to database of known vulnerabilities); EX1077 (Khalid), 2:45-51 (comparing characteristics of suspicious objects to CVE database), 8:1-30 (same); EX1078 (Hufsmith), Abstract, ¶¶8, 32-34, 104-05, 149 (analysis of container/VM images includes CVE/CWE/malware matching). The applied teachings of Basavapatna—including determining risks associated with potential cyber threats, determining application usage, reporting prioritized alerts, and mitigating detected problems—were similarly well-known and predictable. *Supra*, §§VIII.D.2-3 (security analysis and prioritized alerts), XI.B (Basavapatna summary); EX1025 (NIST Incident Handling), 32 (prioritizing incidents based on impact of systems and information); EX1036 (Wysopal), 12:36-45 (context-sensitive risk levels); EX1044 (Giakouminakis), 6:9-7:31 (prioritizing based on factors such as connection to other assets in the network), 11:51-67 (prioritizing vulnerabilities and using thresholds); EX1046 (Crabtree), 11:52-12:14 (prioritizing software vulnerabilities); EX1048 (Price), ¶43 (“For example, a particular issue can be identified 510 together with an indication of the

seriousness or priority of the issue.”), Fig. 5; EX1078 (Hufsmith), ¶¶121, 140 (application usage).

116. In my opinion, the combination of Veselov and Basavapatna merely involves using these well-known techniques to perform their known function with predictable results. A POSA would have recognized that Basavapatna’s security-assessment techniques would operate similarly in the context of Veselov’s snapshot-based security assessments of virtual cloud assets taught by Veselov. For example, security assessments of physical machines and security assessments of VMs (and snapshots of their virtual disks) typically examined the same types of data (e.g., applications, configurations, files, etc.) and identified the same types of security issues. *Supra*, §§VIII.A (virtualization), VIII.B (snapshots), VIII.D.1 (types of security risks); *see also* EX1007 (Veselov), 4:1-4 (“In any case, the snapshot is prepared so that the security assessment produces the results that the scanning system otherwise would have obtained if the security assessment were performed on the original virtual machine instance.”), 9:55-67 (examined data includes file information, OS information, network configurations, connections to other resources, installed software, logs, etc.).

117. As I discuss below, a POSA would also have been motivated to combine Veselov and Basavapatna for several reasons.

118. First, a POSA would have been motivated to combine the teachings of Veselov and Basavapatna to more accurately assess risks associated with particular vulnerabilities and assets as a whole. It was well known that computer assets faced a wide variety of different security risks. *Supra*, §VIII.D.1. Consistent with this well-known consideration, Basavapatna teaches determining risk levels associated with cyber threats and filtering the determined risks to allow users to obtain a more comprehensive and accurate view of risks to an asset. EX1008 (Basavapatna), ¶¶72 (“Risk metrics can be generated to illustrate or assess the risk evident in particular assets by virtue of the asset's exposure to a particular vulnerability and thereby also, in some instances, an associated threat known to exploit the particular vulnerability.”), 120 (“However, system administrators, and other users, often want to gain an overall view of their systems. Therefore, aggregating the risk metrics on a per-threat basis, per-vulnerability basis, a per-asset basis, and/or per-asset group basis allows the network monitor to provide such a function.”), 138 (“[N]etwork monitor 102 can list all assets, vulnerabilities, or threats, sorted by aggregate risk metric, or can list a top number, e.g., top ten, of the assets, vulnerabilities, or threats.”). It was also well known that VMs in cloud environments typically had the same types of security considerations as non-virtual assets. *Supra*, §VIII.D.1. A POSA therefore would have recognized that applying Basavapatna’s teachings—

e.g., detecting cyber threats risking the asset and communicating that information to users via prioritized or filtered alerts—would allow snapshot-based security assessments of VMs, such as those taught by Veselov, to provide more useful results that identify and quantify potential harms to the computing environment.

119. Second, a POSA would have been motivated to combine the teachings of Veselov and Basavapatna to obtain known benefits of prioritizing more critical cyber threats and/or filtering the less critical cyber threats. Basavapatna recognizes the well-known fact that assets typically face many security problems of varying importance due to the existence of numerous potential vulnerabilities and threats, and that some security problems and assets were more important than others. *E.g.*, EX1008 (Basavapatna), ¶72 (“In some systems, the number of identified, verified, catalogued, or otherwise known system vulnerabilities can be in excess of 30,000 or more, while the number of known threats exploiting these vulnerabilities can be in the range of 5,000 or more.”); *supra*, §VIII.D.3. Basavapatna also teaches well-known advantages of prioritizing more critical vulnerabilities and assets based on associated risk levels. EX1008 (Basavapatna), ¶¶120 (aggregate risk metrics “can help system administrators determine which metrics, which vulnerabilities, and which threats, pose the most serious problems for a system”), 138 (“Ranking assets, vulnerabilities, and/or threats according to aggregate risk scores allows a user to

quickly identify which assets are most at risk, or which vulnerabilities and threats are most dangerous for a system.”); *see also* EX1045 (Canzanese), ¶¶7-8; EX1046 (Crabtree), 11:52-12:14; EX1063 (Hibbert), ¶¶187-89 (alerts may “indicate the degree of danger”), 207-11, 215-16; EX1078 (Hufsmith), ¶¶6 (noting problems caused by overwhelming vulnerability reports), 155 (complexity can lead to “missed vulnerabilities and less secure code”), 165 (filtering reports to “avoid overloading the user”); 180 (risk classifications “make the security vulnerability faster for a developer to assess”), 183 (partial reports manage user’s cognitive load), 192 (presenting granular information about vulnerabilities of greater interest avoids “overloading the user with information about every security vulnerability”); *supra*, §VIII.D.2-3 (known benefits of prioritization).

120. Given these teachings, a POSA would have reasonably expected the combination of Veselov with Basavapatna to improve the identification and remediation of the more significant security issues. Basavapatna’s teachings are a natural fit with Veselov’s since Veselov broadly teaches detecting and reporting certain security risks for multiple vulnerabilities and threats, while Basavapatna provides additional details regarding determining risk, prioritizing more important threats, and filtering less important threats. *See supra*, §§XI.A-B (reference summaries). A POSA also would have recognized that applying Basavapatna’s

prioritization and filtering teachings would similarly benefit snapshot-based assessments, such as those described in Veselov, since such assessments would have been expected to address multiple security problems of varying importance. *Supra*, §§VIII.D.2-3; *see also* EX1025 (NIST Incident Handling), 32; EX1044 (Giakouminakis), 6:9-18; EX1046 (Crabtree), 11:52-12:14. Likewise, a POSA would have understood that determining risk levels associated with vulnerabilities and alerts would provide useful bases for prioritizing and remediating higher risk vulnerabilities (e.g., prioritizing cyber threats presenting greater risks to the computing environment and filtering out less important threats). *See, e.g., supra*, §§VIII.D.1-3; EX1025 (NIST Incident Handling), 3 (discussing basis for prioritization); EX1045 (Canzanese), ¶¶7-8.

121. Finally, a POSA would have had a reasonable expectation of success in combining the teachings of Veselov and Basavapatna to arrive at the claimed inventions. The snapshot-based techniques disclosed in Veselov were well understood and routinely practiced. *See, e.g., supra*, §VIII.D.2 (detecting threats and vulnerabilities). Determining risks associated with vulnerabilities and assets, as taught by Basavapatna, was similarly well-known and predictably implemented. EX1025 (NIST Incident Handling), 32 (“Prioritizing the handling of the incident is perhaps the most critical decision point in the incident handling process.”); EX1078

(Hufsmith), ¶¶31-33, 36, 96, 120, 128, 180-83; *supra*, §VIII.D.3 (alert prioritization). A POSA would have been familiar with the techniques applied in the combination of Veselov and Basavapatna—including, for example, taking snapshots, accessing snapshots, analyzing snapshots to detect cyber threats (e.g., based on matching against known lists), determining risks associated with the threats (based on factors such as application usage and asset configurations), prioritizing/filtering/alerting the threats, and reporting/mitigating security problems—and combining the references’ teachings would not have presented meaningful technical challenges or produced unexpected results. *Supra*, §§VIII.B (snapshots), VIII.C (cloud tools), VIII.D.1-3 (types of security issues, types/basis of analysis, and reporting techniques).

B. Independent Claims 1 and 18

1. Preambles

122. As I explain below, Veselov teaches the preambles of claims 1 and 18. Claim 1 recites a “method for securing virtual cloud assets in a cloud computing environment against cyber threats, comprising” and claim 18 recites a “system for securing virtual cloud assets in a cloud computing environment against cyber threats, comprising.”

123. Veselov describes methods and systems for securing virtual cloud assets against cyber threats. These methods and systems perform “a security assessment of

a target computing resource, such as a **virtual machine or an instance of a virtual machine.**” EX1007 (Veselov), Abstract (emphasis added); *see also id.*, Figs. 1-7 (depicting illustrative systems and methods involving security assessment of VMs); *supra*, §XI.A (Veselov summary). A POSA would have understood such VMs to be “virtual cloud assets” as recited, and I discuss the “cloud” aspect further below. Veselov explains that its assessments identify various types of security risks, including vulnerabilities associated with the target resource. *Id.*, 2:22-41 (assessment of “virtualized compute resources” includes identification of CVEs), 3:20-4:18 (“The scanning system may store the assessment results, or provide them to an owner or administrator of the virtual machine in order to evaluate the vulnerabilities of the target resources.”), 7:1-5 (scanning service may search for changes to settings “that exposed or repaired a security vulnerability”), 10:17-36 (assessment results and remediation provided to user).

124. The ’798 patent confirms that the types of security problems identified in the security assessments taught by Veselov are “cyber threats.” *E.g.*, EX1001, 3:13-16 (describing method of detecting “cyber threats including potential vulnerabilities”), 4:43-45 (same), 6:38-41 (same), 7:11-14 (same). *Compare id.*, 4:32-42 (disclosing use of CVEs as a means to detect vulnerabilities), *with* EX1007 (Veselov), 2:22-41 (teaching same). Basavapatna also expressly discusses

identifying “threats” that apply to the scanned asset and determining associated risk levels. *Infra*, §XII.B.5.

125. Veselov further teaches, or at a minimum suggests to a POSA, securing virtual “cloud” assets “in a cloud computing environment,” as recited. Veselov explains that its snapshot-based scanning techniques are particularly useful in cloud computing environments with virtualized resources. EX1007 (Veselov), 2:22-63 (discussing security challenges its techniques seek to overcome), 2:64-67 (“This problem may become more acute when the target of the security assessment is a service or distributed application being provided using a distributed computing environment (e.g., a **‘cloud’ computing environment**).” (emphasis added)); see also *id.*, 2:67-3:19 (discussing security challenges in virtualized cloud environments), 12:21-36 (resource may be performing services in “production environment”). This comports with the understanding of a POSA that cyber security techniques were important and routinely used to secure virtual assets in cloud computing environments. *Supra*, §VIII.D.1. Veselov also describes its system components in terms that a POSA would have understood to encompass typical cloud environments. EX1007 (Veselov), 19:30-37 (data stores may include devices “in any standard, distributed, virtual or clustered environment”), 19:60-64 (operations of the security assessment may be “performed collectively by multiple

devices, which may form a distributed and/or virtual system”), 20:46-56 (securing “a distributed and/or virtual computing environment utilizing several computer systems and components that are interconnected via communication links, using one or more computer networks or direct connections”).

126. Veselov also describes applying its security techniques to typical computing resources of cloud environments (i.e., virtual “cloud” assets). For example, Veselov explains that its teachings are “particularly suited for **security assessments performed in a data center at any level of abstraction**, non-limiting examples including: block-level storage devices, memory, hard disk drives, and other physical volumes; file systems; logical volumes and partitions; operating systems and **virtual file systems**; application-specific frameworks; and **virtual machines and virtual environments implementing virtual machines**.” EX1007 (Veselov), 2:27-35 (emphasis added). As I previously explained, a POSA would have understood that cloud computing services were commonly hosted on these types of resources—e.g., VMs executing on physical servers in data centers—further confirming that Veselov teaches applying its security techniques to “virtual cloud assets.” *Supra*, §§VIII.A, VIII.C. Thus, Veselov teaches securing virtual “cloud” assets “in a cloud computing environment.”

127. It is therefore my opinion that Veselov teaches the preambles of claims 1 and 18.

2. Element 18.i

128. Element 18.i recites “at least one hardware processor; and a storage device comprising computer readable instructions, to cause the at least one hardware processor to....” As I explain below, Veselov teaches this element.

129. First, Veselov teaches “at least one hardware processor,” as recited. Veselov discloses that the “physical hardware implementing any of the physical, logical, and/or virtual computing resources, the computing resource service provider 104, and/or the scanning service 110, may include one or more server computers.” EX1007 (Veselov), 5:23-31. Such servers include one or more processors to perform their operations. *Id.*, 5:31-40 (“A server computer may be equipped with any needed processing capability including one or more processors....”).

130. Furthermore, Veselov’s teaches a “storage device comprising computer readable instructions, to cause the at least one hardware processor to” perform the operations of claim 18, as recited. Veselov discloses the use of hard disks and random-access memory that store instructions executed by the processor to cause a computing device to perform the system’s operations. EX1007 (Veselov), 20:38-45 (“Each server typically will include an operating system that provides executable program instructions for the general administration and operation of that server and

typically will include a computer-readable storage medium (e.g., a hard disk, random access memory, read only memory, etc.) storing instructions that, when executed (i.e., as a result of being executed) by a processor of the server, allow the server to perform its intended functions.”), 20:57-60 (system includes “memory and a processor, the memory including computer-executable instructions”), 20:60-23:18 (scanning system’s actions caused by executing instructions), claim 1; *supra*, §XII.C.1. Veselov also teaches that a single system can perform the operations associated with both the scanning service and the target resource’s environment. *Id.*, 4:60-5:1 (“In some embodiments (e.g., as illustrated in FIG. 3B and described below), the scanning service 110 may be a component of the computing resource service provider 104, and may be implemented on the server computers 142 described in detail below. In other embodiments including the example of FIG. 1, the scanning service 110 may be implemented on one or more computing devices (not shown) outside of the computing resource service provider 104.”). In the context of the combination of references that I discuss in this ground and the combinations that I discuss in subsequent grounds, a POSA therefore would have understood that execution of the instructions contained on the medium would “cause the at least one hardware processor to” perform each of the operations recited in claim 18 and its dependent claims—claims 19-34.

3. Elements 1.1 and 18.1

131. These elements recite “[taking/take] a snapshot of at least one virtual disk of a protected virtual cloud asset, wherein the virtual cloud asset is instantiated in the cloud computing environment.” As I explain below, Veselov teaches these elements.

132. First, Veselov teaches “taking a snapshot,” as recited. Veselov’s security assessment is based on a snapshot of the target resource that represents the state of the resource at a particular point in time. EX1007 (Veselov), Abstract, 3:20-27 (scanning system obtains/accesses “a state of the resource at a point in time (e.g., a ‘snapshot’)”), 9:14-16 (snapshot represents “state of the target resource”); *see also id.*, 2:42-63, 4:53-54 (service may “obtain and analyze snapshot data 146”), 6:25-30 (service allows “analysis of various snapshot data 146”). Additionally, Veselov teaches the scanning service may take a snapshot as part of the process of obtaining/accessing a snapshot of a VM. For example, Veselov explains:

In various non-limiting examples, **the scanning service may obtain the snapshot data by executing the snapshot itself**, by requesting another service to capture the snapshot and provide it to the scanning service, by accessing a data store (e.g., the snapshot storage service) containing the snapshot, etc....

Id., 8:62-9:25 (emphasis added); *see also id.*, 3:22-27 (snapshots may be taken in “conjunction with, initiating the security assessment”), 16:15-26 (step 404 involves obtaining snapshot), 18:15-21 (step 602 involves obtaining snapshot).

133. Second, Veselov teaches the snapshot taken as part of its security assessment is a snapshot “of at least one virtual disk of a protected virtual cloud asset.” Discussing the embodiments of Figures 3A-B—which describes techniques and components broadly applicable to all embodiments (*supra*, ¶91)—Veselov notes that a “logical volume” was “also known as a ‘logical device’ or **‘virtual disk drive’** or ‘virtual storage device’ (e.g., the logical volumes 306, 316).” *Id.*, 11:28-33 (emphasis added); *see also* Figs. 3A-B (depicting logical volumes 306, 316). A POSA therefore would have understood Veselov’s various disclosures of scanning a snapshot of a logical volume to encompass scanning a snapshot of a virtual disk. *See, e.g., id.*, Abstract (“The snapshot can be an image of a logical volume implementing the target computing resource.”), 3:20-40 (snapshot may be “block-level image of the entire logical volume” or portions thereof), 3:61-67 (scanning system “may mount the logical volume (or transform the snapshot into a data file by identifying the file system of the original logical volume and then mount the snapshot in the duplicate logical volume) as a hard drive with data stored thereon; the system may then perform the security assessment on the data at the file system

level”), 6:7-13 (“The virtual computing environments enabled by the virtualization layer(s) 144 may include various virtual computer components, such as one or more virtual CPUs, virtual memory, virtual disk storage, and the like.”), 8:52-62 (target resource in Fig. 2 can be “one or more physical and/or virtual computing resources or segment(s) thereof, such as a virtual machine, virtual machine instance, virtual computing environment implementing a plurality of virtual machine instances, **logical volume**, physical or virtual storage device, etc.” (emphasis added)), 8:64-9:9 (target resource may be “logical volume attached to the corresponding virtual machine and/or containing the instance”), 11:5-33 (target VM may be stored on logical volume, which “may store data implementing and / or used by the virtual machine”), 11:34-41 (snapshot “may be a volume image 308 of at least that portion (e.g., a partition) of the physical storage device that is associated with the logical volume 306 and/or that contains the virtual machine 312”), 16:15-26 (in step 404 of Fig. 4, scanning service obtains “a logical volume image snapshot”), 18:16-26 (scanning service may “obtain a snapshot (i.e., volume image) of the virtual machine (step 604)”); *see also id.*, 11:41-67 (describing aspects of exemplary “volume image”); Figs. 2-6.

134. Veselov further teaches the taken snapshots in its security assessment are of “virtual disks” with its discussion of scanning virtual storage elements of

VMs. For example, Veselov describes performing the security assessment in a data center at any level of abstraction, including physical or virtual machines and elements thereof such as physical memory (e.g., “block-level storage devices” and “hard disk drives”) and virtual memory (e.g., “virtual file systems”). *Id.*, 2:22-35. A POSA would have understood these elements—e.g., “virtual file systems”—as including, or at a minimum suggesting, “at least one virtual disk of a protected virtual cloud asset” as recited. Veselov also explains that the snapshot may be:

a copy of the state of memory, **the state of any devices (virtual or physical)** allocated to the resource, block-level image of the entire logical volume; or, the snapshot may be an image of only a portion of the logical volume containing the data required to embody an exact copy of the virtual machine instance; or the snapshot may simply be a copy of certain files of the target computing resource, such as a database of software packages installed in a virtual machine.

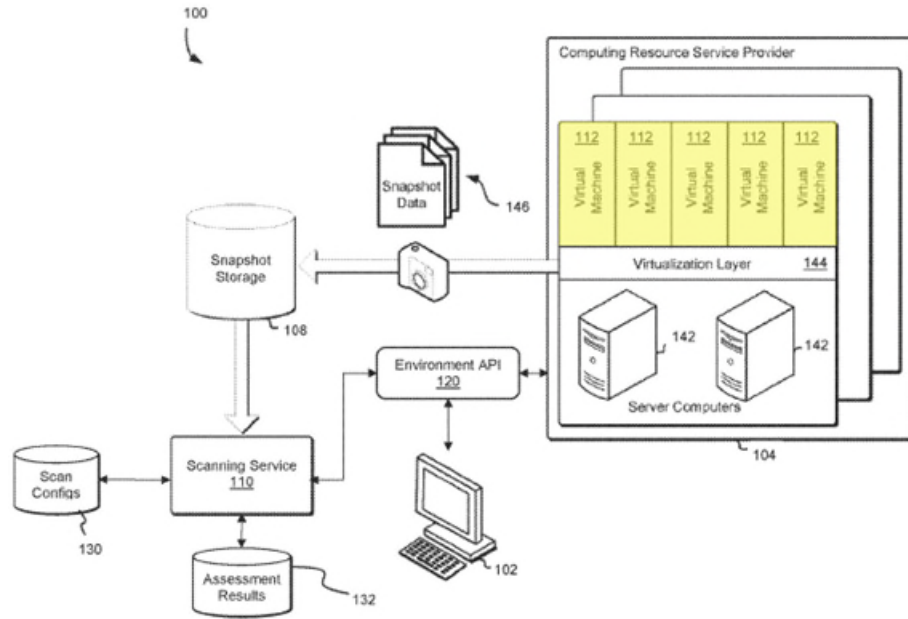
Id., 3:29-40. Based on my experience, a POSA would have understood the “state of any devices (virtual or physical),” especially read in the context of the surrounding discussion of the “state of memory” in general and copies of “certain files,” as encompassing virtual disks of virtual cloud assets. *Id.*

135. Moreover, for each of these disclosures regarding taking and scanning a snapshot of a virtual disk, it would have been clear to a POSA that Veselov is describing, or at least suggesting, a virtual disk of “a protected virtual cloud asset”

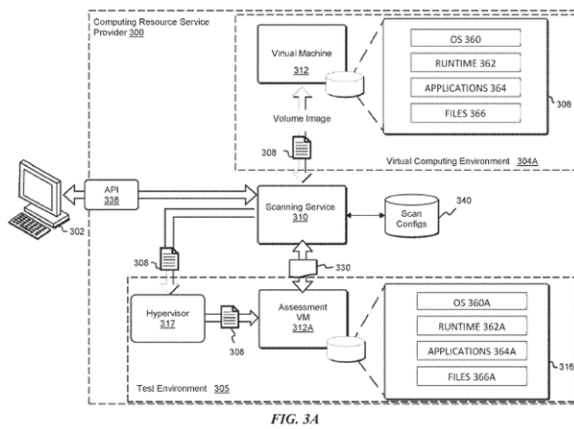
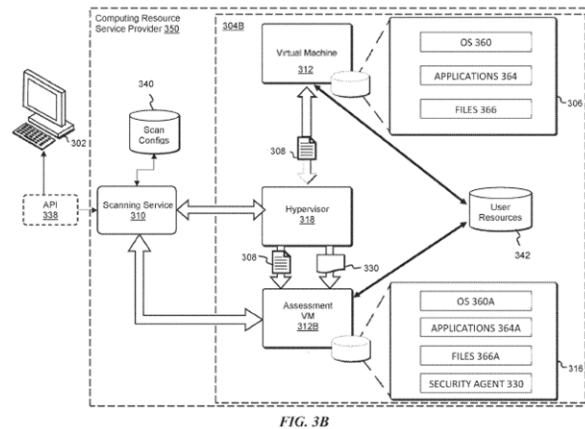
such as a VM. I previously explained why Veselov teaches performing its security assessments on “virtual cloud assets” such as VMs in a cloud computing environment (*supra*, §XII.B.1), and a POSA would have viewed such an asset to be a “protected” virtual cloud asset because it is being protected by the security assessment.

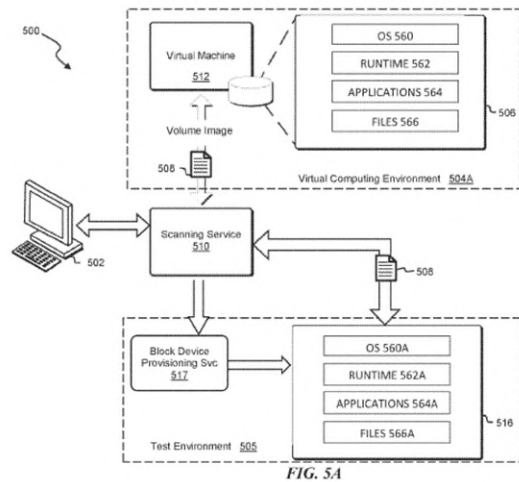
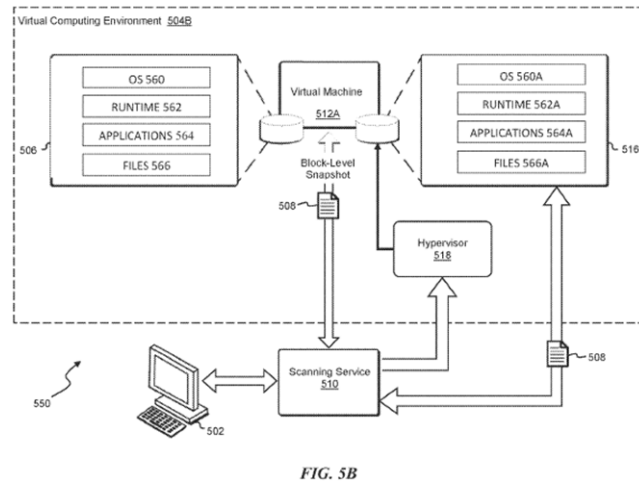
136. Thus, Veselov teaches that the security assessment includes “taking a snapshot of at least one virtual disk of a protected virtual cloud asset,” as recited.

137. Second, Veselov teaches “wherein the virtual cloud asset is instantiated in the cloud computing environment” because the virtual cloud assets are instantiated in the host environment (i.e., cloud computing environment). For example, a “security assessment may be initiated by receipt of a request” by a customer through a user device and a “server computer 142 may host a first virtual machine 112 instantiated from a first volume image and operated by [the] first customer.” EX1007 (Veselov), Abstract, 6:9-34; *see also id.*, 8:52-60 (target resource may be “one or more physical and/or virtual computing resources or segment(s) thereof, such as a virtual machine, virtual machine instance”), 10:37-48 (discussing the protected VM instances). An example of instantiated VMs is illustrated below in Fig. 1:



Id., Fig. 1 (annotation added). Figs. 3A-B and 5A-B depict VMs instantiated in analogous host environments:





Id., Figs. 3A-B, 5A-B.

138. It is therefore my opinion that Veselov teaches all aspects of elements 1.1 and 18.1.

4. Elements 1.2 and 18.2

139. These elements recite “[accessing/access] the snapshot of the at least one virtual disk.” As I explain below, Veselov teaches these elements.

140. Veselov’s teaches the scanning service “[accessing/access] the snapshot of the at least one virtual disk,” as recited. Veselov’s scanning service accesses the snapshot regardless of whether the scanning service obtains a copy of the snapshot to be analyzed elsewhere or accesses the snapshot so that the analysis is performed in the original environment. For example, in discussing Fig. 2, Veselov contrasts obtaining the snapshot (which involves bringing a copy of the snapshot

into the local testing environment in various ways) and obtaining access to the snapshot:

In various non-limiting examples, **the scanning service may obtain the snapshot data** by executing the snapshot itself, by requesting another service to capture the snapshot and provide it to the scanning service, by accessing a data store (e.g., the snapshot storage service) containing the snapshot, etc.; **alternatively, the scanning system may obtain access to the snapshot data, such as at a storage location of the snapshot data, but may not obtain (i.e., make a copy of) the snapshot data.**

EX1007 (Veselov), 9:14-25 (emphasis added). Veselov discusses how “the scanning system **obtains, or obtains access to**, a state of the resource at a point in time (e.g., a ‘snapshot’) prior to, or in conjunction with, initiating the security assessment.” *Id.*, 3:20-27 (emphasis added); *see also id.*, 4:53-55 (“In some embodiments, the scanning service 110 may obtain and analyze snapshot data 146.”), 5:9-22 (“[T]he computing resource provider 104 may provide the snapshot data 146 to the scanning service 110, or may store the snapshot data 146 or otherwise make the snapshot data 146 accessible by the scanning service 110.”), 6:25-49 (“The virtualization layer 144 may be configured to obtain the snapshot data 146 and send it to the scanning service 110 or store the snapshot data 146 via a snapshot storage service 108.”), 11:34-36 (“A scanning service 310 tasked with performing the

security assessment of the virtual machine 312 obtains snapshot data of the virtual machine 312 as described above.”), 22:60-23:18 (describing computing resource provider making snapshot data available), Figs. 2, 4, 6 (depicting obtaining step), claim 1 (processor configured to “obtain access to a first volume image”). Even though Veselov does not use the word “access” in the first approach, it would have been apparent to a POSA that obtaining the snapshot entails accessing the snapshot so that it can be obtained. EX1074 (VMware Infrastructure), 9 (describing a file storage system (“VMFS”) that stores and provides access to snapshot files by providing “a mechanism for a virtual machine to have direct access to a LUN on the physical storage subsystem (Fiber Channel or iSCSI only)”), Fig. 1-10 (illustrating the storage architecture of the datastore containing VM snapshot .vmdk files and how they are accessed).

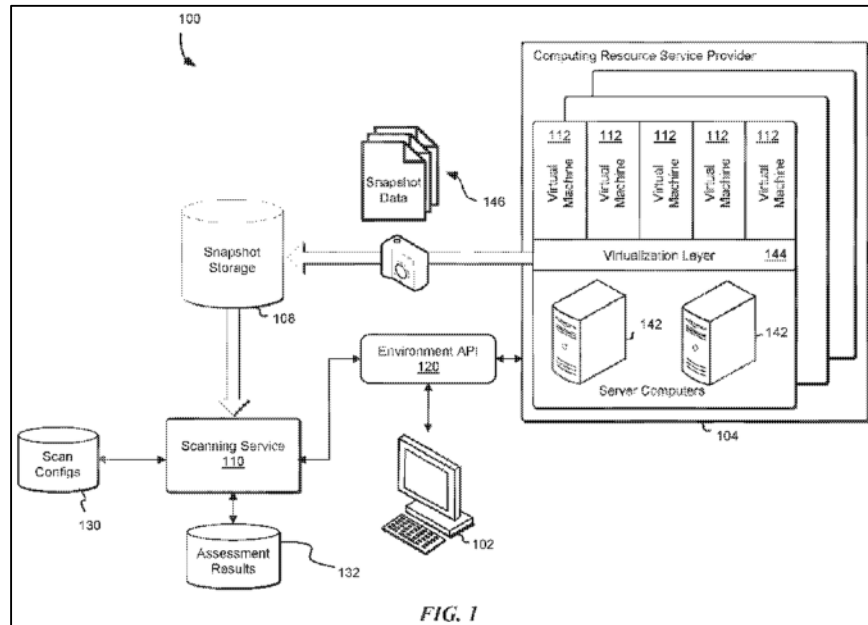
141. It is therefore my opinion that Veselov teaches all aspects of elements 1.2 and 18.2.

5. Elements 1.3 and 18.3

142. These elements recite “[analyzing/analyze] the snapshot of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset.” As I explain below, the combination of Veselov and Basavapatna teaches these elements.

143. First, Veselov teaches analyzing the snapshot via its scanning service.

The scanning service is shown, for example, as scanning service 110 in Fig. 1 below:



EX1007 (Veselov), Fig. 1; *see also id.*, Figs. 3A-B (depicting scanning service 310), 5A-B (depicting scanning service 510). As Veselov explains, “the scanning service 110 may obtain and analyze snapshot data 146.” EX1007 (Veselov), 4:49-54 (emphasis added); *see also id.*, 8:16-17 (“The scanning service 110 performs the desired security assessment using the snapshot data 146.”), 8:17-51 (describing variety of analysis techniques), Figs. 1 (depicting scanning service 110 analyzing snapshot data 146 to produce assessment results 132), 2 (step 210), Figs. 3A-B (scanning service 310), 4 (step 414), 5A-B (scanning service 510), 6 (step 614).

144. Veselov's snapshot-based security analysis can be performed in multiple ways. For example, in the following passage, I have emphasized Veselov's description of two different ways the analysis can be performed:

At step 208, the scanning service may generate a scannable volume, or cause a scannable volume to be generated, based at least in part on the snapshot data. A scannable volume is any suitable configuration of computing resources allocated for the purpose of the security assessment, into which the snapshot data may be installed, mounted, copied, or otherwise applied to produce a duplicate implementation of the target resource that can be scanned by the scanning service.

Examples of generating the scannable volume described below include launching a duplicate virtual machine instance in an allocated logical volume, mounting the snapshot data in an allocated logical volume like a storage device and reading the target resource state like a data file, and incorporating data elements of the snapshot data into a partial framework of the target resource to produce the duplicate target resource.

Id., 10:1-16 (emphasis added). Thus, the analysis can be performed by analyzing an assessment VM instantiated based on the snapshot, or the snapshot data can be analyzed directly (i.e., without instantiating an assessment VM). Each of these types of analysis satisfies the construction of analyzing the at least one snapshot that I discussed above in §X.A.

145. In the first approach, the scanning service uses the snapshot to launch an active instance of the VM (which Veselov refers to as an assessment VM) that is essentially a duplicate of the target VM at the time the snapshot was captured. EX1007 (Veselov), 3:54-61 (“In one embodiment, the scanning system may set up and launch an active virtual machine instance in the allocated computing resources, the active virtual machine instance having the same configuration as the original virtual machine instance had at the time the snapshot was captured; the system may then perform the security assessment on the active virtual machine instance in a static or runtime environment.”), 12:1-20 (“In an example embodiment, the scanning service 310 may produce, or cause the production of, a duplicate of the virtual machine 312 (e.g., of an instance thereof) using the volume image....”). This active instance of the virtual machine can be instantiated in the original computing environment or in a separate computing environment. *Id.*, 3:44-51 (“In some embodiments, the resources for the duplicate virtual machine instance may be within a virtual computing environment of the original virtual machine instance, while in certain embodiments the duplicate virtual machine instance may be implemented outside of the original virtual computing environment, such as within an account of the service provider used to perform analysis on resources.”); *see also id.*, 3:51-54 (“The scanning system may implement the duplicate virtual machine instance in any

manner that allows the scanning system to perform the desired security assessment on the duplicate instance.”), 14:13-22 (scanning service “may perform any suitable security assessment” on assessment virtual machine), Figs. 3A (assessment virtual machine 312A in original computing environment), 3B (assessment virtual machine 312B in testing environment).

146. In the second approach, the scanning service analyzes the snapshot data without instantiating a live instance of the target VM. For example, “the scanning system may mount the logical volume (or transform the snapshot into a data file by identifying the file system of the original logical volume and then mount the snapshot in the duplicate logical volume) as a hard drive with data stored thereon....” *Id.*, 3:61-67. The scanning system “may then perform the security assessment on the data at the file system level.” *Id.*; *see also id.*, 16:56-29 (“The scanning service 510 may then mount the volume image 508 in the new logical volume 516 so that the scanning service 510 can access the volume image 508 (e.g., can read files and other data such as the OS 560A, runtime environment 562A, software applications 564A, and files/file system 566A) as if it were a disk drive.”). Veselov also describes this approach when discussing the variety of ways the snapshot data can be processed:

The scanning service 110 may, in some embodiments, use the scan data to determine the parameters of the security assessment, and

further to identify the data elements of the virtual machine 112 that are needed to perform the security assessment. The scanning service 110 may use this information to configure the scan environment, non-limiting examples of which include: determine that the security assessment is performed on static or dynamic data; request that the snapshot include only the subset of virtual machine instance data needed to perform the security assessment; **determine whether to reconstitute an active duplicate virtual machine instance or to analyze the snapshot as a data file**, as described below; determine whether to allocate a duplicate logical volume inside or outside of the computing resource service provider 104; determine where and/or how to store and/or transmit the assessment results; and the like.

Id., 8:30-45 (emphasis added).

147. Neither of these analysis techniques in Veselov uses an agent installed on the target VM, nor do they require an agent installed on the assessment VM. These types of scanning techniques—i.e., security scans that are not performed by agents installed on the target instance of a VM—were consistent with well-known security-assessment techniques. *See, e.g.*, EX1040 (Ammons), Abstract (monitoring virtual server “without utilizing agents executing within the virtual server”), ¶¶8-12, 23, 43-45, 48, 52; EX1073 (Ammons Mirage), 1 (discussing “offline image content introspection, and manipulation capabilities”); EX1095 (Burdett), ¶¶5-6 (noting desirability of performing security analysis “without use of an agent installed on the

template image used in cloud-based environments”); *see also supra*, §VIII.D.2 (discussing other examples of agentless scanning).

148. For both of these analysis techniques, Veselov also teaches that the analysis is done “to detect potential cyber threats risking the protected virtual cloud asset,” as recited. Veselov broadly teaches detecting a wide variety of potential security problems risking the VM. It first discusses known security-assessment services that identify “vulnerabilities, bad configurations, and the like.” EX1007 (Veselov), 1:41:-49. Veselov then discusses the expansive scope of the types of security problems that its scanning service may detect when analyzing the snapshot:

The present disclosure contemplates implementation of any suitable security assessment, including security assessments that are defined by rules packages, such as Common Vulnerabilities and Exposures (CVEs), Center for Internet Security (CIS) benchmarks, “best practices” packages, static or runtime behavior analysis, host configuration assessments, and the like.

Id., 2:35-41; *see also id.*, 4:1-18 (assessment allows user to “evaluate the vulnerabilities of the target resources”), 4:53-59 (assessment identifies “security risk information”), 5:9-11 (“The scanning service 110 may be configured to perform security assessments and produce assessment results based at least in part on snapshot data....”), 7:1-4 (change log data in snapshot may allow determination of what caused security problem), 10:17-36 (performing remediation “if vulnerabilities

are identified in the assessment results”), 14:41-15:3 (scan configuration determines “tests or tasks to perform, types of vulnerabilities to evaluate, and the like”), 20:57-21:3, 23:19-24:4, claims 1, 16-18. As I previously explained, common security assessments performed by systems like Veselov’s routinely detected cyber threats to protected virtual cloud asset. *Supra*, §VIII.D.1-2. As I explain below, this broad disclosure of identifying different types of security problems teaches analyzing the snapshot to detect “potential cyber threats risking the protected virtual cloud asset” in several ways, including Veselov’s discussion of CVE assessments and configuration assessments.

149. First, Veselov teaches applying rule packages such as CVEs (i.e., Common Vulnerabilities and Exposures, which I discussed in §VIII.D.1). *Id.*, 2:22-54; *see also id.*, 7:16-22 (discussing “software application-level CVEs assessment”), 14:41-65 (user/administrator may configure scanning service to perform CVE assessment). It was well known that the types of security assessments taught by Veselov, such as CVE assessments, typically identified a plurality of security problems on a target resource such as a VM. *See, e.g.*, EX1007 (Veselov), 2:22-41; EX1077 (Khalid), 8:1-15 (determining information associated with an alert, including “one or more identifiers associated with Common Vulnerabilities and Exposures (CVEs) that is, one or more identifiers for publicly known cyber security

vulnerabilities”). Moreover, as I explained above when discussing the claim preambles, the potential security problems detected by Veselov’s security assessment include potential “cyber threats.” *Supra*, §XII.C.1. As one example, a POSA would have known that many CVEs involved features of applications that may pose a security threat to the system on which the applications are installed. *See, e.g.*, EX1090 (CVE-2006-2923)³⁴ (security risk associated with library used by multiple software products); EX1091 (CVE-2007-0018)³⁵ (same). A POSA also would have known that many CVEs were associated with services that were commonly run when the computer system starts up. *See, e.g.*, EX1100 (CVE-2018-

³⁴ EX1090 is an archived copy of “Vulnerability Details: CVE-2006-2923,” archived on July 29, 2014, by the Internet Archive’s “Wayback Machine” at <https://web.archive.org/web/20140729171850/https://www.cvedetails.com/cve/CVE-2006-2923/>.

³⁵ EX1091 is an archived copy of “Vulnerability Details: CVE-2007-0018,” archived on August 9, 2014, by the Internet Archive’s “Wayback Machine” at <https://web.archive.org/web/20140809144720/https://www.cvedetails.com/cve/CVE-2007-0018/>.

6954)³⁶; EX1101 (CVE-2018-15687)³⁷; EX1102 (CVE-2018-1312)³⁸; EX1103 (CVE-2013-2765)³⁹. The '798 specification confirms that both of these types of

³⁶ EX1100 is an archived copy of “Vulnerability Details: CVE-2018-6954,” archived on December 11, 2018, by the Internet Archive’s “Wayback Machine” at <https://web.archive.org/web/20181211090742/http://www.cvedetails.com/cve/CVE-2018-6954>.

³⁷ EX1101 is an archived copy of “CVE-ID: CVE-2018-15687,” archived on December 11, 2018, by the Internet Archive’s “Wayback Machine” at <https://web.archive.org/web/20181211082546/https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-15687>.

³⁸ EX1102 is an archived copy of “CVE-ID: CVE-2018-1312,” archived on January 5, 2019, by the Internet Archive’s “Wayback Machine” at <https://web.archive.org/web/20190105194958/https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-1312>.

³⁹ EX1103 is an archived copy of “Vulnerability Details: CVE-2013-2765,” archived on July 20, 2014, by the Internet Archive’s “Wayback Machine” at

security issues are “potential cyber threats,” as recited, since it identifies them as examples of cyber threats:

In yet another embodiment, the security system 140 is configured to detect cyber threats that do not represent vulnerabilities. For example, the security system 140 may detect and alert on sensitive data not being encrypted on the logical disk, private keys found on the disks, system credentials stored clearly on the disk, **risky application features** (e.g., support of weak cipher suites or authentication methods), weak passwords, weak encryption schemes, a disable address space layout randomization (ASLR) feature, suspicious manipulation to a boot record, suspicious PATH, LD_LIBRARY_PATH, or LD_PRELOAD definitions, **services running on startup, and the like**.

EX1001, 6:19-30 (emphasis added). Moreover, the '798 specification indicates that vulnerabilities are a type of cyber threat. EX1001, 4:43-45 (“[T]he security system 140 is configured to detect vulnerabilities and other cyber threats related to the execution VM 119.”), 6:38-41 (same), 7:11-14 (same). Thus, in my opinion,

<https://web.archive.org/web/20140720123338/https://www.cvedetails.com/cve/CVE-2013-2765>.

Veselov’s discussion of performing CVE assessments as part of its snapshot analysis teaches, or at least suggests, analyzing the snapshot to determine the existence of a potential cyber threats risking the protected virtual cloud asset.

150. Second, Veselov teaches identifying nonsecure system configurations, which encompass additional types of “potential cyber threats risking the protected virtual cloud asset.” As I noted above, Veselov discusses known security assessment services that identify “vulnerabilities, bad configurations, and the like.” EX1007 (Veselov), 1:46-49. Veselov further explains that its own analysis techniques may include “host configuration assessments” (*id.*, 2:35-41) and that the analyzed snapshot may include various types of configuration information (*id.*, 9:42-67). For example, the analyzed snapshot may include properties such as “properties of logical volume(s) containing the target resource; properties of physical storage devices allocated to the logical volumes; file system and file information; operating system information; system and software patches applied; configurations of network protocols and services; properties of connections to other computing resources; permission settings; software packages installed, and configurations thereof; system log properties; and the like.” *Id.*, 9:42-67; *see also id.*, 21:53-22:6 (“The security assessment may include an evaluation of a virtual machine configuration....”), claim 9 (same). Given Veselov’s discussion of performing host configuration assessments

as part of the snapshot analysis, a POSA would have understood Veselov as teaching, or at least suggesting, that the assessment detects security problems based on the target asset's configuration data, including the types of configuration data in the snapshot that I noted above. *Id.*, 9:42-67 (listing properties that may be part of snapshot, including configuration information). A POSA also would have understood such security problems to include potential cyber “threats” since unsecure system configurations pose potential threats to system security. This understanding is also consistent with the '798 patent specification since the listed examples of threats include some unsecure asset configurations (e.g., weak passwords and weak encryption schemes). *See* EX1001, 6:19-30.

151. Consistent with Veselov, Basavapatna also teaches performing an analysis on a computer asset “to detect potential cyber threats” risking the protected asset, as recited. Basavapatna describes detecting security problems as part of determining threat-centric risk metrics and vulnerability-centric risk metrics (*supra*, §XI.B), and both involve detecting “potential cyber threats” risking the asset. *See also supra*, §XII.A (reasons to combine). For the threat-centric analysis, Basavapatna describes detecting potential threats targeting the asset. For example, Basavapatna describes processing “data about potential threats on the network ... to generate risk metrics for assets and threats.” EX1008 (Basavapatna), ¶31; *see also*

id., 20 (threats include “unauthorized attacks”). This data includes threat definition data (which provides information about the potential threats) and the asset’s configurations (which allows the security system to determine which threats are applicable to the asset). *Id.*, ¶¶16 (overview of threat factors used to determine threat-centric risk), 34 (network monitor receives threat definition data that describes “identified threats, what countermeasures (if any) protect assets from the threats, and the severity of the threat”), 37 (“Depending on the configuration, various vulnerabilities and/or threats may be applicable to an asset.”), 38-40 (threat definition data can specify threat vectors for each threat which represents “represents a particular attack associated with the threat”), 43-49, 76-81. The threat definition data may include “applicability data” specifying “a configuration that an asset must have in order to be vulnerable to the threat.” *Id.*, ¶44 (applicability data can specify that threat “only attacks particular operating systems or particular software products installed on an asset” or particular versions/configurations thereof); *see also id.*, ¶83 (threat-centric analysis determines “when the asset is running software that has a known vulnerability that can be exploited by the threat”). As Basavapatna explains, the threat’s applicability data is compared with asset configuration data to determine if the threat applies to the asset. *Id.*, ¶¶76-81. Basavapatna’s threat-centric analysis thus teaches detecting “potential cyber threats risking” the protected asset, as recited.

152. Basavapatna also teaches detecting potential cyber threats risking the asset with its discussion of vulnerability-centric analysis. Basavapatna detects vulnerabilities in an asset based on vulnerability definition data which includes information regarding potential cyber threats to the asset. *See id.*, ¶¶99-104. For example, the vulnerability definition data “identifies one or more vulnerabilities” and is “associated with known threats ... by virtue of an identification that a particular threat exploits a particular vulnerability.” *Id.*, ¶52. Basavapatna explains that such “relationships can be specified in one or more threat vectors and may be further linked to or identified within the vulnerability definition data.” *Id.* Basavapatna also teaches detecting cyber threats by comparing vulnerability definition data to the asset configuration data to detect, for example, non-secure configurations. For instance, Basavapatna teaches that a vulnerability is applicable to an asset if it includes asset configurations—such as operating systems, applications, and versions thereof—that are “known to introduce the vulnerability onto the asset.” *Id.*, ¶53 (“[M]isconfiguration of a web browser can be identified as a vulnerability that can be exploited via visiting a malicious website.”); *see also id.*, ¶60 (discussing vulnerability detection data). The ’798 patent confirms that these types of potential security problems are “cyber threats.” EX1001, 4:43-51 (discussing detection of vulnerabilities and “other” cyber threats such as “vulnerable

software, non-secure configuration[s], ... and so on”). Basavapatna further teaches detecting such cyber threat that are risking the asset with the calculation of a vulnerability detection score that estimates whether the asset has a particular security problem. EX1008 (Basavapatna), ¶104. Basavapatna also explains that, for vulnerabilities associated with a known threat, the vulnerability score estimates “whether the asset is vulnerable to the associated threat.” *Id.* Basavapatna’s vulnerability-centric analysis thus teaches detecting “potential cyber threats risking” the protected asset, as recited.

153. It is therefore my opinion that Veselov alone, or in combination with Basavapatna, teaches all aspects of elements 1.3 and 18.3.

6. Elements 1.4 and 18.4

154. These elements recite “[alerting/alert] detected potential cyber threats based on filtering or prioritizing using a determined risk of each detected threat.” As I explain below, the combination of Veselov and Basavapatna teaches this element.

155. As an initial matter, Veselov teaches “alerting detected potential cyber threats.” Veselov’s assessment results describe the identified risks, including vulnerabilities discovered in the snapshot data. *E.g.*, EX1007 (Veselov), 10:17-36 (“Generally, performing the security assessment includes applying the rules of the assessment to the properties of the reproduced target resource to produce assessment results that describe ‘findings’ relative to the assessed properties....”), 20:57-21:13

(system “perform[s] a security assessment on the second virtual machine instance to produce a first assessment result describing a security vulnerability of the first virtual machine instance”), 23:19-47 (method includes “performing the security assessment on the second computing resource to produce a first assessment result describing one or more security vulnerabilities of the first computing resource”); *see also id.*, 5:9-13 (“The scanning service 110 may be configured to perform security assessments and produce assessment results based at least in part on snapshot data...”), 8:19-21 (scanning service may “process the snapshot data 146 to generate one or more assessment results”), 10:17-36 (generating assessment results), claim 1. Veselov then provides the assessment results, including the identified vulnerabilities, to the user through a user device and/or user interface (i.e., reports at least some of the determined potential cyber threats). For example, once the scanning service produces the assessment results it “provid[es] the assessment results in a user-readable format to a user interface to be displayed on the user device” and identifies remedial actions. *Id.*, 10:30-36, 18:53-56 (“[T]he scanning service may perform actions on the assessment results, such as storing or delivering the results...”); *see also id.*, Figs. 2, 4, 6 (depicting steps 212, 416, and 616 that provide assessment results). Thus, consistent with common practice in the field at the time, Veselov notifies the user of detected cyber threats by, for example, sending the results to be

displayed to user. *Supra*, §VIII.D.3 (sending alerts regarding detected security issues was commonplace); *see, e.g.*, EX1025 (NIST Incident Handling), 21.

156. While Veselov does not expressly state that the alerting is “based on filtering or prioritizing using a determined risk of each detected threat,” this type of alerting was commonplace well before 2019. Basavapatna describes an example of the common technique of prioritizing and filtering potential cyber threats as part of an analogous security assessment. Basavapatna describes multiple methods for determining priorities of and filtering cyber threats based on a determined risk, including the use of a vulnerability-centric and a threat-centric risk metrics and various aggregate risk metrics.

157. First, Basavapatna teaches determining risk associated with cyber threats with its discussion of generating threat-centric risk metrics. The threat-centric risk determines a risk metric for a given threat that exploits one or more vulnerabilities on a given asset. *See* EX1008 (Basavapatna), ¶¶73-98, Fig. 3A. It is calculated based on multiple factors that quantify risk associated with this type of cyber threat to the asset. In a disclosed example, the threat-centric risk metric is based on an exposure factor (which is based on a threat factor, vulnerability score, and countermeasure score) and a criticality score. *Id.*, ¶¶82, 92. The “threat factor” is based on “threat severity score” (received from a threat feed and specifying

information such as one or more vulnerabilities exploited by the threat) and an “applicability score” (indicating whether or not the asset has the hardware/software vulnerabilities associated with the threat). *Id.*, ¶¶75-76 (threat factor), 77 (threat severity score), 78-81 (applicability score). The “vulnerability score” represents the likelihood that an asset has a particular vulnerability targeted by the threat. *Id.*, ¶¶83-85.

158. Second, Basavapatna teaches determining risk associated with cyber threats with its discussion of generating vulnerability-centric risk metrics. *See* EX1008 (Basavapatna), ¶¶99-113. This process determines a risk metric for a given vulnerability on a given asset (or in some cases a set of related vulnerabilities on the asset) based on several underlying scores, including a “standard vulnerability score” and a “composite vulnerability score.” *Id.*, ¶¶100 (vulnerability-centric risk metric “for an asset and a vulnerability”), 101-02 (standard vulnerability score), 106 (composite vulnerability score), 112 (ultimate calculation of risk metric), Fig. 3B; *see also id.*, ¶¶103, 113 (discussing optional combination of risk scores for related vulnerabilities, such as multiple vulnerabilities tied to a common threat). The standardized vulnerability score (e.g., a CVSS score) indicates “a relative level of risk associated with the particular vulnerability.” *Id.*, Abstract, ¶¶101-02. The composite vulnerability score is calculated based on this standardized vulnerability

score and a vulnerability detection score (representing the probability that an asset has the vulnerability) estimates “the potential risk a particular vulnerability poses to an asset.” *Id.*, ¶¶104-06. The vulnerability-centric risk metric is then calculated based on the composite vulnerability score and a countermeasure score. *Id.*, ¶112. As I previously explained, a POSA would have understood the vulnerabilities discussed in Basavapatna to encompass cyber threats (*supra*, §XII.B.5), so a POSA would have similarly understood the risk determinations for each vulnerability discussed in Basavapatna as encompassing risk determinations of each cyber threat.

159. Third, Basavapatna teaches determining risk associated with cyber threats with its discussion of aggregating risk metrics in at least two ways. First, Basavapatna teaches aggregating risk metrics on a per-threat/vulnerability basis. For example, Basavapatna teaches receiving the vulnerability-centric risk metric for a particular vulnerability for each asset of a group of assets and then calculating a per-vulnerability aggregate risk metric based on the sum of those risk metrics. *Id.*, ¶¶123-128 (“The process calculates an aggregate risk metric for the particular vulnerability from the risk metrics for the assets for the particular vulnerability”). Basavapatna teaches “similar techniques can be used to aggregate threat-centric risk metrics on a per-threat basis.” *Id.*, ¶122. Second, Basavapatna teaches calculating

an asset-level risk metric that may be based on the vulnerability-centric or threat-centric risk metrics. *Id.*, ¶130 (process can be “applied to either threat-centric risk metrics or vulnerability-centric metrics for an asset”), Fig. 5. For example, the aggregate risk metric for a given asset may be a sum, mean, maximum, or other combination of the risk metrics for the detected cyber threats on that asset. *Id.*, ¶¶132 (“In some implementations, the aggregate risk metric is a sum of the risk metrics for the asset for the vulnerabilities....”), 133 (“[I]n other implementations, the aggregate risk metric is the mean risk metric for vulnerabilities that are applicable to the asset.”), 134-35 (aggregate risk metric may also be maximum, median, mode, etc.).

160. Basavapatna further describes “alerting detected potential cyber threats based on filtering or prioritizing using” the determined risk of each detected threat, as recited. Basavapatna teaches that the threat and vulnerability “risk metrics and aggregated risk metrics can be used in various ways” such as allowing the user to “view assets, vulnerabilities, or threats, sorted by the aggregate risk metric for the asset, vulnerability, or threat.” *Id.*, ¶138. For example, Basavapatna explains that the system may filter and/or prioritize the potential cyber threats by listing “all assets, vulnerabilities, or threats, sorted by aggregate risk metric, **or can list a top number, e.g., top ten, of the assets, vulnerabilities, or threats.**” *Id.* (emphasis added); see also *id.*, ¶139 (“The user interface lists the names of the assets 602,

sorted by aggregate risk metric as well as the aggregate risk metrics 604 themselves.”). A POSA would have understood that by listing only the top ten assets, threats, or vulnerabilities, the potential cyber threats would have been filtered and prioritized based on the determined risk. Moreover, Basavapatna teaches that “if an aggregate risk metric for any vulnerability or threat rises above a specified threshold, the user should be alerted,” thereby filtering out and deprioritizing threats below the specified threshold. *Id.*, ¶140 (rules used to “filter what data the user views”). Basavapatna also explains the well-known importance of such filtering and prioritization, noting that “[r]anking assets, vulnerabilities, and/or threats according to aggregate risk scores allows a user to quickly identify which assets are most at risk, or which vulnerabilities and threats are most dangerous for a system.” *Id.*, ¶138.

161. It is therefore my opinion that claims 1 and 18 were obvious.

C. Dependent Claims

1. Claims 2 and 19

162. Claims 2 and 19 depend from claims 1 and 18, respectively, and further recite “[prioritizing/prioritize] each of the detected potential cyber threats based on their respective risk to the protected virtual cloud asset.” I previously explained how the combination of Veselov and Basavapatna teaches alerting the detected potential cyber threats based on filtering or prioritizing—and my discussion addressed both filtering and prioritizing—using a determined risk for each detected threat. *Supra*,

§XII.B.6; *see also supra*, XII.B.1 (combination teaches performing techniques for protected “virtual cloud” assets). As I explain below, the combination of Veselov and Basavapatna teaches the additional aspects recited in claims 2 and 19. In particular, I address prioritizing “each” of the detected potential cyber threats and performing this prioritizing of each of the detected potential cyber threats “based on their respective risk to the protected virtual cloud asset.”

163. Both Veselov and Basavapatna teach that alerts are provided for “each” of the detected potential cyber threats. *See supra*, §XII.B.6. For example, Veselov teaches generating assessment results describing findings for the scanned asset and then “providing the assessment results” to the user’s device for display. EX1007 (Veselov), 10:17-36. Basavapatna similarly discusses determining risk metrics and performing the subsequent alerting process for each detected security issue. EX1008 (Basavapatna), ¶¶14-16 (determine risk “for each of the one or more vulnerabilities” on each asset), 30 (“Therefore, it is helpful to determine a risk metric for each asset and each threat.”), 138 (“For example, the network monitor 102 can list all assets, vulnerabilities, or threats, sorted by aggregate risk metric....”), claim 27 (reciting “determining a respective threat-centric risk metric, the determining including, for each asset and each threat...”).

164. Basavapatna further teaches determining a risk metric for each cyber threat “based on their respective risk to the” protected asset, as recited. As I discussed above, Basavapatna discloses determining threat-centric and vulnerability-centric risk metrics for each detected cyber threat. *Supra*, §XII.B.6; EX1008 (Basavapatna), ¶¶92, 112. As I discuss in detail below, both the threat-centric and vulnerability-centric risk metrics represent a risk to the protected asset. Moreover, Basavapatna uses the threat-centric and vulnerability-centric risk metrics to determine per-threat/vulnerability and asset-level aggregate risk metrics. *Supra*, §XII.B.6; EX1008 (Basavapatna), ¶¶120-135. Accordingly, for the same reasons discussed below, these aggregate risk metrics also represent a risk to the asset.

165. First, the threat-centric risk metric is based on several factors in order to represent a risk to the asset, including the exposure factor, the applicability score, and the countermeasure score. *See id.*, ¶¶82, 92. The exposure factor for the asset “estimates the risk that an asset will be affected by the threat.” *Id.*, ¶82. In a similar fashion, the applicability score estimates “the applicability of the particular threat to the particular asset.” *Id.*, ¶16. Lastly, the countermeasure component score estimates the probability that “a countermeasure will mitigate [a] risk associated with the particular vulnerability.” *Id.*, ¶12. A POSA would have understood that each of these “risks” associated with a detected cyber threat were risks to the asset

which contained the cyber threat. A POSA therefore would have understood that the threat-centric risk metric calculated based on these factors represents a risk to the asset.

166. Second, the vulnerability-centric risk metric is also based on several factors in order to represent a risk to the asset, including a standardized vulnerability score, a composite vulnerability score, and a countermeasure component score. *Id.*, ¶¶106, 112. The standardized vulnerability score indicates “a relative level of risk associated with the particular vulnerability.” *Id.*, ¶12. Further, the composite vulnerability score considers the relative risks of a particular vulnerability to a particular asset and the probability that a particular asset possesses a particular vulnerability. *Id.*, ¶¶12-13 (describing a “vulnerability detection score indicating an estimated probability that a particular asset possess[es] the particular vulnerability”). Finally, for the same reasons discussed regarding counter measure score used in the vulnerability-centric metric, the countermeasure component score also represents a risk to the asset. *Id.*, ¶12. A POSA therefore would have understood that the vulnerability-centric risk metric based on these factors represents a risk to the asset.

167. Basavapatna also teaches prioritizing each of the detected potential cyber threats based on their associated vulnerability-centric, threat-centric, and aggregate risk metrics (i.e., “based on” their respective risk to the protected virtual

cloud asset,” as recited). As I discussed above, Basavapatna teaches “alerting detected potential cyber threats” based on these determined risks and thus teaches alerting based on a prioritizing using the determined risk. *Supra*, §XII.C.6. For example, Basavapatna discusses examples in which all assets, vulnerabilities, or threats are listed “sorted by aggregate risk metric.” EX1008 (Basavapatna), ¶138; *see also id.*, ¶¶139-40.

168. Accordingly, Basavapatna teaches “[prioritizing/prioritize] each of the detected potential cyber threats based on their respective risk to the protected virtual cloud asset,” as recited.

169. I previously explained the various reasons for combining Basavapatna’s prioritization teachings with Veselov’s teachings regarding snapshot-based security analysis of virtual cloud assets, and that reasoning applies equally here. Applying Basavapatna’s teachings in that context would thus involve prioritizing each of the detected threats based on their respective risk “to the protected virtual cloud asset.” *See supra*, §XII.B.3 (discussing Veselov’s teachings regarding protected virtual cloud assets).

170. It is therefore my opinion that the combination of Veselov and Basavapatna teaches all aspects of claims 2 and 19.

2. Claims 3 and 20

171. Claims 3 and 20 depend from claims 1 and 18, respectively, but recite substantially similar limitations. Claim 3 preamble recites the “method of claim 1, wherein analyzing the snapshot of the protected virtual cloud asset further comprises” and the claim 20 preamble recites the “system of claim 18, wherein analyzing the snapshot of the protected virtual cloud asset further comprises.” Both claims then further recite identical limitations. Given these similarities, I discuss the additional limitations of claims 3 and 20 together in the following sections.

a. Elements 3.1 and 20.1

172. These elements recite “parsing a copy of the snapshot.” As I explain below, Veselov teaches these elements.

173. As an initial matter, Veselov teaches parsing a “copy” of the snapshot. As I previously discussed, Veselov describes several ways of performing snapshot analysis, including obtaining a copy of a snapshot of the target resource (e.g., a VM):

In various non-limiting examples, the scanning service may obtain the snapshot data by executing the snapshot itself, by **requesting another service to capture the snapshot and provide it to the scanning service, by accessing a data store (e.g., the snapshot storage service) containing the snapshot, etc.**; alternatively, the scanning system may obtain access to the snapshot data, such as at a storage

location of the snapshot data, but may not **obtain (i.e., make a copy of)** the snapshot data.

EX1007 (Veslov), 9:16-25 (emphasis added); *see also id.*, Figs. 3A (depicting a copy of snapshot data 308 being brought into test environment 305), 5A (depicting analogous copy 508 being brought into test environment 505); *supra*, §XII.B.5. A POSA would have understood that by bringing the target resource’s snapshot into the scanning system’s testing environment—e.g., obtaining a snapshot from the aforementioned service or from a snapshot storage service—the scanning system was performing its analysis on a “copy” of the snapshot.

174. Veslov also describes “parsing” this copy of the snapshot as part of the analysis in several ways. For example, Veslov explains that “[p]rocessing the snapshot data 146 may require **applying one or more rules, instructions, and/or transformations to the snapshot data** 146.” EX1007 (Veslov), 8:21-23 (emphasis added). Veslov also describes formatting the snapshot data with tags identifying the target resource. *Id.*, 9:32-37 (“Additionally or alternatively, the scanning system may specify parameters related to the format of the resulting snapshot data. For example, the scanning service may cause the snapshot data to include metadata, such as a tag, that identifies the target resource embodied in the snapshot data.”). Veslov explains that its scanning system “may mount the logical volume (or transform the snapshot into a data file by identifying the file system of

the original logical volume and then mount the snapshot in the duplicate logical volume) as a hard drive with data stored thereon” such that “the system may then perform the security assessment on the data at the file system level.” *Id.*, 3:61-67; *see also id.*, 17:4-8 (“The scanning service 510 may **transform the volume image 508 into a format in which the scanning service 510 can read the copied files and other data**; in the illustrated examples, the scanning service 510 does this by mounting the volume image as a virtual storage device.” (emphasis added)); *supra*, §XI.A (¶91) (teachings broadly apply to other embodiments). A POSA would have viewed this type of processing to transform the data into a scannable format as “parsing a copy of the snapshot,” as recited, or at a minimum suggesting such parsing.

175. At a minimum, parsing a copy of a snapshot would have been obvious. As I described earlier (*supra*, §VIII.D.2), the field of Virtual Machine Introspection had developed numerous techniques for parsing VM data as a flat file to detect cybersecurity risks. These VMI techniques, included transforming and interpreting unstructured data such as VM images similar to those taught by Veselov. EX1033 (Garfinkel), 6-7 (discussing using VMI for intrusion detection); EX1034 (Tang), 5643 (antivirus software gets information from guest OS using VMI techniques); EX1035 (Lengyel), 387 (VMI used to observe malware execution); EX1037

(Bauman), 10:2 (using VMI to analyze software running on VM); EX1038 (Suneja), 134 (analyzing different VMI techniques); EX1039 (Payne), 3 (VMI used to assess VM's runtime state); EX1041 (Kornfeld), 3:1-9 (parsing algorithm identifies and organizes computer data); EX1042 (Rubin), 10:45-55, 12:54-60 ("Preferably, immediately after parser 220 performs a reduce operation, it calls analyzer 230 to check for exploits. Analyzer 230 searches for specific patterns of content that indicate an exploit."); EX1043 (Jiang), 12:10 (parsing the virtual image as part of the scanning process to gather information on the running processes); *supra*, §VIII.D.2 (parsing scanned data). It was also well known that VMI parsing techniques were routinely used to gain the same sort of information from snapshots as from live VMs. *Supra*, §VIII.D.2; EX1043 (Jiang), 12:10 (parsing raw memory image of Linux VM to "reconstruct detailed semantic information about each running process"). A POSA, in view of the teachings in Veselov, would have found it obvious to, for example, apply routine VMI techniques (including parsing a snapshot as a data file) as a part of "transform[ing] the volume image" into a format that Veselov's scanning service could read. EX1007 (Veselov), 17:4-8.

176. It is therefore my opinion that Veselov teaches all aspects of elements 3.1 and 20.1.

b. Elements 3.2 and 20.2

177. These elements recite “scanning the parsed copy to detect the potential cyber threats, wherein the potential cyber threats include known and unknown vulnerabilities, and wherein the detection is based on a type of vulnerability.” As I explain below, the combination of Veselov and Basavapatna teaches these elements.

178. Veselov teaches various ways of parsing the copy such that the snapshot is provided in a scannable format. *See, e.g.*, EX1007 (Veselov), 3:62-67 (scanning system may “transform the snapshot into a data file by identifying the file system of the original logical volume and then mount the snapshot in the duplicate logical volume”). Veselov further describes “scanning” this parsed copy. *See, e.g., id.*, 3:62-67 (after transformation, “the system may then perform the security assessment on the data at the file system level”). Similar processing is described with respect to the exemplary method of Fig. 6:

At step 610, the scanning service may format the new logical volume into the corresponding file system and then mount the volume image in the logical volume as a read-only virtual hard disk. Optionally at step 612, the scanning service may attach the mounted volume image to the original virtual machine instance that was imaged. At step 614, the scanning service may perform the security assessment by **reading the files of the mounted volume image**. In some embodiments, **the scanning service may use string or other pattern comparisons to identify files, software packages, application versions, and the**

like, for software installed on the target virtual machine 312. For example, once the file system of the mounted volume image can be read, the scanning service may compare the names of directories and/or files in the image to character patterns representing known directory and/or filenames to identify “signature” directories and files that identify the software application, version, installation attributes, etc.

Id., 18:16-44 (emphasis added); *see also supra*, §XI.A (¶91) (Veselov states that Fig. 6 is a technique involving the environments of Figs. 5A-B). Further, scanning the parsed copy is done for the purpose of checking for cyber threats. Veselov continues:

The scanning service may maintain a database of assessment tasks (e.g., the scan parameter data store 130 of FIG. 1) that are associated with particular software applications and versions thereof. In this manner, the scanning service may identify which assessment tasks to run based on the installed software packages. The scanning service may then, in some embodiments, run the assessment tasks on the raw file data of the mounted volume image **to obtain the assessment results.**

Id., 18:44-53 (emphasis added). As I explained earlier, Veselov’s assessment results, describe, for example, detected cyber threats. *Supra*, §XII.B.5. Accordingly, Veselov teaches the recited limitation.

179. At a minimum, scanning a parsed copy to detect potential cyber threats would have been obvious. As I discussed above, for example, VMI techniques were well known and commonly used to both parse data regarding VMs (and their snapshots) and scan the parsed data for cybersecurity risks. A POSA, in view of Veselov’s disclosure of performing a security scan on a “mounted volume image” whose data had been transformed such that “the file system of the mounted volume image can be read,” would have expected Veselov’s technique to involve these well-known, common VMI techniques, or at least would have found it obvious to use such scanning techniques to read to the parsed/transformed copy as taught by Veselov. EX1007 (Veselov), 18:26-44.

180. Additionally, Veselov and Basavapatna each describe detecting potential cyber threats that “include known and unknown vulnerabilities.” As an initial matter, a POSA would have viewed most if not all scans, including those described in Veselov and Basavapatna, as including a search for “known” vulnerabilities. *Supra*, §VIII.D.1 (discussing common search targets in security scans); *see also, e.g.*, EX1023 (NIST Security Patches), 1-3. As I explained above, “unknow[n]” vulnerabilities were also a common part of standard security scans, which typically looked not simply for malware and other already-known threats, but also system behavior indicative of the presence of new vulnerabilities that had not

yet been identified as a known problem. *Supra*, §VIII.D.1; *see also, e.g.*, EX1025 (NIST Incident Handling), 12.

181. Veselov also specifically describes detecting known vulnerabilities. For example, it discusses searching for known vulnerabilities based on rule packages, including those with defined sets of vulnerabilities:

The present disclosure contemplates implementation of any suitable security assessment, including security assessments that are defined by rules packages, such as **Common Vulnerabilities and Exposures (CVEs)**, Center for Internet Security (CIS) benchmarks, “best practices” packages, static or runtime behavior analysis, host configuration assessments, and the like.

EX1007 (Veselov), 2:35-41 (emphasis added). I understand that the ’798 patent expressly acknowledges that CVEs identify “known vulnerabilities.” *See* EX1001, 4:36-42 (“The external intelligence systems may include common vulnerabilities and exposures (CVE®) databases, reputation services, security systems (providing feeds on discovered threats), and so on. The information provided by the intelligence systems may detect certain **known vulnerabilities identified in, for example, a CVE database.**” (emphasis added)).

182. Veselov also teaches or suggests detection of “unknown” vulnerabilities. For example, it describes analyzing the behavior of computing resources, not merely scanning for known malicious files or other data. *See, e.g.*,

EX1007 (Veselov), 1:46-50 (discussing known systems that, given the large number of factors at play, “analyze the behavior of computing resources to identify vulnerabilities ... and the like”), 2:35-41 (discussing assessments involving “static or runtime behavior analysis”), 16:6-14 (disclosing embodiments that may “generate ‘dummy’ traffic to the assessment virtual machine 312A-B that is similar to the traffic of the target virtual machine” and “may perform destructive testing on the assessment virtual machine 312A-B to verify the system’s health”), claim 5 (reciting “generating a partial assessment result for the first virtual machine instance by analyzing behavior of the second virtual machine instance in response to the simulated network traffic”). This type of behavioral analysis is a typical example of an assessment that looks for unknown vulnerabilities. See EX1025 (NIST Incident Handling), 26-30. At a minimum, these disclosures suggest that the potential cyber threats discussed in Veselov include “unknown” vulnerabilities.

183. Basavapatna also teaches detecting “known and unknown vulnerabilities.” With respect to known vulnerabilities, Basavapatna describes a network monitor that, “when executed by the processor, can identify a standardized vulnerability score for a particular vulnerability in a plurality of **known vulnerabilities**.” EX1008 (Basavapatna), ¶13 (emphasis added); see also *id.*, ¶17 (“At least some vulnerabilities in the plurality of known vulnerabilities may be

associated with at least one in a plurality of known threats, while the particular vulnerability is not associated with any of the known threats.”). Basavapatna also discusses the use of passive countermeasures can protect against unknown vulnerabilities. EX1008 (Basavapatna), ¶22 (“Passive countermeasures generally protect the host until vulnerabilities can have active countermeasures applied, or to protect against unknown vulnerabilities.”).

184. Accordingly, Veselov and Basavapatna independently teach that the “potential cyber threats include known and unknown vulnerabilities,” as recited. While Veselov’s teachings are sufficient on their own to satisfy this limitation, a POSA also had reason to apply Basavapatna’s teachings in the context of virtual cloud systems such as those taught by Veselov, so at a minimum, security scanning for both known and unknown vulnerabilities would have been obvious in view of the combination. I previously discussed reasons for combining Veselov’s teachings with Basavapatna’s teachings regarding alerting detected threats (*supra*, §XII.A), and that reasoning applies equally to all types of threats discussed in Basavapatna, including unknown vulnerabilities. For example, in addition to the reasons I already discussed, both known and unknown vulnerabilities were routinely addressed by security scans (*supra*, §VIII.D.1), and it would have been clear to a POSA that unknown vulnerabilities are just as relevant to security assessments involving virtual

cloud resources, such as those discussed in Veselov, as they are to computer resources generally.

185. Lastly, Veselov also teaches detecting potential cyber threats is “based on a type of vulnerability.” For example, Veselov explains that “various aspects of the security assessment(s) may be configured by user input, non-limiting examples of such aspects including: selection of the assessment target(s); parameters of the security assessment(s), such as tests or tasks to perform, **types of vulnerabilities to evaluate**, and the like.” EX1007 (Veselov), 14:41-46 (emphasis added). As I previously explained, a POSA would have recognized that this teaching of the assessment being configurable to detect certain types of vulnerabilities broadly applies to the various embodiments in Veselov. *Supra*, §XI.A (¶91). Veselov also explains that the security assessment can apply rules packages as CVEs. *See id.*, 2:35-41 (“The present disclosure contemplates implementation of any suitable security assessment, including security assessments that are defined by rules packages, such as Common Vulnerabilities and Exposures (CVEs), Center for Internet Security (CIS) benchmarks, ‘best practices’ packages, static or runtime behavior analysis, host configuration assessments, and the like.”), 7:16-22; *see also supra*, §XII.B.5. A POSA would have understood that CVEs in general are a “type of vulnerability” since CVEs are a group of security vulnerabilities that were

routinely organized in a central repository and evaluated as part of a distinct detection step in security assessments. *See supra*, §VIII.D.1-2 (¶¶50, 59, 61); EX1075 (Mohanty), 4:46-64 (vulnerability scanner performs CVE assessment); EX1078 (Hufsmith), ¶¶33, 84 (scanner detects “a variety of different types of vulnerabilities” such as CVEs and CWEs). A POSA also would have understood each individual CVE to be a “type of vulnerability” since each detected CVE was typically a distinctly categorized vulnerability associated with a particular identifier and description. *See, e.g.*, EX1075 (Mohanty), 3:43-54 (discussing CVE identifier); EX1078 (Hufsmith), ¶113 (same).

186. It is therefore my opinion that Veselov alone, or in combination with Basavapatna, teaches all aspects of elements 3.2 and 20.2.

187. Accordingly, in my opinion, claims 3 and 20 were obvious.

3. Claims 4 and 21

188. Claims 4 and 21 depend from claims 3 and 20, respectively, but recite substantially similar limitations. Claim 4 preamble recites the “method of claim 3, wherein scanning the parsed copy further comprises any one of” and the claim 21 preamble recites the “system of claim 20, wherein scanning the parsed copy further comprises any one of.” The claims then list the same four steps, which I will label 4.1 to 4.4 (or 21.1 to 21.4) to simplify further discussion. Given these similarities in

the limitations of claims 4 and 21, I discuss them together below. These limitations are listed below with the labels for claim 4:

[4.1] checking configuration files of applications and operating system installed in the protected virtual cloud asset;

[4.2] verifying access times to files by the operating system installed in the operating system;

[4.3] analyzing system logs to deduce what applications and modules executed in the protected virtual cloud asset; or

[4.4] analyzing machine memory stored in the snapshot to deduce what applications and modules executed in the protected virtual cloud asset.

189. Furthermore, the preambles recite that scanning the parsed copy further comprises “any one of” these steps, which I understand to mean that the additional requirements of claims 4 and 21 are satisfied if even one of these steps is met. As I explain below, Veselov further teaches elements 4.1 and 21.1 and elements 4.4 and 21.4, either one of which is sufficient to satisfy claims 4 and 21. It is therefore my opinion that claims 4 and 21 were obvious over Veselov and Basavapatna.

a. Elements 4.1 and 21.1

190. These elements recite “checking configuration files of applications and operating system installed in the protected virtual cloud asset.” As I explain below, the combination of Veselov and Basavapatna teaches these elements.

191. Veselov explains that the snapshot—a copy of which is parsed as I explained when discussing claims 3 and 20 (*supra*, §XII.C.2)—includes configuration files associated with the VM. EX1007 (Veselov), Abstract (“To reproduce a target virtual machine, the snapshot may include a configuration used to instantiate the target virtual machine....”). Discussing virtualization generally, Veselov also notes that computing devices may be associated with particular operating system configurations and software applications, and that security services may seek to identify “bad configurations.” *Id.*, 1:22-34 (“In some scenarios, various computing devices may be associated with different combinations of operating systems or operating system configurations, virtualized hardware resources, and software applications to enable a computing device to provide different desired functionalities, or to provide similar functionalities more efficiently.”).

192. Veselov also teaches that the analyzed snapshot may contain configurations of installed software packages and operating system information (i.e., “configuration files of applications and operating systems installed in the protected virtual machine” as recited). For example, Veselov states:

The “state” of the target resource represented by the snapshot data may include any properties of the target resource instantiated in the computing resource service provider 104 that are relevant to the security assessment. Non-limiting examples of such properties

include: properties of logical volume(s) containing the target resource; properties of physical storage devices allocated to the logical volumes; file system and file information; **operating system information**; system and software patches applied; configurations of network protocols and services; properties of connections to other computing resources; permission settings; **software packages installed, and configurations thereof**; system log properties; and the like.

Id., 9:55-67 (emphasis added); *see also id.*, 7:16-22 (“In one example of reproducing a virtual machine for a software application-level CVEs assessment, **the snapshot may only need to contain** a template, such as a virtual machine image, from which the virtual machine is instantiated, and a copy of the virtual machine package manager database, which **lists all software packages installed on the virtual machine and their configurations**.” (emphasis added)). Similarly, discussing Fig. 6, Veselov states:

In some embodiments, the scanning service may use string or other pattern comparisons to **identify files, software packages, application versions, and the like**, for software installed on the target virtual machine 312. For example, once the file system of the mounted volume image can be read, the scanning service may compare the names of directories and/or files in the image to character patterns representing known directory and/or filenames to identify “signature” directories and **files that identify the software application, version,**

installation attributes, etc. The scanning service may maintain a database of such character patterns. Further, the scanning service may maintain a database of **assessment tasks** (e.g., the scan parameter data store 130 of FIG. 1) that are **associated with particular software applications and versions thereof**. In this manner, the scanning service may identify which assessment tasks to run based on the installed software packages.

Id., 18:32-50.

193. Veselov further teaches analyzing such configuration files as part of the scan (i.e., scanning the parsed copy further comprises “checking” the recited configuration files, as recited). Veselov generally teaches that the “security assessment may include an evaluation of a virtual machine configuration....” *Id.*, 21:61-22:6; *see also id.*, claim 9 (security assessment may include analysis of VM configuration). Veselov also explains that security assessments seek to identify “bad configurations.” *Id.*, 1:43-50 (“The large numbers of customers, end users, virtual machine configurations, software packages, and hardware computing devices invite security issues to arise. The service provider may thus provide or enable security assessment services that analyze the behavior of computing resources to identify vulnerabilities, bad configurations, and the like.”); *see also id.*, 2:35-41 (rule packages applied by scanning service may include “host configuration assessments”); 18:32-44 (describing scanning a parsed snapshot, searching for “files

that identify the software application, version, installation attributes, etc.,” then determining “which assessment tasks to run based on the installed software packages”). In this context, it would have been clear to a POSA that Veselov’s assessment of the snapshot—which includes parsing copy of a snapshot containing “operating system information” and configurations of the “software packages installed” and then scanning that parsed copy—includes “checking configuration files of applications and operating systems installed in the protected virtual machine” as recited.

194. This limitation is also taught further in view of Basavapatna. Basavapatna teaches checking configuration files of applications as part of determining risks associated with detected vulnerabilities. EX1008 (Basavapatna), ¶¶37 (asset configuration data includes “configuration of the software running on the asset”), 53 (vulnerability definition data includes applicability data such as “application settings”), 60 (vulnerability detection data generated by “comparing asset configuration data 207 and vulnerability definition data 205”), 79 (determining threat-centric risk metric involves comparing “applicability data for a given threat to configuration data for a given asset”), 105 (determining vulnerability-centric risk metric involves “analyzing the vulnerability detection data 206 and/or asset configuration data 207” to determine whether assess possesses “the corresponding

vulnerability or configurations evidencing the vulnerability”). Basavapatna also teaches checking configuration files of an operating system installed on the asset. *Id.*, ¶¶37 (asset configurations include “specific versions of operating systems (e.g., Windows Vista™)”), 44 (applicability data can specify that threat only attacks “particular operating systems or particular software products installed on an asset”), 53 (same), 60 (partial match when vulnerable operating system “and the operating system running on the asset” are in same family), 78 (“For example, if a particular operating system has a known vulnerability that is exploited by a threat, and an asset is running that operating system, the threat is applicable to the asset.”), 80 (partial match), 104 (identifying vulnerability where “a particular operating system has a known vulnerability, and an asset is running that operating system”). For example, the system may identify that the installed operating system is an exact or partial match for a given vulnerability by checking whether the installed operating system has “different versions, patches, service packs, etc.” (*id.*, ¶60), which is information commonly found in configuration files. As I previously explained, a POSA had reason to apply Basavapatna’s threat-detection and risk-determination teachings in combination with Veselov’s teachings regarding snapshot-based assessments of protected virtual cloud assets. *Supra*, §XII.A. The combination of Veselov and

Basavapatna thus teaches “checking configuration files of applications and an operating system installed in the protected virtual machine,” as recited.

195. It is therefore my opinion that Veselov alone, or in combination with Basavapatna, teaches all aspects of elements 4.1 and 21.1.

b. Elements 4.4 and 21.4

196. These elements recite “analyzing machine memory stored in the snapshot to deduce what applications and modules executed in the protected virtual cloud asset.” As I explain below, the combination of Veselov and Basavapatna teaches these elements.

197. First, Veselov teaches “analyzing machine memory stored in the snapshot.” It explains that the snapshot of a VM, or elements thereof, can include the state of that virtual resource’s memory (i.e., “machine memory stored in the snapshot,” as recited). For example, it states:

The snapshot may include all of the data needed to recreate the state of the computing resource within a duplicate, virtual computing resource. For example, the target computing resource may be an instance of a virtual machine implemented within block-level storage device resources allocated to a logical volume. **The snapshot may be a copy of the state of memory, the state of any devices (virtual or physical) allocated to the resource, block-level image of the entire logical volume**; or, the snapshot may be an image of only a portion of the logical volume containing the data required to embody an exact

copy of the virtual machine instance; or the snapshot may simply be a copy of certain files of the target computing resource, such as a database of software packages installed in a virtual machine.

EX1007 (Veselov), 3:27-40. In discussing elements 1.2 and 1.3 (*supra*, §§XII.B.4-5) and claims 3 and 20 (*supra*, §§XII.C.2.a-b), I also explained how Veselov teaches analyzing a snapshot by, for example, mounting an image as a logical volume (e.g., as a virtual disk) and reading files of this mounted image or by reading the snapshot as a data file. Veselov's security assessment, which can involve parsing of a copy of the snapshot, thus includes "analyzing machine memory stored in the snapshot."

198. The combination Veselov and Basavapatna also teaches performing this analysis "to deduce what applications and modules executed in the protected virtual cloud asset," as recited. As an initial matter, Veselov teaches that the analysis looks for security issues associated with software packages in the protected virtual cloud asset. *See, e.g.*, EX1007 (Veselov), 7:15-22 (snapshot can include a "copy of the virtual machine package manager database, which lists all software packages installed on the virtual machine and their configurations"), 9:55-67 (evaluated snapshot properties can include "software packages installed" on the target resource), 18:44-50 (Veselov's "scanning service may identify which assessment tasks to run based on the installed software packages"); *see also supra*, §XII.C.3.a. A POSA would have understood these software packages as encompassing, or at

least suggesting, both “applications” and “modules” executed in the virtual cloud asset since applications and modules were well-known software elements running on typical VMs. I also previously explained that Veselov teaches detecting cyber threats associated with such software packages by, for example, performing a CVE assessment. *Supra*, §§XII.B.5, XII.C.2.b; EX1007 (Veselov), 2:35-41, 7:16-22, 14:41-65.

199. Basavapatna further teaches deducing what software (which encompasses “applications and modules” as I explained above) “executed” in the scanned asset in the context of an analogous security scan. For example, Basavapatna teaches that determining the previously discussed risk metrics involves determining whether the assessed vulnerability (or a given security problem targeted by the assessed threat) is applicable to the protected asset. EX1008 (Basavapatna), ¶¶14-16, 53, 60, 78-81, 104, claims 22, 27. Consistent with common knowledge in the field, Basavapatna also teaches, or at least suggests, that determining risk/applicability includes determining whether the asset is running vulnerable/targeted software. *See id.*, ¶¶60, (asset vulnerable when it has an installed “or running version of a particular software product”), 78 (threat applicable “when the asset is running software” exploitable by threat), 104 (applicable when “asset is running software” associated with vulnerability); *see also id.*, ¶¶37 (asset

configuration data includes “particular software products executing on the system”), 79-80 (comparing configuration data to applicability data); *supra*, §VIII.D.3 (¶68) (citing EX1063 (Hibbert), ¶¶80, 100-12) (software usage was well-known risk factor). I previously explained why a POSA had reason to apply Basavapatna’s risk assessment and prioritization teachings to Veselov (*supra*, §XII.A), and those reasons apply equally to the above teachings of Basavapatna since they are part of Basavapatna’s risk/prioritization analysis.

200. It is therefore my opinion that the combination of Veselov and Basavapatna teaches all aspects of elements 4.4 and 21.4.

4. Claims 5 and 22

201. Claims 5 and 22 depend from claims 1 and 18, respectively, but recite substantially similar limitations. The preamble of claim 5 recites the “method of claim 1, further comprising...” and the preamble of claim 22 recites the “system of claim 18, wherein the instructions further cause the at least one hardware processor to....” Both claims then recite analogous limitations. Given these similarities, I discuss the analogous limitations of claims 5 and 22 together in the following sections:

a. Elements 5.1 and 22.1

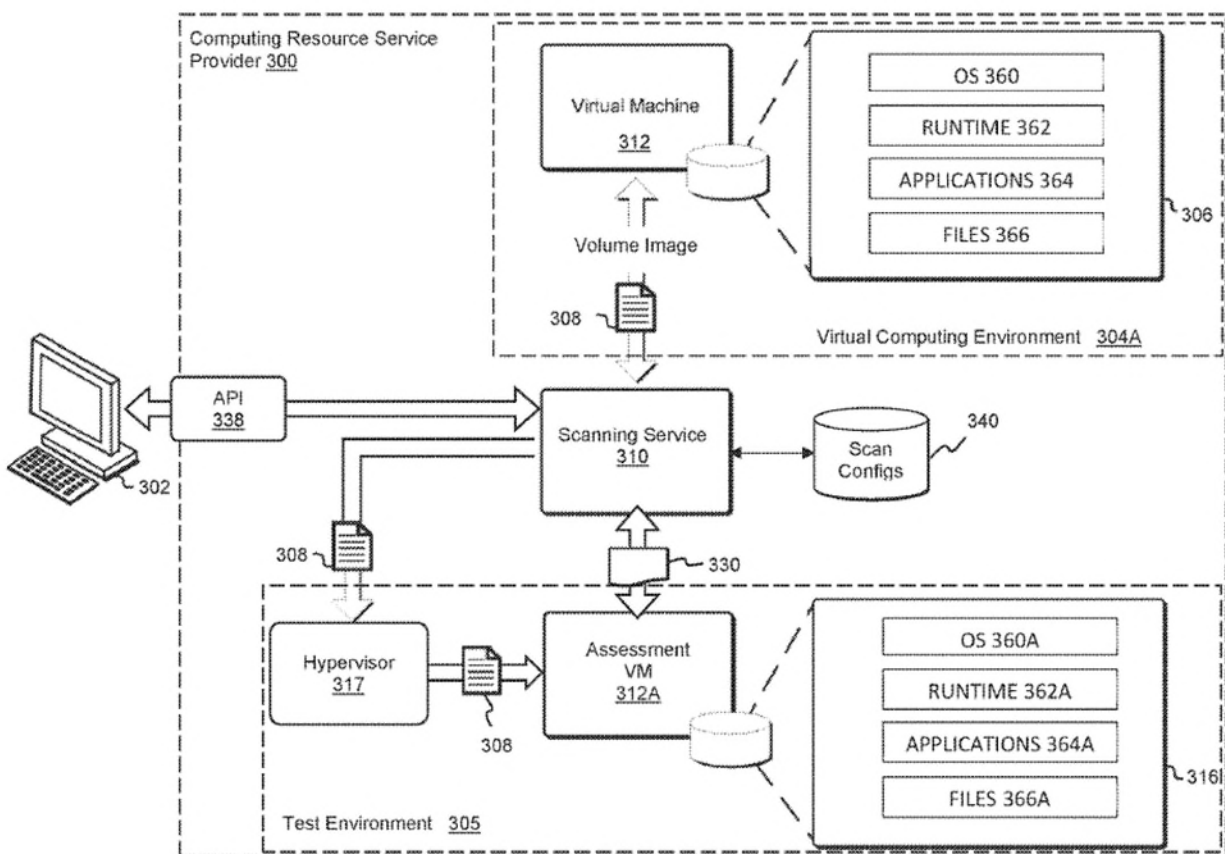
202. These elements recite “instantiating an instance copy of the protected virtual cloud asset from the snapshot.” As I explain below, Veselov teaches these elements.

203. Before turning to Veselov’s specific teachings regarding instantiation, I note that, as previously discussed, Veselov teaches several different ways of performing its scans, including analysis of a parsed copy of the snapshot data (as I discussed with respect to claims 3 and 20) and analysis of an instantiated VM. *Supra*, §XI.A (Veselov summary). Veselov also teaches performing multiple scans involving the same snapshot and explains that disclosures in the context of particular embodiments may be combined. *E.g.*, EX1007 (Veselov), 4:1-18 (after scan, the “scanning system may then delete the snapshot or alternatively store the snapshot in the account of the user so that the user can perform additional analysis on the snapshot”), 14:37-65 (“In various embodiments, an owner or administrator of the target computing resources to be assessed may configure one or more security assessments....”), 24:54-25:2 (“Furthermore, it is to be understood that particular features, structures, or characteristics described may be combined in various ways in one or more embodiments.”). Veselov thus teaches, or at a minimum suggests, performing multiple scans involving the same snapshot, which would include both the snapshot-parsing techniques I discussed above (*supra*, §XII.C.3) and the VM-

instantiation techniques I discuss below. A POSA would have had further reason, in some circumstances, to perform additional scans using the instantiation approach discussed in Veselov because the POSA would have recognized that there are certain types of security issues that the instantiation approach would be more likely to detect (e.g., where detection of a security problem involves monitoring live activity from a running instance of a VM, as Veselov discusses in certain examples). See EX1007 (Veselov), 15:41-16:3. I discuss such examples in more detail below in §XII.C.4.b.

204. Veselov teaches “instantiating an instance copy of the protected virtual cloud asset from the snapshot.” For example, it describes a security scanner that operates on a reproduced target VM. As Veselov explains, the snapshot data can be applied “to produce a duplicate implementation of the target resource that can be scanned by the scanning service,” including “launching a duplicate virtual machine instance in an allocated logical volume.” EX1007 (Veselov), 10:1-16; *see also id.*, Abstract (“To reproduce a target virtual machine, the snapshot may include a configuration used to instantiate the target virtual machine; the scanning service may implement a duplicate virtual machine that is instantiated with the same configuration.”).

205. Veselov also depicts such instantiation of a copy in Figure 3A:



Id., Fig. 3A; see also *id.*, Fig. 3B (illustrating an analogous assessment VM 312B).

Veselov explains that “the volume image 308 may be installed in the second logical volume 316 so that the assessment virtual machine 312A-B may be **instantiated with the same configuration as the target instance** of the original virtual machine 312 had at the time the volume image was captured.” *Id.*, 13:53-58 (emphasis added); see also *id.*, 3:54-61 (“In one embodiment, the scanning system may set up and launch an active virtual machine instance in the allocated computing resources, the active virtual machine instance having the same configuration as the original virtual machine had at the time the snapshot was captured; the system may then

perform the security assessment on the active virtual machine instance in a static or runtime environment.”), 12:1-20 (describing instantiating a new instance copy of the target virtual machine).

206. It is therefore my opinion that Veselov teaches all aspects of elements 5.1 and 22.1.

b. Elements 5.2 and 22.2

207. These elements recite “monitoring all activity performed by the instance copy of the protected virtual cloud asset.” As described below, Veselov teaches this limitation.

208. Veselov teaches “monitoring all activity performed by the instance copy of the protected virtual cloud asset.” Discussing the VM 312 that I mentioned in the previous section (i.e., the “instance copy”), Veselov explains:

In some embodiments, the scanning service 310 can cause processing activity to occur within the assessment virtual machine 312A-B in order to apply tasks of the security assessment to real-time dynamic or static data, network connections, communications to computing resources, and other computational activities of the assessment virtual machine 312A-B. In order to stimulate activity that mirrors the activity of the target virtual machine 312, the snapshot data may need to include some such activity of the virtual machine 312. The scanning service 310 may therefore cause the snapshot to be captured for a capture period that lasts a predetermined duration; during the

capture period, the capturing device may monitor all relevant communications, data processing, and other activities of the virtual machine 312, storing in the snapshot data a real-time snapshot describing the activity and connections of the virtual machine 312.

EX1007 (Veselov), 15:41-57. Veselov teaches that once an instance copy has been instantiated, the scanning service may “intercept traffic to the target virtual machine 312 and duplicate the intercepted traffic to the assessment virtual machine 312A-B in order to stimulate activity in the assessment virtual machine 312A-B.” *Id.*, 15:60-16:3; *see also id.*, 16:3-10 (discussing optional interception of traffic during snapshot generation), claim 5 (“generating a partial assessment result for the first virtual machine instance by **analyzing behavior of the second virtual machine instance in response to the simulated network traffic**” (emphasis added)).

209. It is therefore my opinion that Veselov teaches all aspects of elements 5.2 and 22.2.

210. Accordingly, in my opinion, claims 5 and 22 were obvious.

5. Claims 7 and 24

211. Claims 7 and 24 depend from claims 3 and 20, respectively, but recite substantially similar limitations. The preamble of claim 7 recites the “method of claim 3, wherein scanning the parsed copy further comprises...” and the preamble of claim 24 recites the “system of claim 20, wherein scanning the parsed copy further

comprises....” Both claims then recite analogous limitations. Furthermore, the elements are recited using “or” language, so I understand that any one of them is sufficient to satisfy claims 7 and 24. As I explain below, Veselov alone, or in combination with Basavapatna, teaches at least elements 7.1/24.1, 7.2/24.2, and 7.4/24.4. It is therefore my opinion that claims 7 and 24 were obvious.

a. Elements 7.1 and 24.1

212. These elements recite “matching at least one installed application list against a list of vulnerable applications.” As I explain below, Veselov alone, or in combination with Basavapatna, teaches these elements.

213. Veselov teaches, or at least suggests, that scanning the parsed copy further comprises “matching at least one installed application list against a list of vulnerable applications.” For example, the parsed snapshot data scanned as part of the security assessment I discussed above (§XII.C.2) represents the “state” of the target resource including any of its properties that are relevant to the security assessment, including “system and software patches applied” and “software packages installed.” EX1007 (Veselov), 9:55-67. In another example, Veselov teaches that the “scanning service may use string or other pattern comparisons to identify files, software packages, application versions, and the like,” that are installed on the target VM. *Id.*, 18:32-38; *see also supra*, §XI.A (¶91) (teachings applicable to other embodiments). Veselov explains that the scanning service may

compare “the names of directories and/or files in the image to character patterns representing known directory and/or filenames” to identify the applications installed on the VM. *Id.*, 18:38-48. The scanning service may then run certain assessment tasks “based on the installed software packages.” *Id.*, 18:45-53, 22:35-46 (“[I]dentify, from the plurality of data files, a plurality of file directories stored on the virtual storage device; compare the plurality of file directories to each of one or more character patterns to identify a software application installed on the first logical volume; determine a plurality of tasks in the security assessment that are associated with the software application....”), claim 12 (same). Veselov also teaches that CVE assessments may be based on such lists of installed application. *Id.*, 7:6-22 (snapshot data “may contain copies of essential files of the target computing resource, the files being required to create a reproduction of the target computing resource in the state the resource was in at the time the snapshot was created.... In one example of reproducing a **virtual machine for a software application-level CVEs assessment**, the snapshot may only need to contain a template, such as a virtual machine image, from which the virtual machine is instantiated, and a copy of the virtual machine package manager database, **which lists all software packages installed on the virtual machine and their configurations**” (emphasis added)). Veselov thus

teaches that a basis for the threat-detection analysis may be a list of applications installed on the asset.

214. Veselov further teaches, or at least suggests, matching such an application list “against a list of vulnerable applications,” as recited. Veselov teaches that its assessment applies rule packages such as “Common Vulnerabilities and Exposures (CVEs) ... and the like.” *Id.*, 2:35-41, 14:58-65 (scanning service may use CVE assessment). A POSA would have understood that conventional rule-based security assessments commonly involved CVE matching or comparable techniques for matching the target asset’s applications (e.g., a list of the applications/versions installed on a system) against a list of vulnerable applications. *Supra*, §VIII.D.1 (discussing CVEs); EX1023 (NIST Security Patches), 1-3 (CVE provides security community with “comprehensive list of publicly known vulnerabilities”); EX1076 (Milner), 2:24-42 (security tool may “compare the components of the software package to the black list,” which “can include a list of archival files that are known to be insecure, such as known insecure packages/subpackages referenced in a Common Vulnerabilities and Exposures (CVE) list”), 2:42-45 (blacklist can include file details such as name/version); EX1077 (Khalid), 2:45-51 (system compares object’s characteristics “with those stored on one or more cyber-security databases, such as, for example, a Common

Vulnerabilities and Exposures (CVEs) database”), 8:1-30 (discussing CVE database and “content searching/matching”); EX1084 (Czarny), 4:25-33, 6:25-38 (discussing the use of CVE databases as list of known vulnerabilities); *see also* EX1044 (Giakouminakis), 4:22-35 (discussing configuring a vulnerability database to “store a record of known vulnerabilities for various types of known software resources and hardware resources ... to identify vulnerabilities in the computing system 104 and/or the VMs”), 13:59-14:7 (security tool examines VMs to “identify the software resources” such as “type of software installed, e.g. OS and application programs, version of the software installed, configuration of the software installed” and then “compare the identified details of the software resources ... to the vulnerability database 108 in order to identify vulnerabilities” in VMs). Veselov’s teachings are consistent with these commonplace techniques since, for example, Veselov explains that a CVE assessment may be performed at “a software application-level.” EX1007 (Veselov), 7:16-22. Thus, in my opinion, a POSA would have understood that the use of a CVE rules package in Veselov teaches, or at least suggests, the routine technique of matching a list of the installed applications against a list of vulnerable applications. I also note that the ’798 patent acknowledges that CVE databases include known vulnerabilities. EX1001, 4:40-42 (system may detect “known vulnerabilities identified in, for example, a CVE database”).

215. While this limitation is taught or suggested by Veselov alone, it is further taught by Basavapatna. As I discussed previously, Basavapatna’s process for determining vulnerability-centric risk metrics includes comparing “applicability data” with asset configuration data to “detect vulnerabilities on particular assets and generate corresponding vulnerability detection data and generate corresponding vulnerability detection data 206 reporting the discovery of such vulnerabilities.” EX1008 (Basavapatna), ¶53; *see also id.*, ¶¶104-06, Fig. 3A; *supra*, §XII.B.5. Basavapatna describes similar use of applicability data and vulnerability detection data for generating threat-centric risk metrics. EX1008 (Basavapatna), ¶¶76-85, Fig. 3B; *supra*, §XII.B.5. The asset configuration data describes the asset’s hardware/software configuration, including “the software running on the asset” (which would also be installed applications since the application would be “running on the asset” if it were not installed on the asset). *Id.*, ¶37; *see also id.*, ¶60 (“In another example, identifying that a particular asset has an **installed** or running version of a particular software product can be interpreted to mean that the asset is vulnerable to particular vulnerabilities known to exist on or that are attributable to the particular version of the product.” (emphasis added)). This also teaches, or at least suggests, matching using an installed application “list” since Basavapatna is describing analysis of the set of installed applications. The applicability data

includes, or at least suggests, a known “list of vulnerable applications” as recited since it can “specify certain operating systems, patch versions, peripheral devices, **software application versions**, application settings, plug-ins, etc. **known to introduce the vulnerability onto the asset.**” *Id.*, ¶53 (emphasis added); *see also id.*, ¶44 (applicability data specifies that “the threat only attacks particular operating systems or particular software products installed on an asset” or particular versions/configurations of such products). Accordingly, Basavapatna teaches “matching at least one installed application list against a list of vulnerable applications,” as recited. *Id.*, ¶60 (detecting vulnerabilities based on whether vulnerability definition data exactly or partially matches asset configuration).

216. It is therefore my opinion that Veselov teaches all aspects of elements 7.1 and 24.1.

b. Elements 7.2 and 24.2

217. These elements, which recite “matching at least one application file against a database of files of vulnerable applications.” As I previously explained, Veselov and Basavapatna each teach identifying vulnerabilities by comparing a list of installed applications against a list of vulnerable applications such as CVE repositories/lists (*supra*, §XII.C.5.a) and that analysis also applies here. Below I provide additional explanation of why Veselov alone, or in combination with Basavapatna, teaches elements 7.2 and 24.2.

218. Veselov’s discussion of matching processes such as CVE assessments, viewed alone or in combination with Basavapatna’s similar discussion, teaches or suggests matching at least one “application file against a database of files of vulnerable applications.” Veselov teaches performing “a software application-level CVEs assessment” and teaches that the snapshot being analyzed may include application files. *See* EX1007 (Veselov), 2:35-41, 3:20-40 (snapshot may be image of entire logical volume or “a copy of certain files of the target computing resource, such as a database of software packages installed in a virtual machine”), 7:16-22, 9:42-67 (snapshot may be entire logical volume or may include relevant properties for analysis such as “file system and file information” and “software packages installed”), 17:9-29 (scanning service may mount volume image to access “software applications”). Basavapatna similarly describes matching installed applications with applicability data that lists known vulnerabilities such as applications and application settings. EX1008 (Basavapatna), ¶¶44 (applicability data identifies vulnerable software and software configurations), 53 (same), 60 (discussing exact and partial matches), 79-80 (same). A POSA would have known that the types of software-vulnerability detection described in Veselov and Basavapatna routinely involved matching the types of application files in Veselov’s snapshot against applications files in a known list/database of vulnerable applications. *See, e.g.,*

EX1017 (Rosendahl), 7:58-64 (“Each entry in the threat database 132 may include ... a computer-readable signature, i.e., a means of allowing a computer to identify the CVE within the actual software source.”); EX1023 (NIST Security Patches), 3-4 (discussing use of vulnerability databases/repositories). For example, the CVE analysis described in Veselov and Basavapatna routinely included a binary or a hash-based comparison of installed application files with a database of files of vulnerable applications such as a CVE database. EX1047 (Bourquin), 1-5, 7 (discussing known binary comparison tools such as BinDiff for identifying malware); EX1068 (Epstein), 14:58-15:18 (discussing “byte-by-byte comparison”); EX1076 (Milner), 2:42-67 (blacklist can include hashes of files); EX1084 (Czarny), 3:56-4:47 (discussing matching hash of software product with hash of known vulnerability), 11:27-47 (discussing performing binary comparison with binary of known vulnerability). A POSA thus would have understood the references to teach or at least suggest using these commonplace binary/hash matching techniques to perform the matching analysis discussed in the references. Accordingly, at a minimum it would have been obvious for the snapshot-based analysis taught by the references to include “matching at least one application file against a database of files of vulnerable applications,” as recited.

219. It is therefore my opinion that Veselov teaches all aspects of elements 7.2 and 24.2.

c. Elements 7.4 and 24.4

220. Veselov teaches these elements, which recite “reading installation logs of package managers.” As I discussed above, Veselov and Basavapatna teach matching a list of installed applications against a list of vulnerable applications by, for example, performing a CVE assessment. *Supra*, §XII.C.5.a. Veselov further explains that its application-level CVE assessment may be based on a package manager’s list of installed applications, which at least suggests “reading installation logs of package managers” as recited. EX1007 (Veselov), 7:6-22 (snapshot includes “copy of the virtual machine package manager database, **which lists all software packages installed on the virtual machine and their configurations**” (emphasis added)). At a minimum, this would have been obvious since a POSA would have understood that package managers routinely maintained a log of installed applications, and reading such logs was a common, simple way of obtaining that information. *See, e.g.*, EX1117 (GeekStuff), 2 (/var/log/yum.log contains “information that are logged when a package is installed using yum” and /var/log/dpkg.log contains “information that are logged when a package is installed

or removed using dpkg command”);⁴⁰ EX1118 (LinuxMoz), 5 (describing how to use yum package manager to list installed applications (e.g., “yum list installed”).⁴¹

221. It is therefore my opinion that Veselov teaches all aspects of elements 7.4 and 24.4.

222. Accordingly, in my opinion, claims 7 and 24 were obvious.

6. Claims 8 and 25

223. Claims 8 and 25 depend from claims 1 and 18, respectively, and further recite “wherein the protected virtual cloud asset includes any one of: a virtual

⁴⁰ EX1117 is an archived copy of “20 Log Files that are located under /var/log Directory,” archived on October 13, 2018, by the Internet Archive’s “Wayback Machine” at <https://web.archive.org/web/20181013220819/https://www.thegeekstuff.com/2011/08/linux-var-log-files/>.

⁴¹ EX1118 is an archived copy of “Linux YUM Tutorial,” archived on November 10, 2017, by the Internet Archive’s “Wayback Machine” at <https://web.archive.org/web/20171110120546/https://linuxmoz.com/linux-yum-tutorial/>.

machine, a software container, or a micro-service.” Given the requirement that the protected virtual cloud asset include “any one of” the listed elements, I understand that the additional requirement of claims 8 and 25 is satisfied if even one of the listed elements is met. As I explain below, Veselov teaches that the protected virtual cloud asset includes at least “a virtual machine” and “a container” as recited.

224. Veselov teaches “systems for performing a security assessment of a target computing resource, **such as a virtual machine** or an instance of a virtual machine.” EX1007 (Veselov), Abstract; *see also id.*, 3:30-32 (“the target computing resource may be an instance of a virtual machine implemented within block-level storage device resources allocated to a logical volume.”). I also explained why Veselov teaches that the protected virtual cloud asset is a VM when discussing element 1.1. *Supra*, §XII.B.3. Veselov also teaches that the protected asset may be a software container or that the protected VM may include software containers. *See* EX1007 (Veselov), 1:34-40 (“Virtual machines may themselves be partitioned into multiple isolated virtual systems, called ‘**containers**.’ The virtual machine controls allocation of resources such as processing power and memory, and **each container has its own process and network space in which the container may, for example, execute software programs**.” (emphasis added)), 2:44-51 (“A security assessment may involve sending requests and/or other information to one or more targets, such

as a virtual machine, a group of interconnected virtual machines, **a container of a virtual machine**, a software application, a service, a server, a hard disk drive or logical volume, etc.” (emphasis added)).

225. It is therefore my opinion that claims 8 and 25 were obvious.

7. Claims 9 and 26

226. Veselov teaches the additional limitations of claims 9 and 26, which depend from claims 1 and 18, respectively, and further recite “wherein taking the snapshot is performed at a predefined schedule.” Veselov explains that the user may authorize periodic assessments of a target resource, which causes the scanning system to “obtain and analyze snapshots on a scheduled basis.” EX1007 (Veselov), 4:11-18; *see also id.*, 8:62-9:5 (scan may be initiated by a scheduler at a “scheduled time”), 14:41-53 (user may configure “assessment scheduling, such as the time to execute the assessment and/or whether to repeat the assessment according to an interval”), 15:15-23 (same), 21:23-32 (same), 23:48-56 (same), claim 4 (receiving “request to perform periodic security assessments”). Moreover, as I explained above, Veselov teaches that the scanning service may take a snapshot as part of obtaining a snapshot (e.g., by executing the snapshot itself or requesting that another service capture the snapshot). *Id.*, 8:62-9:25, 3:22-27 (snapshot taken in conjunction with assessment); *supra*, §XII.B.3. Veselov’s discussion of *obtaining* a snapshot

according to predefined schedule thus teaches *taking* the snapshot at a predefined schedule as recited.

227. It is therefore my opinion that claims 9 and 26 were obvious.

8. Claims 10 and 27

228. Veselov teaches claims 10 and 27, which depend from claims 1 and 18, respectively, and further recite “wherein taking the snapshot is performed based on receiving a request to take the snapshot.” Veselov teaches this in several ways. First, Veselov explains the assessment “may be initiated by receipt of a request.” EX1007 (Veselov), 6:30-31; *see also id.*, 6:30-43 (“For example, a user device 102 may transmit a request to the scanning service 110 for a particular security assessment of one or more instances of a virtual machine 112.”), 14:41-52 (user may request assessment be performed based on “trigger events”), Fig. 1. Veselov explains that the requested assessment may involve taking a snapshot (*supra*, §XII.B.3), so Veselov’s disclosure of receiving a request to perform such an assessment at least suggests that the taking of the snapshot is performed based on receiving a request to take the snapshot. Second, as I discussed above, Veselov teaches that the scanning service can take the snapshot itself or obtain the snapshot “by requesting another service to capture the snapshot and provide it to the scanning service.” *Id.*, 9:14-25; *see also id.*, 6:38-43 (scanning service may “determine the corresponding virtualization layer 144 for the virtual machine 112 identified by the request and

transmit a command to the virtualization layer 144 to provide snapshot data”), 7:24-31 (separate “snapshot service” may be “used to generate snapshot data ... using computing resources of the server computer”), 8:60-9:14 (obtaining snapshot from service); *supra*, §XII.B.3. In other words, Veselov also teaches that the snapshot may be taken based on another service receiving a request from the scanning service to take the snapshot.

229. It is therefore my opinion that claims 10 and 27 were obvious.

9. Claims 11 and 28

230. Veselov teaches claims 11 and 28, which depend from claims 1 and 18, respectively, and further recite “wherein taking the snapshot is performed based on detection of a network event or an abnormal event.” Veselov teaches that the user may identify “trigger events that cause the assessment to begin.” EX1007 (Veselov), 14:41-57. As I previously explained, Veselov teaches that the assessment may include taking the snapshot (*supra*, §XII.B.3) so a POSA would have understood Veselov as teaching, or at least suggesting, that the taking of the snapshot is performed “based on” the trigger event. Moreover, taking the snapshot based on detection of a “network event or an abnormal event” would have been obvious in view of Veselov’s teachings of trigger events since trigger events for initiating security assessments routinely included abnormal events or network events that indicate a potential security problem. *See, e.g.*, EX1036 (Wysopal), 14:51-15:13

(discussing security scans triggered by events such as “changes in the external environment (e.g., threat space, business intelligence, detected attacks)”); EX1064 (Lutas), Abstract, ¶¶31 (forensic tools include “snapshot-generating tools”), 42-44 (performing event-triggered VM introspection when there is “an attempt modify a function of OS ..., an attempt by one software entity to inject code into another software entity, an attempt to launch a software component which is not digitally signed, and an attempt to circumvent a digital signature verification, among others ... establishing a new network connection, [and] receiving a network packet”), 63 (“[W]hen event indicator shows that a code injection has occurred, server 14 may request an analysis using the ‘application snapshot’ and/or Volatility® memory forensic tools.”). The ’798 patent confirms that these common types of trigger events are “abnormal events or network events” as recited. EX1001, 7:19-25 (network event or abnormal event includes “changes to infrastructure” and “access requests from unauthorized users”). A POSA therefore would have understood that Veselov teaches, or at least suggests, that taking the snapshot “is performed based on detection of a network event or an abnormal event,” as recited.

231. It is therefore my opinion that claims 11 and 28 were obvious.

10. Claims 12 and 29

232. Claims 12 and 29 depend from claims 1 and 18, respectively, and further recite “further comprising mitigating a potential cyber threat posing a risk to

the protected virtual cloud asset.” As I explain below, Veselov alone, or in combination with Basavapatna, teaches these limitations.

233. Veselov teaches identifying security risks based on its snapshot analysis and then “mitigating a potential cyber threat posing a risk to the protected virtual cloud asset,” as recited. For example, it describes “comparing the assessment results to a remediation framework to identify one or more actions the user can take to address the vulnerabilities, and **providing the identified actions** to the user; and the like.” EX1007 (Veselov), 10:17-36 (emphasis added). Veselov thus teaches providing actions that address the identified security problems. This is consistent with common practice in the field since scanning systems such as Veselov’s routinely mitigated identified security problems (e.g., fixing the problem or taking an action to prevent damage). *Supra*, §VIII.D.3; EX1040 (Ammons), ¶49 (repairing problematic change to file); EX1070 (Hutchins), Abstract (“In response to detecting malicious code during the scan operation, action is taken at the first virtual machine to address the detection of the malicious code.”); EX1056 (Chari), ¶¶41, 60, 85 (discussing mitigation techniques upon detection of security problems).

234. This limitation is also taught by Basavapatna, which similarly teaches mitigating potential cyber threats by providing countermeasures to detected threats. Basavapatna teaches applying active countermeasures, such as installing software

patches or turning off vulnerable machines, and passive countermeasures, such as blocking network traffic from an attack source using a firewall. EX1008 (Basavapatna), ¶¶20-25, 28-29. These countermeasures mitigate detected potential threats. *Id.*, ¶¶21 (“Active countermeasures can be countermeasures that eliminate, in whole or in part, the existence of the vulnerability.”), 22 (“Passive countermeasures cover up the existence of the vulnerability to shield it from exploitation (e.g., by a threat).”), 24 (passive countermeasure may provide protection by “mitigating damage caused by an attack”). As I explained above, a POSA had reason to combine Basavapatna’s teachings with Veselov’s. *Supra*, §XII.A. Basavapatna’s discussion of applying particular mitigation techniques is also consistent with Veselov’s general discussion of providing remedial actions and consistent with common practice in the field. *Supra*, §VIII.D.3; *see also, e.g.*, EX1025 (NIST Incident Handling), 24.

235. It is therefore my opinion that claims 12 and 29 were obvious.

11. Claim 13 and 30

236. Claims 13 and 30 depend from claims 1 and 18, respectively, and further recite “wherein analyzing the snapshot to detect potential cyber threats includes detecting cyber threats that do not represent vulnerabilities.” As I explain below, Veselov, alone or in combination with Basavapatna, teaches the limitations of claims 13 and 30.

237. Veselov teaches analyzing the snapshot to detect “cyber threats that do not represent vulnerabilities,” consistent with the examples of such cyber threats listed in the ’798 patent. The ’798 patent states that “risky application features (e.g., support of weak cipher suites or authentication methods)” and “services running on startup” are cyber threats that do not represent vulnerabilities.” EX1001, 6:19-30. As I explained above with respect to element 1.3, Veselov teaches or at least suggests analyzing the snapshot to detect cyber threats associated with risky application features and services run on startup with its teachings of snapshot-based analysis that detects CVEs. *Supra*, §XII.B.5 (citing, *inter alia*, EX1007 (Veselov), 2:22-54, 7:16-22, 14:41-65; EX1090 (CVE-2006-2923); EX1091 (CVE-2007-0018), EX1100 (CVE-2018-6954); EX1101 (CVE-2018-15687); EX1102 (CVE-2018-1312); EX1103 (CVE-2013-2765)). As an example, Veselov teaches the security assessment may “analyze the behavior of computing resources to identify ... bad configurations, and the like.” EX1007 (Veselov), 1:46-49. To accomplish this behavioral analysis, Veselov explains that in some embodiments a “capturing device may monitor all relevant communications, data processing, and other activities” of the VM to generate a “real-time snapshot describing the activity and connections” of the VM. *Id.*, 15:41-59. While Veselov discusses this snapshot generation technique in relation to Figs. 3A-B, such techniques would equally apply to all other

embodiments. *Id.*, 16:56-67 (discussing “virtual machine 512 operating system 560, runtime environment 562, software applications 564, and files/file system 566 ... and volume image 508, as well as their implementation and interactions” in Figs. 5A-B are same as Figs. 3A-B); *supra*, ¶91.

238. Basavapatna also teaches detecting the types of cyber threats that the ’798 patent identifies as “cyber threats that do not represent vulnerabilities.” In addition to the examples noted above—which included risky application features such as support of weak authentication methods—the ’798 patent also states that “weak passwords” are a type of cyber threat that does not represent a vulnerability. EX1001, 6:19-30. Basavapatna teaches or at least suggests detecting weak passwords and weak authentication methods. For example, it discusses fixing identified security problems such as when “a browser setting is too loose, or a password is too short.” EX1008 (Basavapatna), ¶21. This teaches or at least suggests detecting these security problems since detecting a problem is typically a precursor to fixing the problem, and Basavapatna discusses analyzing the types of system configurations that a POSA would have known routinely included authentication and password configurations. *See id.*, ¶¶14-16 (analyzing asset configuration), 34-37 (discussing asset configuration), 44 (threat applicability data specifies configuration that asset must have in order to be vulnerable to threat such

as “particular versions of products, or products configured in a particular way”), 49 (threat definition data may include “additional details on needed configurations of the assets or other details of the threat, for example, particular product versions or settings including the applicability data described above”). I previously explained why a POSA had reason to combine Basavapatna’s detection and risk-assessment techniques with Veselov’s snapshot-analysis techniques (*supra*, §XII.A), and those reasons apply equally here since the above teachings of Basavapatna are part of its detection and risk-assessment process.

239. It is therefore my opinion that claims 13 and 30 were obvious.

12. Claims 14 and 31

240. Claims 14 and 31 depend from claims 1 and 18, respectively, and further recite “wherein analyzing the snapshot to detect potential cyber threat includes detecting a vulnerable application file by comparing a hash value of the vulnerable application file against a database of known application files.” As I explain below, the combination of Veselov and Basavapatna teaches these limitations.

241. The combination of Veselov and Basavapatna teaches or suggests “detecting a vulnerable application file by comparing a hash value of the vulnerable application file against a database of known application files,” as part of a snapshot analysis. As I explained above, Veselov teaches analyzing the snapshot to detect

security problems by matching application files against known lists such as CVE repositories, and Basavapatna similarly teaches detecting vulnerable application files by comparing installed applications to known lists of vulnerable applications. *Supra*, §XII.C.5. These types of application-matching processes were routinely performed by computing cryptographic hashes of the scanned system's files (i.e., a "vulnerable application file") and comparing that hash value against a database of known application files such as a CVE repository. EX1017 (Rosendahl), 7:58-60 ("These CVEs may be periodically entered into the threat database 132...."); EX1053 (Pahlad), ¶162 (discussing use of "cryptographic hash" as a substantially unique identifier of virtual data object for matching purposes); EX1063 (Hibbert), ¶¶75 (discussing use of CVEs for vulnerability identification), 244-48 (discussing use of identifier such as "a hash of an application"); EX1068 (Epstein), 4:29-39 (discussing hash computation/matching); EX1076 (Milner), 2:39-42 ("A black list can include a list of archival files that are known to be insecure, such as known insecure packages/subpackages referenced in a Common Vulnerabilities and Exposures (CVE) list."), 2:52-59 ("The security tool can compare hashed versions of the identified components to hashed versions of the known insecure components included in the black list. The security tool can be configured to verify the security of the software package based on the comparison. The security tool can verify that

the software package is insecure if an identified component of the software package is found in the black list.”). Moreover, a POSA would have recognized that this predictable, commonplace technique for performing the types of CVE analysis described in Veselov provided a fast but still effective way to match data (e.g., a hash of an application file vs. hashes of a known vulnerable applications in a database). EX1063 (Hibbert), ¶¶75, 244-48; EX1076 (Milner), 2:39-42, 2:52-59; EX1068 (Epstein), 4:29-39; *see also* EX1053 (Prahlad), ¶162. Therefore, at a minimum, it would have been obvious in view of the references’ teachings for the snapshot analysis to include “detecting a vulnerable application file by comparing a hash value of the vulnerable application file against a database of known application files,” as recited.

242. It is therefore my opinion that claims 14 and 31 were obvious.

13. Claims 15 and 32

243. Claims 15 and 32 depend from claims 1 and 18, respectively, and further recite “wherein analyzing the snapshot to detect potential cyber threats includes looking for cyber threats inside containers stored on the virtual disk.” As described below, Veselov teaches these limitations.

244. Veselov teaches that the previously discussed snapshot analysis includes “looking for cyber threats inside containers stored on the virtual disk,” as recited. As I discussed above, Veselov teaches that the analyzed assets may be

virtual assets such as VMs or containers and that a VM itself may include multiple containers. *Supra*, §XII.C.6; EX1007 (Veselov), 1:34-40 (“Virtual machines may themselves be partitioned into multiple isolated virtual systems, called ‘**containers**.’” (emphasis added)), 2:44-51 (targets may be “a virtual machine, a group of interconnected virtual machines, **a container of a virtual machine**, a software application, a service, a server, a hard disk drive or logical volume, etc.” (emphasis added)). Veselov also teaches direct analysis of the snapshot by, for example, reading files and other data included in the snapshot. *Supra*, §XII.B.5; EX1007 (Veselov), 16:56-17:8 (“The scanning service 510 may transform the volume image 508 into a format in which the scanning service 510 can read the copied files and other data; in the illustrated examples, the scanning service 510 does this by mounting the volume image as a virtual storage device.”). A POSA would have recognized that in VMs that include multiple containers, as taught by Veselov (e.g., EX1007, 1:34-40), the containers’ data (e.g., container images) were routinely stored in the VM’s filesystem and would therefore be included in the snapshot of the virtual disk I discussed above (i.e., these would be “containers stored on the virtual disk”). *Supra*, §XII.B.3 (discussing snapshot contents); EX1007 (Veselov), 3:20-40 (snapshot may include “block-level image of the entire logical volume” or portions thereof); EX1120 (Xue), ¶¶41-42, Fig. 8 (depicting VM snapshot that includes

multiple container images); EX1121 (Bo), 15:1-44 (describing extraction of container data from image), Fig. 2A (illustrating same); *see also* EX1017 (Rosendahl), 17:28-18:12 (describing multiple container images on single disk image), 19:51-63 (discussing container images), Fig. 3; EX1120 (Xue), ¶¶42-44, Figs. 9-10. In other words, by teaching analysis of the snapshot's contents/files to detect cyber threats, Veselov teaches or at least suggests analyzing container data included in the snapshot in the disclosed scenario where the VM includes containers. *See, e.g.*, EX1007 (Veselov), 1:34-40, 3:61-67 (scanning system "may mount the logical volume (or transform the snapshot into a data file by identifying the file system of the original logical volume and then mount the snapshot in the duplicate logical volume) as a hard drive with data stored thereon; the system may then perform the security assessment on the data at the file system level"). Veselov thus teaches or at least suggests that analyzing the snapshot to detect potential cyber threats includes "looking for cyber threats inside containers stored on the virtual disk."

245. At a minimum, this limitation would have been obvious to a POSA in view of Veselov's teachings. As I noted above, Veselov teaches performing robust security assessments of target assets to identify security problems and explains that the assessment may include analyzing containers of a VM. *Supra*, §XI.A (Veselov

summary); EX1007 (Veselov), 1:34-40, 2:46-58. Analysis of such containers (e.g., directly scanning the container images) to look for cyber threats inside the containers—whether the containers are included in a target VM or are the target assets themselves—was also commonplace in the field and routine for a POSA. *See, e.g.*, EX1078 (Hufsmith), Abstract, ¶¶31-37. Given Veselov’s discussion of performing comprehensive detection of cyber threats via direct analysis of snapshot data (*supra*, §§XI.A, XII.B.5)—snapshot data that may include container images, as I discussed above—a POSA would have found it obvious to perform the routine techniques of identifying container data (e.g., container images) in the snapshot of the virtual disk and look for cyber threats inside the containers (e.g., directly analyzing the container images).

246. It is therefore my opinion that claims 15 and 32 were obvious.

14. Claims 17 and 34

247. Claims 17 and 34 depend from claims 1 and 18, respectively, and further recite “wherein the virtual cloud asset is a server or a storage system in a cloud computing platform.” I understand these claim limitations are recited in the alternative such that either the virtual cloud asset being a server or a storage system in a cloud computing platform is sufficient to satisfy claims 17 and 34. As I explain below, Veselov teaches these limitations.

248. First, Veselov teaches that the virtual cloud asset is a server in a cloud computing platform. As I explained above, Veselov teaches that the target asset may be a virtual cloud asset such as a VM. *Supra*, §XII.B.1; EX1007 (Veselov), Abstract. Veselov also teaches that that this target asset may be a server. *Id.*, 2:54-58. Veselov also discusses a specific example in which the scanned virtual cloud asset is a web server. *See* EX1007 (Veselov), 9:5-8 (assessed logical volume may be a boot volume for a web server). Moreover, in addition to the virtual cloud asset itself being a server, Veselov also teaches that the virtual cloud asset is hosted on a server in a cloud computing environment. *See id.*, 5:23-50, 18:57-19:20, Figs. 1 (depicting VMs hosted on servers 142), 7 (depicting web server).

249. Second, Veselov teaches that the virtual cloud asset is a “storage system” in a cloud computing environment, as recited. Veselov explains that the target asset stores data and has storage capabilities such as virtual disk storage. *See id.*, 6:7-24 (discussing virtual computing components of VMs such as “virtual memory, virtual disk storage, and the like”), 11:21-33 (discussing VM storage); *see also id.*, 2:22-35 (security assessment may be performed in data center at any level of abstraction, including “block-level storage devices, memory, hard disk drives, and other physical volumes; file systems; logical volumes and partitions; operating systems and virtual file systems; application-specific frameworks; and virtual

machines and virtual environments implementing virtual machines”). Veselov thus teaches, or at least suggests, that the virtual cloud asset is a storage system. This is also consistent with common practice at the time since VMs in datacenters, such as those discussed in Veselov, were routinely used as part of storage systems in cloud computing platforms. *See, e.g.*, EX1012 (Doctor), 1:14-37 (discussing use of virtual resources in cloud storage); EX1026 (ENISA Report, 2018), 74 (discussing “widespread adoption of cloud storage”).

250. Lastly, Veselov teaches that the virtual cloud asset is a server or storage system “in a cloud computing platform,” as recited. As I previously explained, Veselov teaches or at least suggests that the virtual cloud asset operates as part of a cloud computing platform. *See supra*, §XII.B.1; EX1007 (Veselov), 2:64-3:19 (teachings apply to distributed computing environment “(e.g., a ‘cloud’ computing environment)”). The server or storage service I discussed above would therefore be a server or storage service “in a cloud computing platform” for the same reasons.

251. It is therefore my opinion that claims 17 and 34 were obvious.

XIII. GROUND 2: CLAIMS 6 AND 23 WERE OBVIOUS OVER VESELOV, BASAVAPATNA, AND KAPOOR

252. In this ground, I discuss a combination of Veselov, Basavapatna, and Kapoor. This builds upon the combination of Veselov and Basavapatna that I discussed in Ground 1, further applying Kapoor’s disclosure of reading PID files to

determine running processes as part of a cyber security assessment. In §XIII.B, I explain how these teachings apply to certain limitations of the dependent claims 6 and 23. The combination with Kapoor does not change my previous explanation in Ground 1 of why the other limitations of the claims addressed in that ground were obvious (*supra*, §XII), since the applied teachings of Kapoor are consistent with the general approach to the security assessment taught by Veselov, and Basavapatna.

A. Reasons to Combine Veselov, Basavapatna, and Kapoor.

253. In my opinion, a POSA would have had reason to combine the teachings of Veselov, Basavapatna, and Kapoor to arrive at claims 6 and 23.

254. As an initial matter, Veselov, Basavapatna, and Kapoor each describe systems and methods that are analogous to one another and to the '798 patent. Kapoor describes methods of performing a security assessment of a virtualized cloud environment, similar to that taught by Veselov, Basavapatna, and the '798 patent. EX1111 (Kapoor), 4:27-34 (discussing the use of “cloud-based resources”), 15:54-67 (discussing the use of VMs); *supra*, §§XI.A-C (summary of references), XII.A (Ground 1 reasons to combine). It is therefore in the same field as Veselov, Basavapatna, and the '798 patent, and its teachings are pertinent to issues addressed by the '798 patent such as securing virtual assets from cyber threats. EX1001, 1:12-14 (“This disclosure relates generally to cyber-security systems and, more specifically, to techniques for securing virtual machines.”).

255. The combination of Veselov, Basavapatna, and Kapoor would have merely involved using well-known techniques to perform their known, predictable function, with predictable results. I discussed the well-known, predictable nature of the Veselov-Basavapatna combinations above. *Supra*, §XII.A. Reading PID files to determine running processes, as taught by Kapoor, was also well-known and predictable. EX1111 (Kapoor), 7:21-24 (scanning “within the /proc/pid directory” to evaluate running processes), 7:63-67, 37:35-37 (PIDs “identify processes running at a given time”); *see also see also* EX1112 (LinuxMan), 1-2 (“/proc/[pid]” is “numerical subdirectory for each running process; the subdirectory is named by the process ID” and each “/proc/[pid] subdirectory contains the pseudo-files and directories”), 3-14, 15 (PID files include “[s]tatus information about the process” include whether processing is “R Running” and are used by Linux “ps” command), 16-22 (describing other PID files of running processes);⁴² EX1113 (LinfoPID), 1,

⁴² EX1112 is an archived copy of “Linux Programmer’s Manual: proc – process information pseudo-filesystem,” archived on December 15, 2018, by the Internet Archive’s “Wayback Machine” at

(“The PIDs for the processes currently on the system can be found by using the ps command”), 2 (“[E]xtensive information can be found about any PID and the process it represents by reading the PID’s status file....”).⁴³ The further combination with Kapoor would simply use the well-known method of reading PID files to predictably perform its known function (i.e., identifying running processes/applications) without impacting the rest of the previously discussed security assessment. Applying these teachings is consistent with the combination I discussed in Ground 1 since Basavapatna generally teaches determining the running applications as part of the security assessment. EX1008 (Basavapatna), ¶¶37 (asset configuration data includes “particular software products executing on the system (e.g., a word processor or a particular web server)”), 60, 75-76, 78 (threat is

<https://web.archive.org/web/20181215072140/http://www.man7.org:80/linux/man-pages/man5/proc.5.html>.

⁴³ EX1113 is an archived copy of “PID Definition,” archived on November 1, 2018, by the Internet Archive’s “Wayback Machine” at <https://web.archive.org/web/20181101094536/https://www.linfo.org/pid.html>.

applicable “when the asset is running software that is of a type that can have a vulnerability that is exploited by the threat.”), 79, 82, 92, 104, 106, 112.

256. A POSA would have been motivated to apply Kapoor’s teachings to the combination of Veselov and Basavapatna that I discussed in Ground 1. As I previously explained, Basavapatna teaches determining whether applications are running as part of the risk-assessment process. *Supra*, §XII.C.3.b; EX1008 (Basavapatna), ¶¶37 (asset configuration data may include software “running” on asset), 78-81 (threat-centric risk metric analyzes asset configuration data to determine threat applicability, which may be based on whether asset “is running” particular software), 83-85 (same); *see also id.*, ¶¶60, 104-05 (similar assessment for vulnerability-centric risk metric). While Basavapatna does not provide a detailed discussion of what data is analyzed to determine what applications are running on the system, Kapoor teaches a simple and effective way to make this determination using PID files that are routinely maintained in many types of computer systems (e.g., systems running a Unix-like OS such as Linux). EX1111 (Kapoor), 7:21-23, 7:63-67, 10:56-61, 36:5-26, 37:35-37; *see infra*, §XIII.B (discussing Kapoor’s teachings in more detail). While Kapoor reads PID files to determine running processes for purposes of its network-connection security analysis, a POSA would have recognized that this determination was equally applicable and useful to the

Veselov-Basavapatna combination, which similarly considers running applications as I explained in §XII.C.3.b. Moreover, given Basavapatna's general discussion of considering whether applications are running as part of the security assessment, a POSA had reason to look to references discussing the determination of running applications in Linux and other Unix-like OS's since Veselov and Basavapatna explain that the assessed systems may run various OS's, and Basavapatna expressly teaches analyzing Linux systems. *See, e.g.*, EX1007 (Veselov), 1:22-40, 10:55-67; EX1008 (Basavapatna), ¶37. A POSA also would have understood that the PID files read in Kapoor would be included in the snapshots described by Veselov in many circumstances. *See, e.g.*, EX1007, 3:32-40 (snapshot may include machine memory and state of devices). A POSA therefore would have recognized that the types of PID files discussed in Kapoor would have been present and readily analyzed in Veselov's snapshots in many circumstances and would therefore provide a simple and effective way to determine whether a given application is running. EX1112 (LinuxMan), 4 ("'/proc/[pid]/cmdline ... [is a] read-only file hold[ing] the complete command line for the process'"), 4-5 ("'/proc/[pid]/exe ... is a symbolic link containing the actual pathname of the executed command.'"), 15 (process status information). The combination is therefore a straightforward, efficient

implementation of Basavapatna’s general teachings to determine if an application associated with a cyber threat is running.

257. A POSA would have had a reasonable expectation of success in combining the above teachings of Veselov and Basavapatna, and Kapoor to arrive at the claimed invention. The method of reading PID files to determine information about the running processes as discussed in Kapoor was well understood and routinely practiced. EX1112 (LinuxMan), 1-2 (“/proc/[pid]” is subdirectory for PID files of running processes), 3, 4 (“cmdline” PID file contains command line of running process”), 4-5 (“exe” PID file contains pathname information), 6-14 (discussing various PID files), 15 (“stat” PID file contains process status information), 16-22 (discussing various PID files); EX1113 (LinfoPID), 1-2. A POSA also would have expected this technique to be readily accomplished in the context of the snapshot-based analysis since snapshot analysis—including the analysis of memory structures, filesystem data, and system processes—was well understood and predictable. *See supra*, §§VIII.B (snapshots and contents thereof well known), VIII.C.2 (discussing VMI techniques); *see also* EX1007 (Veselov), 3:61-67 (assessment may be performed on snapshot “at the file system level”), 8:34-47 (discussing analysis of snapshot as a data file), 9:28-32 (file-system analysis), 10:1-36 (snapshot analysis), 11:47-67 (discussing contents of snapshot), 13:53-63 (assessment VM in Figs. 3A-B

may include file system/files “identical” to those of original VM), 16:56-17:9 (snapshot in embodiments of Figs. 5A-B includes files/file system that is the same as described for Figs. 3A-B), 17:9-29 (scanning service can “read files and other data” such as OS, applications, “and files/file system”), 18:16-56 (discussing analysis of files/directories). Therefore, in my opinion, combining the references’ teachings would not have presented meaningful technical challenges or produced unexpected results.

B. Claims 6 and 23

258. Claims 6 and 23 depend from claims 3 and 20, respectively, and further recite “wherein scanning the parsed copy further comprises any one of: reading process identification number (PID) files; and checking if the at least access times of the PID files match against process descriptors.” I understand these claim limitations are recited in the alternative and either limitation satisfies the claim. As I explain below, the combination of Veselov, Basavapatna, and Kapoor teaches “reading process identification number (PID) files,” as part of scanning the parsed copy.

259. As a preliminary matter, I previously explained that the combination of Veselov and Basavapatna teaches that “scanning the parsed copy” of a snapshot may include analyzing machine memory stored in the snapshot to deduce what applications and modules executed in the asset. *Supra*, §XII.C.3.b. I also explained that Veselov teaches that scanning the parsed copy includes “reading the files of the

mounted volume image” by identifying “files, software packages, application versions, and the like, for software installed on the target virtual machine 312.” EX1007 (Veselov), 18:16-44; *see also supra*, §§XII.B.5, XII.C.2.b. Moreover, Veselov teaches the parsed copy may include, “file system and file information; operating system information; system and software patches applied ... software packages installed, and configurations thereof; system log properties; and the like.” EX1007 (Veselov), 9:55-67. The combination of Veselov and Basavapatna thus teaches that the parsed copy of the snapshot may include the VM’s files/file system and that the analysis of this parsed copy may including analyzing machine memory in the snapshot to determine running applications.

260. While Basavapatna does not expressly teach that the determination of whether an application is running is performed by reading process identification number files, Kapoor teaches or at least suggests determining the running applications in this manner. *Supra*, §XI.C (Kapoor summary). Kapoor teaches collecting information about a node—which may be a VM (EX1111 (Kapoor), 13:30-54, 14:20-44, 15:54-67)—including the processes running on it. *E.g., id.*, 6:41-7:67, 8:31-37, 14:14-19, Fig. 2. It was well known at the time that many Unix-like systems (e.g., Linux) store a PID file for each running process. For example, Linux systems routinely stored various PID files for each running process in the

“/proc/” directory. *See, e.g.*, EX1112 (LinuxMan), 1-2 (describing PID files located in “/proc” directory); EX1113 (LinfoPID), 2 (PID status file located in “/proc” directory); EX1114 (TutorialPoints), 3 (“Whenever you issue a command in Unix, it creates, or starts, a new process.... A process, in simple terms, is an instance of a running program. The operating system tracks processes through a five-digit ID number known as the pid or the process ID.”).⁴⁴ Consistent with this understanding, Kapoor explains that PIDs “identify processes running at a given time.” EX1111 (Kapoor), 37:28-37. Kapoor further teaches reading PID files as a simple way of determining which processes are running at a given time. For example, while discussing various ways of analyzing processes associated with network connections, Kapoor describes reading the PID files in the /proc/ directory to determine the running processes. *Id.*, 7:10-31. In particular, it describes mapping inodes and PIDs by “scan[ning] within the /proc/pid directory” and then examining

⁴⁴ EX1114 is an archived copy of “Unix/Linux–Processes Management,” archived on December 17, 2018, by the Internet Archive’s “Wayback Machine” at <https://web.archive.org/web/20181217170426/https://www.tutorialspoint.com/unix/unix-processes.htm>.

file descriptors “[f]or each of the processes currently running.” *Id.*, 7:21-24; *see also id.*, 7:63-67 (identifying new running processes by keeping track of latest reported process and beginning “search of /proc/ in a forward order starting from the PID of that process”). In other words, Kapoor is interested in obtaining the set of processes currently running on the system, so it determines this information by reading the PID files. Kapoor thus teaches “reading [PID] files” as recited.

261. It therefore would have been obvious for the scanning of the parsed copy of the snapshot to include reading PID files, as recited, in view of the references’ teachings. In sum, Veselov teaches performing a security assessment by scanning a parsed copy of the snapshot, which can include the VM’s files/file systems, and Basavapatna further teaches evaluating risk levels associated with potential cyber threats based in part on whether the assessed asset is running a given application, though it does not describe how the “running” determination is made. Kapoor, however, describes the well-known technique of determining which applications are running by reading the system’s PID files. EX1111 (Kapoor), 8:31-37, 14:17-19 (process level information collected includes “what user/application is responsible for launching a given process and the binary being executed.”), claims 15, 36 (determining “binary associated with the process”), Figs. 2 (illustrating method step to determine process binary), 3A-B; EX1112 (LinuxMan), 4

(“/proc/[pid]/cmdline ... [is a] read-only file hold[ing] the complete command line for the process”), 4-5 (“/proc/[pid]/exe ... is a symbolic link containing the actual pathname of the executed command.”), 15. As I explained above (*supra*, §XIII.A), a POA had reason to apply Kapoor’s teachings to the Veselov-Basavapatna combination such that the determination of running applications, as described generally in Basavapatna, would be performed using the well-known PID-file-reading technique taught or at least suggested in Kapoor. The combination thus teaches the limitations of claims 6 and 23 as a whole.

262. It is therefore my opinion that claims 6 and 23 were obvious.

XIV. GROUND 3: CLAIMS 11 AND 28 WERE OBVIOUS OVER VESELOV, BASAVAPATNA, AND ROTH

263. In this ground, I discuss a combination of Veselov, Basavapatna, and Roth. This builds upon the combination of Veselov and Basavapatna that I discussed in Ground 1, further applying Roth’s teachings regarding the well-known technique of taking snapshots based on the detection of network or abnormal events. In §XIV.B, I explain how these teachings apply to certain limitations of dependent claims 11 and 28. This combination does not change my previous explanation in Ground 1 of why the other limitations of the claims addressed in that ground were obvious (*supra*, §XII), since the applied teachings of Roth are consistent with

Veselov's teachings of event-triggered snapshots and consistent with the security assessment taught by the combination of Veselov and Basavapatna.

A. Reasons to Combine Veselov, Basavapatna, and Roth

264. In my opinion, a POSA would have had reason to combine the teachings of Veselov, Basavapatna, and Roth to arrive at claims 11 and 28.

265. As an initial matter, Veselov, Basavapatna, and Roth describe systems and methods that are analogous to each other and to the '798 patent. For example, like Veselov, Basavapatna, and the '798 patent, Roth discusses techniques that facilitate security analysis of computer assets. EX1116 (Roth), Abstract, claim 1; *supra*, §XI.F (Roth summary), XII.A (reasons to combine Veselov and Basavapatna). Roth is also pertinent to the issues addressed by the '798 patent since Roth discusses ways to capture the state of virtual assets in the form of snapshots. EX1001, 1:12-14; EX1116 (Roth), 1:36-44, 2:67-3:67, claim 1; *supra*, §§IX ('798 patent summary), XI.F (Roth summary). For example, a POSA would have recognized that the computer-forensic analysis discussed in Roth was pertinent to vulnerability assessments since they involve similar computer resources/data and address substantially overlapping security concerns. Moreover, computer forensic analysis was routinely used to discover and understand new threats and vulnerabilities that have not been seen or cataloged before. *E.g.*, EX1116 (Roth), 1:28-43 (discussing concern with potentially insecure system states), 2:56-3:16

(discussing detection of events such as “an intrusion/security compromise of the virtual machine”), 3:50-67 (taking snapshots).

266. In addition to the known elements of Veselov and Basavapatna that I discussed above (*supra*, §XII.A), Roth’s techniques of taking snapshots in response to the detection of various events were well known and predictably implemented in the context of computer systems such as those described in Veselov and Basavapatna. EX1116 (Roth), 2:67-3:67; *see also* EX1119 (Hiltgen), 19:29-63 (“At the virtualization layer of a virtual machine, snapshotting is activated. This may be done using a variety of trigger mechanisms. For example, snapshotting may be activated by a manual event, a time based event, in accordance with a configuration file, a notification or command received via a communications medium, etc.”). Indeed, Veselov already teaches event-triggered initiation of a security-assessment process that may include the taking of a snapshot. EX1007 (Veselov), 9:14-25, 14:41-53, 15:35-40, 23:48-56, claim 4. In other words, Veselov teaches event-triggered snapshots, and Roth further discloses taking snapshots based on specific types of events. The combination would thus use Roth’s known techniques to achieve their known function (i.e., determining an appropriate moment to trigger the taking of a snapshot that will be used for subsequent security analysis) with

predictable results. *See, e.g.*, EX1116 (Roth), 2:42-47 (saved VM state used for subsequent forensic analysis).

267. Furthermore, a POSA would have been motivated to apply Roth’s teachings in order to provide snapshots with data relevant to pressing potential security problems. Veselov uses snapshots as the basis for its security analysis and seeks to identify potential security problems in the associated virtual asset (*supra*, §XI.A), and Basavapatna discusses the well-known fact that certain security problems may be a higher priority than others (*supra*, §XI.B). As I noted above, Veselov further teaches that the snapshot-based security assessment (which may include the taking of a snapshot) may be triggered by the occurrence of an event. EX1007 (Veselov), 9:14-25, 14:41-53. Roth similarly recognizes that certain types of events indicate potential security problems. For example, it discusses detecting events such as “an intrusion/security compromise of the virtual machine.” EX1116 (Roth), 2:67-3:16; *see also id.*, 1:25-36 (discussing well-known security concerns regarding virtual assets being “changed to a nonstandard configuration” or being “compromised by an installation of unauthorized executable code”). In cloud computing systems and servers, applications and configurations were often controlled by system administrators, so an unauthorized change was typically considered an abnormal event. Roth further explains that detecting an event triggers

the taking of a snapshot, which captures the state of the virtual asset at that time, and this saved state is then stored so that it may be analyzed for security purposes. *Id.*, 2:42-47 (saved VM state used for subsequent forensic analysis), 3:50-67; *supra*, §XI.F. In other words, certain events indicate the existence of a potential security problem, so using the detection of such events to trigger the taking of a snapshot provides a targeted mechanism for efficiently capturing system state that can then be analyzed to evaluate security issues. A POSA would have recognized that applying this technique in combination with the teachings of Veselov and Basavapatna would similarly facilitate the efficient, targeted capture of snapshots (and the subsequent security analysis of those snapshots) upon the occurrence of security-relevant events.

268. A POSA would have had a reasonable expectation of success in combining the above teachings of Veselov, Basavapatna, and Roth to arrive at the claims. Detecting security-relevant events, as taught by Roth, was routinely and predictably performed as part of security assessments such as Veselov's and Basavapatna's. *See, e.g.*, EX1007 (Veselov), 14:41-53, 15:35-40, 23:48-56, claim 4 (discussing event-triggered assessments); EX1119 (Hiltgen), 19:29-63 (discussing event-triggered snapshots). Applying Roth's teachings is consistent with the Veselov-Basavapatna combination, as I previously discussed, and would have been

readily implemented in that context. For example, Veselov and Roth—both Amazon patents—discuss analogous virtual assets and resources, and Roth discusses taking the same type of snapshots that Veselov analyzes for purposes of its security assessment. *See, e.g.*, EX1007 (Veselov), 9:14-17 (snapshot represents state of virtual resource at the time), 9:42-55 (discussing change-based snapshots); EX1116 (Roth), 3:64-67 (snapshot represents state of virtual resource at the time), 2:42-47, 8:25-30, 13:37-14:58 (discussing change-based snapshots); *compare* EX1007 (Veselov), Fig. 7 *with* EX1116 (Roth), Fig. 10. Moreover, Veselov already teaches that the assessment (which can include the taking of a snapshot) may be triggered by an event. EX1007 (Veselov), 9:14-25, 14:41-53; *supra*, §XII.C.9. Thus, it is my opinion that the combination would not have presented significant technical challenges or produced unexpected results.

B. Claims 11 and 28

269. Claims 11 and 28 depend from claims 1 and 18, respectively, and further recite “wherein taking the snapshot is performed based on detection of a network event or an abnormal event.” Below I explain why, in addition to the teachings of Veselov I previously discussed (*supra*, §XII.C.9), these limitations are taught by Roth.

270. Consistent with Veselov, Roth teaches taking a snapshot based on detection of an event. Roth discusses capturing snapshots that represent the state of

a virtual asset at a given time and explains that such snapshots will be used for purposes of security analysis. EX1116 (Roth), Abstract (“A system and method for capturing forensic snapshots of virtual machines....”), 3:64-67 (“Like the initial snapshot, the current snapshot may be a snapshot of the state of the memory of the virtual machine and/or the state of the storage attached to the virtual machine at the current time.”), 7:1-21 (discussing snapshot content). To capture the VM’s state at the relevant time, Roth teaches triggering the taking of a snapshot based on the detection of various types of events. *Id.*, Abstract (capturing snapshot upon “detection of an occurrence of the specified event”), 2:20-41 (“The techniques described include configuring an event trigger for causing, upon occurrence of an associated event, a set of local changes to the computing system of a set of computing systems in a distributed computing environment to be captured.”). For example, Roth explains that once an initial snapshot has been captured, “the virtual machine may be allowed to run and perform its tasks **until the occurrence of the predetermined event (i.e., thereby triggering the event trigger), whereupon a second snapshot may be obtained.**” *Id.*, 3:50-67 (emphasis added). The type of event that triggers the taking of the snapshot is configurable. *Id.*, Abstract (discussing “configuration that species an event”), 2:26-30 (discussing “configuring an event trigger”). As I explain further below, Roth discusses several examples of

the types of events that may trigger the taking of the snapshot, including both a “network event” and an “abnormal event” (though I understand that only one of these events is needed to satisfy the claims).

271. First, Roth teaches that taking the snapshot is performed based on detection of “a network event,” as recited. Roth discusses several examples of the triggering event, including an API request from a separate system and a system intrusion. *Id.*, 2:67-3:11 (event “may be the receipt of a request through an application programming interface from a customer..., the computing resource service provider, or other authorized entity”), 3:11-16 (event may be “detection of an intrusion”). A POSA would have understood these exemplary events to encompass, or at least suggest, a “network event.” For example, Roth explains that its system may operate in a datacenter, and API requests to such systems were routinely received over a network. *See, e.g., id.*, 1:14-24 (“Customers are often able to configure and reconfigure their infrastructure remotely and programmatically in order to best suit their needs.”), 8:30-47 (discussing datacenter), 23:60-24:22 (discussing example environment of Fig. 10, including devices for sending/receiving requests over network), Fig. 2 (depicting data center and remote computer systems), Fig. 10 (depicting example environment); *see also* EX1007 (Veselov), Fig. 1 (depicting remote interaction via API over network); *supra*, §VIII.C (cloud

computing). Intrusion detection in VMs hosted in a datacenter also routinely occurred over a network for similar reasons (i.e., intruders, like customers, often operated remotely from the datacenter). *See, e.g.*, EX1008 (Basavapatna), ¶59 (discussing “network intrusion detection system”); EX1012 (Doctor), 10:16-25 (discussing “network intrusion detection software”); EX1025 (NIST Incident Handling), 16, 55-57 (discussing network intrusion); EX1026 (ENISA Report, 2018), 133 (same); EX1033 (Garfinkel), 1-2 (same).

272. Second, Roth teaches that taking the snapshot is performed based on detection of “an abnormal event,” as recited. Roth explains that the snapshot-triggering event may also be “security compromise of the virtual machine” or a “software or hardware error of the virtual machine.” EX1116 (Roth), 2:67-3:16. For example, Roth describes detection of “the occurrence of certain execution errors” and detection of potentially problematic configurations such as “unauthorized software” installed on the VM or authorized software “being modified from an approved configuration.” *Id.*, 17:35-43. A POSA would have understood such events to encompass, or at least suggest, an “abnormal event” since these examples—i.e., compromised security, unauthorized software, unapproved software modifications, and software/hardware errors—were typically viewed as deviations from normal operation.

273. As I previously explained, a POSA had reason to apply these teachings in combination with the teachings of Veselov and Basavapatna to arrive at claims 11 and 28 as a whole. *Supra*, §XIV.A. It is therefore my opinion that claims 11 and 28 were obvious.

XV. GROUND 4: CLAIMS 14 AND 31 WERE OBVIOUS OVER VESELOV, BASAVAPATNA, AND CZARNY

274. In this ground, I discuss a combination of Veselov, Basavapatna, and Czarny. This builds upon the combination of Veselov and Basavapatna that I discussed in Ground 1, further applying Czarny's disclosure of well-known methods of hash comparison for purposes detecting a vulnerable application file. In §XV, I explain how these teachings apply to dependent claims 14 and 31. The combination with Czarny does not change my previous explanation in Ground 1 of why the other limitations of the claims addressed in that ground were obvious (*supra*, §XII), since the applied teachings of Czarny are consistent with the general approach of the security assessment taught by Veselov and Basavapatna.

A. Reasons to Combine Veselov, Basavapatna, and Czarny

275. In my opinion, a POSA would have had reason to combine the teachings of Veselov, Basavapatna, and Czarny to arrive at claims 14 and 31.

276. As an initial matter, Veselov, Basavapatna, and Czarny each describe systems and methods that are analogous to one another and to the '798 patent.

EX1084 (Czarny), Abstract (“Computer security vulnerability assessment is performed with product binary data and product vulnerability data....”), 3:37-4:47, 9:17-10:42, Figs. 1, 7-10; *supra*, §§XI.A-B, XI.D (summary of references), XII.A (reasons to combine Veselov and Basavapatna). Czarny is therefore in the same field as Veselov, Basavapatna, and the ’798 patent, and its teachings are pertinent to issues addressed by the ’798 patent since those teachings involve detecting security problems in computer assets. EX1001, 4:43-45 (“According to the disclosed embodiments, the security system 140 is configured to detect vulnerabilities and other cyber threats....”).

277. Furthermore, like the elements of Veselov and Basavapatna that I discussed above, comparing application files against a database of known vulnerable applications files using a cryptographic hash, as taught by Czarny, was well-known and predictable. EX1084 (Czarny), 3:37-4:47, 5:44-52 (“A conventional vulnerability assessment system, for example, typically uses data similar to that in the products-to-vulnerabilities mapping database 107.”), 7:31-67, 9:17-10:42, 11:16-12:18, Figs. 1, 7-10; *supra*, §XII.C.12; EX1068 (Epstein), 4:29-39 (discussing hash comparison); EX1063 (Hibbert), ¶¶75 (discussing using CVE for vulnerability identification), 244-48 (using application hash to retrieve risk information); EX1076 (Milner), 2:24-67 (comparing hashes of software packages against hashes in CVE

blacklist); EX1053 (Prahlaḍ), ¶162 (cryptographic hash used as unique identifier for matching purposes).

278. The combination of Veselov, Basavapatna, and Czarny would have merely involved using this well-known matching technique to perform its known, predictable function, with predictable results. I discussed the well-known, predictable nature of the Veselov-Basavapatna combination in Ground 1. *Supra*, §XII.A. The further combination with Czarny would merely involve using a well-known, predictable matching technique to perform its known functions (e.g., comparing a hash value of applications files against a database of known vulnerable applications).

279. For example, Veselov teaches matching installed applications against applications in databases of known vulnerabilities such as CVE repositories. EX1007 (Veselov), 2:35-41 (security assessment may use CVE rules package), 7:15-22 (discussing “software application-level CVEs assessment”). Similarly, Basavapatna teaches matching vulnerability definition data from a vulnerability information service against the installed software of a target asset. EX1008 (Basavapatna), ¶¶15 (“Vulnerability definition data can be received for the particular vulnerability, the vulnerability definition data including an identification of the particular vulnerability...”), 51 (“The vulnerability definition data 205 is received

from one or more vulnerability information services 213. A vulnerability information service 21s identifies vulnerabilities....”), 60 (discussing matching vulnerability definition data with installed applications on the asset). Czarny then describes a particular hash-based comparison technique as part of, for example, a CVE-based vulnerability analysis similar to that described at a higher level in Veselov and Basavapatna. EX1084 (Czarny), 3:56-4:29 (system correlates software product binary data with list of known vulnerabilities and the binary data may be “binary hashes of binary level files of the software products” or “binary hashes of strings of bits”), 6:25-38 (“There are a variety of sources 109 for known product vulnerability and severity information, including CVE (Common Vulnerabilities and Exposures)....”). Thus, using a hash-based comparison in the context of the similar matching analysis taught by Veselov and Basavapatna would merely involve using this comparison technique according to its known function for the same or similar purpose as described in Czarny. The use of this technique would not be meaningfully impacted in the context of snapshot-based analysis since, as I previously explained, the type of snapshot-based analysis taught by Veselov involves comparable data (e.g., applications) and detection techniques (e.g., matching against lists of vulnerable applications) as non-snapshot-based analysis. *Supra*, §XII.A. Indeed, Czarny teaches analyzing disk data of offline systems, which

is similar to analyzing snapshot data as in Veselov. EX1084 (Czarny), 7:4-21, 10:51-11:6.

280. Additionally, a POSA would have been motivated to apply Czarny's teachings to the combination from Ground 1 to provide robust vulnerability detection. Czarny itself discusses the advantages of a hash-based comparison. For example, Czarny teaches that its comparison technique provides "a more thorough, robust and flexible level of functionality" because it allows for detection of vulnerabilities that may be missed due to "the actual binary level data in the files of the software product [being] different from the official version of the software product." EX1084 (Czarny), 4:7-47. Czarny explains that conventional detection methods may miss these types of vulnerabilities because they "simply look[] at file identification data." *Id.*; *see also id.*, 5:44-6:24 (hash matching improves detection of custom/unofficial software). Furthermore, hash-based comparison was also known to provide fast and accurate matching against databases of known application files. EX1076 (Milner), 2:39-59 ("The security tool can compare hashed versions of the identified components to hashed versions of the known insecure components included in the black list."), 6:40-58 (security tool can use hashes "to uniquely identify the known insecure and secure components ... and to easily and quickly compare the known insecure and secure components to other software packages");

see also EX1025 (NIST Incident Handling), 23 (cryptographic hashes “speed incident analysis, verification, and eradication”); EX1063 (Hibbert), ¶¶244-48 (hash identifiers of application sufficient to retrieve risk information); EX1068 (Epstein), 4:29-39. A POSA therefore would have used common, effective comparison technique such as a hash-based comparison, as taught by Czarny, to perform the types of matching described generally in Veselov and Basavapatna in view of the widely recognized benefit of providing thorough, flexible vulnerability detection. In other words, a POSA would have been motivated to apply Czarny’s comparison technique to identify vulnerabilities, assess risk, and report the results in an accurate and predictable manner.

281. A POSA would have had a reasonable expectation of success in combining the above teachings of Veselov, Basavapatna, and Czarny to arrive at the claimed invention. As I explained above, the cyber-threat matching technique discussed in Czarny, including a hash-based comparison, was well understood and routinely used to perform the types of matching-based analysis described at a high level in Veselov and Basavapatna (e.g., CVE/malware identification). EX1063 (Hibbert), ¶¶75, 244-48; EX1068 (Epstein), 4:29-39; EX1076 (Milner), 2:24-67. Therefore, in my opinion, combining the references’ teachings as discussed above

would not have presented meaningful technical challenges or produced unexpected results.

B. Claims 14 and 31

282. Claims 14 and 31 recite “wherein analyzing the snapshot to detect potential cyber threat includes detecting a vulnerable application file by comparing a hash value of the vulnerable application file against a database of known application files.” As I explain below, the combination of Veselov, Basavapatna, and Czarny teaches these limitations.

283. Czarny teaches detecting a vulnerable application “by comparing a hash value of the vulnerable application file against a database of known application files,” as recited. Czarny detects vulnerabilities by matching an asset’s applications against a database of known vulnerable applications. EX1084 (Czarny), 11:7-47, Fig. 10; *supra*, §XI.D (Czarny summary). Czarny describes several ways to perform this matching, including comparing an application file’s original binary data or a hash of the application file against a database of known application files. EX1084 (Czarny), 4:7-24, 5:59-6:24 (using “complete file binary data for a binary hash of the file binary data”), 9:28-50 (storing “hash of the binary data or ... the original binary data”). For example, Czarny teaches using “binary hashes of binary level files of the software products” as part of a hash-based matching method. *Id.*, 9:28-64, claims 1, 6; *see also id.*, Figs. 4-5 (depicting databases for hash-based matching).

Czarny further teaches that the comparison is made against a database of known application files. *Id.*, 11:7-21; *see also id.*, 6:25-38 (discussing use of CVE database), Figs. 1-5 (depicting databases). For example, Czarny describes matching the assessed system's files against a vulnerability database such that a "match indicates that the binary file contains a known vulnerability." *Id.*, 11:7-47; *see also id.*, Fig. 10. Czarny thus teaches detecting a vulnerable application by computing hashes of potentially vulnerable application files on the system and then comparing those hashes against a database of known application files.

284. As I explained in §XIII.A, it would have been obvious to analyze the snapshot, as taught by Veselov, by comparing a hash value of the vulnerable application file against a database of known application files, as taught by Czarny, since Czarny teaches using a comparison technique for the same types of vulnerability assessments discussed in Veselov and Basavapatna (e.g., CVE assessments), and hash comparisons were routinely used for this type of matching. EX1084 (Czarny), 6:25-38 (discussing use of CVE database); *see also* EX1007 (Veselov), 2:35-41 (discussing use of CVE in security assessment), 7:15-22 (same); EX1008 (Basavapatna), ¶51 (discussing using vulnerability definition data from vulnerability information services to identify vulnerabilities); EX1075 (Mohanty), 3:50-54 (discussing use of CVE IDs in security assessment), 4:59-62 (same);

EX1025 (NIST Incident Handling), 23; EX1053 (Prahlad), ¶162; EX1068 (Epstein), 37:67-38:4; EX1063 (Hibbert), ¶¶75, 244-48; EX1076 (Milner), 2:24-67; *supra*, §XII.C.12.

285. It is therefore my opinion that claims 14 and 31 were obvious.

XVI. GROUND 5: CLAIM 13, 16, 30, AND 33 WERE OBVIOUS OVER VESELOV, BASAVAPATNA, AND CHARI

286. In this ground, I discuss a combination of Veselov, Basavapatna, and Chari. This builds upon the combination of Veselov and Basavapatna that I discussed in Ground 1, further applying Chari’s express disclosure of the well-known process of detecting various types of sensitive data, including encryption keys, credit cards, social security numbers and other personally identifiable information (“PII”). In §§XVI.B-C, I explain how these teachings apply to certain dependent claims 13, 16, 30, and 33. The combination with Chari does not change my previous explanation in Ground 1 of why the other limitations of the claims addressed in that ground were obvious (*supra*, §XII), since the applied teachings of Chari are consistent with the general approach to the security assessment taught by Veselov and Basavapatna.

A. Reasons to Combine Veselov, Basavapatna, and Chari

287. As an initial matter, Veselov, Basavapatna, and Chari describe systems and methods that are analogous to each other and to the ’798 patent. For example,

like Veselov, Basavapatna, and the '798 patent, Chari describes security assessments that include detecting sensitive in virtual images. EX1056 (Chari), Abstract, ¶¶1-5, 14; *supra*, §§XI.A-B, XI.E (summary of references), XII.A (reasons to combine Veselov and Basavapatna). Moreover, Chari, like Veselov and Basavapatna, is pertinent to the issues addressed by the '798 patent since all references in the combination address security assessments of virtual computing resources. EX1001, 1:12-14; *supra*, §§IX ('798 patent summary), XI.A-B, XI.E (summary of references).

288. Furthermore, in addition to the known elements of Veselov and Basavapatna that I discussed above (*supra*, §XII.A), Chari's teachings of detecting various types of sensitive data were well known and predictably used in security assessments such as those described in Veselov and Basavapatna. EX1056 (Chari), ¶¶38-41, 64-66 ("[I]llustrative embodiments focus on scanning and sanitization of virtual machine images in a cloud computing environment in order to remove sensitive data contained within the virtual machine images...."). For example, Chari teaches detecting sensitive data in images/snapshots, which are the same types of files analyzed in Veselov. EX1056 (Chari), ¶¶64 ("Illustrative embodiments label sensitive data at each level of a virtual machine image's software stack, such as, for example, storage, guest operating system, middleware, and application levels."), 73

(“[T]he virtual machine image manager generates the snapshot of the specific instance of the virtual machine image....”). The combination would thus involve using Chari’s well-known techniques in the same, predictable way with predictable results.

289. Furthermore, a POSA would have been motivated to apply Chari’s teachings in order to provide a more complete identification/detection of security risks. The presence of unsecured sensitive data was a common concern in security assessments. Veselov involves cloud systems, and Chari observes that residual sensitive data is of particular concern to cloud systems. EX1007 (Veselov), 2:27-3:19; EX1056 (Chari), ¶¶59-62; *supra*, §XI.B.1. Chari teaches mitigating this issue by scanning images/snapshots to detect and sanitize such data, which a POSA would have recognized was equally beneficial in the security assessments taught by Veselov and Basavapatna. Indeed, Chari explains that in cloud computing environments which include VMs, such as those taught by Veselov, sensitive data may have “persist[ed] across multiple virtual machine images leading to many privacy and confidentiality breaches.” EX1056 (Chari), ¶62. A POSA would have been motivated to address this well-known problem by identifying and labeling sensitive data, as taught by Chari. EX1056 (Chari), ¶63 (“Currently, if the virtual machine images/instances are sanitized at all, a manual process is applied or an

incomplete automatic process is applied, such as sanitizing only the passwords during the creation of virtual machine instances. As a result, sanitization methods currently used are adhoc and/or incomplete.”); *see also id.*, ¶¶64-66. In other words, consistent with Veselov and Basavapatna’s general teachings of detecting vulnerabilities based on the potential exposure of sensitive data (e.g., CVEs and CVSSs), Chari expressly discloses detecting specific instances of sensitive data in an image/snapshot. EX1056 (Chari), Abstract, ¶¶14, 76-80 (discussing use of labeling scripts that define types of sensitive data and are used to “discover ... [the] presence or absence of particular sensitive data items....”); EX1007 (Veselov), 2:35-41, 7:15-22 (discussing use of CVEs for threat analysis); EX1008 (Basavapatna), ¶¶15, 55-56, 101, 106, 112 (discussing use CVSS, including confidentiality impact, to determine threat risk); *see also id.*, ¶¶21 (discussing remediating weak passwords), 45 (discussing impact on confidentiality from exploiting vulnerability).

290. A POSA would have had a reasonable expectation of success in combining the above teachings of Veselov, Basavapatna, and Chari to arrive at the claims. Detecting sensitive data, as taught by Chari, was routinely and predictably performed as part of security assessments such as Veselov’s and Basavapatna’s. Applying these teachings is consistent with and readily implemented in combination with Veselov and Basavapatna. EX1007 (Veselov), 2:35-41 (system can use “any

suitable security assessment,” including best-practice packages and static/runtime analysis), 14:13-22 (similar); EX1008 (Basavapatna), ¶¶21 (discussing remediating weak passwords), 45 (discussing impact on confidentiality from successful exploited vulnerability), 56 (discussing confidentiality impact relating to “restricting access to particular data and assets to authorized users”). Thus, it is my opinion that the combination would not have presented significant technical challenges or produced unexpected results.

B. Claims 13 and 30

291. Claims 13 and 30 depend from claims 1 and 18, respectively, and further recite “wherein analyzing the snapshot to detect potential cyber threats includes detecting cyber threats that do not represent vulnerabilities.” I previously explained how the combination of Veselov and Basavapatna teaches detecting certain types of cyber threats that do not represent vulnerabilities, consistent with the examples of such cyber threats listed in the ’798 patent. Below I explain how Chari teaches additional types of such cyber threats that the ’798 patent specification identifies as cyber threats that “do not represent vulnerabilities,” including “sensitive data not being encrypted,” “private keys,” and “system credentials stored clearly.” EX1001, 6:19-26. I discuss each of these examples below.

292. First, Chari teaches detecting sensitive data that is not encrypted. Chari teaches that a “labeler” may be used to detect and label sensitive data identified in

the target VM. EX1056 (Chari), Abstract (discussing processing of “identified sensitive data”), ¶¶37-38 (discussing labeler and types of sensitive data), 60 (“Threats to the cloud computing environments may include exposure of this sensitive data, such as user credentials (i.e., passwords), cryptographic keys, and access to and/or reconstruction of data within a file system.”), 76 (detecting “presence or absence of particular sensitive data items”); *supra*, §XVI.A (reasons to combine). Chari also teaches that this detected sensitive data is not encrypted. For example, Chari describes sanitizing certain types of detected sensitive data by encrypting the data (*id.*, ¶¶41, 85, claim 10), which implies, or at least suggests, that such data is not encrypted when detected. That understanding is also consistent with the Chari’s description of sensitive data as data that the user “does not want to be made public or made available to other users” since unencrypted forms of such data were known to be particularly vulnerable to such exposure. *Id.*, ¶38.

293. Second, Chari teaches detecting private keys. Chari explains that the detected sensitive data may include “cryptographic keys.” *Id.*, ¶38; *see also id.*, ¶¶41, 60. A POSA would have understood such keys to encompass, or at least suggest, private keys since private keys were a common type of cryptographic key stored by these types of computer systems, and the potential exposure of private keys was a well-known security risk. *See, e.g.*, EX1081 (Budko), ¶¶14 (discussing

protection of asymmetric private key), 110-15 (private key allows decryption of data encrypted with corresponding public key), 117 (“Of course, the asymmetric private keys themselves have to be protected....”), 160-66 (discussing management/protection of cryptographic keys, including maintaining control of private key).

294. Third, Chari teaches detecting system credentials stored clearly. Chari explains that the detected sensitive data may include “user credentials (i.e., passwords).” EX1056 (Chari), ¶60; *see also id.*, ¶61 (assessed systems may store sensitive data such as administrator passwords or application credentials/passwords). As I explained above, Chari teaches that the detected sensitive data may be unencrypted (i.e., stored clearly), which applies equally to this particular type of data. *Id.*, ¶¶41, 85, claim 10; *see also id.*, ¶39 (“For example, labeling policies 230 may define that passwords are sensitive data to be labeled within a virtual machine and that a high sensitivity level label is to be attached to the passwords.”), 77 (similar). This is also consistent with common computer-security practices since it was well known that having system credentials stored in the clear on a computer system was a security problem. *See, e.g.*, EX1018 (Scarfone), 3-2 to 3-3 (“Note that one of the biggest security issues with images and snapshots is that they contain sensitive data (such as passwords, personal data, and so on)....”), 3-4

(discussing use of encryption to “prevent information leakage”); EX1025 (NIST Incident Handling), 58 (noting that use of encryption on computer containing sensitive data may mitigate damage from data loss); EX1081 (Budko), ¶¶94-101 (discussing context-dependent data encryption).

295. Accordingly, Chari teaches detecting multiple types of data that the ’798 patent identifies as “cyber threats that do not represent vulnerabilities.” As I previously explained, a POSA had reason to apply these teachings to the snapshot-based analysis of the Veselov-Basavapatna combination. *Supra*, §XVI.A. It is therefore my opinion that claims 13 and 30 were obvious.

C. Claims 16 and 33

296. The combination of Veselov, Basavapatna, and Chari teaches claims 16 and 33, which depend from claims 1 and 18, respectively, and further recite “wherein analyzing the snapshot to detect potential cyber threat includes looking for sensitive data including at least one of personal identifiable information or private keys.” I understand these claim limitations are recited in the alternative and either limitation satisfies claims 16 and 33. Regardless, Chari teaches detecting both “personal identifiable information and private keys.” As I explained with respect to claims 13 and 30, Chari teaches detecting private keys, so Chari teaches “looking for sensitive data including ... private keys” for the same reasons. *Supra*, §XVI.B; EX1056 (Chari), ¶¶38, 41, 60. I also previously explained why Chari teaches detecting

various types of sensitive data, including “credit card numbers, social security numbers, [and] bank account numbers.” EX1056 (Chari), ¶38; *supra*, §XVI.B; see also EX1056 (Chari), ¶¶41, 60, 76. A POSA would have understood credit card numbers, social security numbers, and bank account numbers to be types of personally identifiable information (“PII”) since PII was well-known to encompass such data. See, e.g., EX1105 (DoD Privacy FAQs),⁴⁵ 2 (“Examples of personally identifiable information (PII) include ... final account or credit card number”). Moreover, as I previously explained, a POSA had reason to apply Chari’s teachings in the context of the snapshot-based analysis of the Veselov-Basavapatna combination such that the detection of this sensitive data, as taught by Chari, would be part of “analyzing the snapshot” to detect cyber threats. *Supra*, §XVI.A.

297. It is therefore my opinion that claims 16 and 33 were obvious.

⁴⁵ EX1105 is a copy of “Defense Privacy, Civil Liberties, and Transparency Division, U.S. Department of Defense, FAQs,” archived on May 17, 2018, by the Internet Archive’s “Wayback Machine” at <https://web.archive.org/web/20180517201830/http://dpcl.d.defense.gov/Privacy/About-the-Office/FAQs/>.

XVII. CONCLUDING STATEMENTS

298. In signing this declaration, I understand that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I acknowledge that I may be subject to cross-examination in this case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

299. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: 01/17/2025

By: /Angelos Stavrou/
Angelos Stavrou, Ph.D.

XVIII. APPENDIX A – MATERIALS CITED

Exhibit No.	Description
1001	U.S. Patent No. 11,868,798 to Shua
1003	<i>Curriculum Vitae</i> of Dr. Angelos Stavrou
1004	File History of U.S. Patent Application No. 17/821,345 ('798 Pat.)
1007	U.S. Patent No. 11,216,563 B1 to Veselov
1008	U.S. Publication No. 2013/0191919 A1, published July 25, 2013, to Basavapatna
1009	Wolf, "Virtualization: From the Desktop to the Enterprise" APRESS (2005)
1010	Waldspurger, "Memory Resource Management in VMware ESX Server," USENIX, Proceedings of the 5 th Symposium on Operating Systems Design and Implementation (Dec. 9-11, 2002)
1011	Goldberg, "Survey of Virtual Machine Research," COMPUTER, Vol. 7, No. 6 (June 1974)
1012	U.S. Patent No. 10,944,811 B2 to Doctor
1013	Wood, "Black-box and Gray-box Strategies for Virtual Machine Migration" NSDI '07: 4 th USENIX Symposium on Networked Systems Design & Implementation, USENIX
1014	Klemperer, "Efficient Hypervisor Based Malware Detection," Carnegie Mellon University (May 2015)
1015	vSphere Web Services SDK Programming Guide, VMWare, Inc. (April 17, 2018)
1016	Souppaya, "Application Container Security Guide," NIST Special Publication 800-190 (Sept. 2017) https://doi.org/10.6028/NIST.SP.800-190

Exhibit No.	Description
1017	U.S. Patent No. 11,106,784 B2 to Rosendahl
1018	Scarfone, “Guide to Security for Full Virtualization Technologies: Recommendations of the National Inst. of Standards and Technology,” NIST Special Publication 800-125 (Jan. 2011)
1019	U.S. Patent No. 9,881,040 B2 to Rawat
1020	U.S. Patent No. 9,116,633 B2 to Sancheti
1021	“NIST Cloud Computing Standards Roadmap,” NIST Special Publication 500-291, Version 2 (July 2013).
1022	Roberts, “Public Cloud Service Definition,” VMware vCloud® Architecture Toolkit™ for Service Providers, VMware, Inc. (2018)
1023	ITL Bulletin, “Security Patches and the CVE Vulnerability Naming Scheme: Tools to Address Computer System Vulnerabilities,” NIST (Oct. 2002)
1024	ITL Bulletin, “Preventing and Handling Malware Incidents: How to Protect Information Technology Systems from Malicious Code and Software,” NIST (Dec. 2005)
1025	Cichonski, “Computer Security Incident Handling Guide: Recommendations of the National Institute of Standards and Technology,” NIST Special Publication 800-61, Rev. 2 (Aug. 2012) http://dx.doi.org/10.6028/NIST.SP.800-61r2
1026	ENISA Threat Landscape Report 2018, “15 top Cyberthreats and Trends,” V1.0 (Jan. 2019)
1027	Verizon, “2019 Data Breach Investigations Report” (May 2019)
1028	Di Pietro, “Virtualization Technologies and Cloud Security: Advantages, Issues, and Perspectives,” Springer Nature Switzerland AG (2018) https://doi.org/10.1007/978-3-030-04834-1_9

Exhibit No.	Description
1029	Singh, “Overview of Attacks on Cloud Computing,” Int’l. J. of Eng. And Innovative Technology (IJEIT), Vol. 1, Issue 4 (Apr. 2012)
1030	Bairwa, “Vulnerability Scanners: A Proactive Approach to Assess Web Application Security,” Int’l. J. on Computational Sciences & Applications (IJSCA), Vol. 4, No. 1 (Feb. 2014)
1031	Stone, “IT Asset Management,” NIST Special Publication 1800-5 (Sept. 2018) http://doi.org/10.6028/NIST.SP.1800-5
1032	EP Patent No. 3,360,071 B1 to Li
1033	Garfinkel, “A Virtual Machine Introspection Based Architecture for Intrusion Detection,” Computer Science Dept., Stanford University (2003)
1034	Tang, “VirtAV: An Agentless Runtime Antivirus System for Virtual Machines,” KSII Transactions on the Internet and Information Systems, Vol. 11, No. 11 (Nov. 2017) https://doi.org/10.3837/tiis.2017.11.026
1035	Lengyel, “Scalability, Fidelity and Stealth in the DRAKVUF Dynamic Malware Analysis System (2014) https://dl.acm.org/doi/pdf/10.1145/2664243.2664252
1036	U.S. Patent No. 8,613,080 B2 to Wysopal
1037	Bauman, “A Survey on Hypervisor-Based Monitoring: Approaches, Applications, and Evolutions,” ACM Computing Surveys, Vol. 48, No. 1, Article 10 (Aug. 2015)
1038	Suneja, “Exploring VM Introspection: Techniques and Trade-offs,” ACM, VEE’15 (Mar. 2015) https://dl.acm.org/doi/abs/10.1145/2817817.2731196
1039	Payne, “Simplifying Virtual Machine Introspection Using LibVMI,” SANDIA Report-SAND2012-7818, Sandia National Labs. (Sep. 2012)

Exhibit No.	Description
1040	U.S. Publication No. 2013/0111018 A1 to Ammons
1041	U.S. Patent No. 5,893,131 to Kornfeld
1042	U.S. Patent No. 7,975,305 B2 to Rubin
1043	Jiang, “Stealthy Malware Detection and Monitoring Through VMM-Based “Out-of-the-Box” Semantic View Reconstruction,” ACM Transactions on Information and System Security, Vol. 13, No. 2, Article 12 (Feb. 2010)
1044	U.S. Patent No. 9,141,805 B2 to Giakouminakis
1045	U.S. Publication No. 2019/0379700 A1 to Canzanese Jr.
1046	U.S. Patent No. 10,783,241 B2 to Crabtree
1047	Bourquin, “Binslayer: Accurate Comparison of Binary Executables,” 2nd ACM SIGPLAN Program Protection and Reverse Engineering Workshop (2013)
1048	U.S. Publication No. 2013/0247133 A1 to Price
1049	Huseinović, “Virtual machine memory forensics,” 2013 21st Telecommunications Forum Telfor (TELFOR) (2013)
1051	Using VMware Workstation Pro: VMware Workstation Pro 14.0, VMware, Inc. (December 17, 2017)
1052	VMware VirtualCenter User’s Manual version 1.4, VMware, Inc. (July 7, 2006)
1053	U.S. Publication No.2010/0070725 A1 to Prahlad
1054	U.S. Patent No. 9,830,182 B2 to Ackley
1055	Virtual Disk Programming Guide: vSphere Storage APIs for Data Protection (VADP) 6.0 & Virtual Disk Development Kit (VDDK) 6.0, VMware, Inc. (March 25, 2016)

Exhibit No.	Description
1056	U.S. Publication No. 2015/0033221 A1 to Chari
1057	Archived copy of “CWE-200: Information Exposure,” archived on January 26, 2019, by the Internet Archive’s “Wayback Machine” https://web.archive.org/web/20190126150012/https://cwe.mitre.org/data/definitions/200.html
1058	Archived copy of “CWE-311: Missing Encryption of Sensitive Data,” archived on January 17, 2019, by the Internet Archive’s “Wayback Machine” https://web.archive.org/web/20190117150049/https://cwe.mitre.org/data/definitions/311.html
1059	Archived copy of “CWE-359: Exposure of Private Information (‘Privacy Violation’),” archived on January 13, 2019, by the Internet Archive’s “Wayback Machine” https://web.archive.org/web/20190113052451/https://cwe.mitre.org/data/definitions/359.html
1063	U.S. Publication No. 2014/0245376 A1 to Hibbert
1064	U.S. Publication No. 2017/0180318 A1 to Lutas
1065	U.S. Patent No. 9,900,333 B2 to Thakar
1066	Zhou, “Always Up-to-date—Scalable Offline Patching of VM Images in a Compute Cloud,” 26th annual computer security applications conference (2010)
1067	Jeswani, “ImageElves: Rapid and Reliable System Updates in the Cloud,” 2013 IEEE 33rd International Conference on Distributed Computing Systems (2013)
1068	U.S. Patent No. 8,984,478 B2 to Epstein
1069	U.S. Patent No. 11,061,682 B2 to Vorbach
1070	U.S. Publication No. 2013/0024940 A1 to Hutchins

Exhibit No.	Description
1071	U.S. Patent No. 9,875,160 B2 to Bezbaruah
1072	U.S. Patent No. 10,536,471 B1 to Derbeko
1073	Ammons, “Virtual machine images as structured data: the Mirage image library,” 3rd USENIX conference on Hot topics in cloud computing (2011)
1074	VMware Infrastructure Architecture Overview White Paper, VMWare, Inc. (2006)
1075	U.S. Patent No. 9,692,778 B1 to Mohanty
1076	U.S. Patent No. 10,055,576 B2 to Milner
1077	U.S. Patent No. 9,824,216 B1 to Khalid
1078	U.S. Publication No. 2020/0097662 A1 to Hufsmith
1079	Carroll, “Challenges in Modern Digital Investigative Analysis,” published in Forensic Science and Forensic Evidence I, U.S. Att’y’s Bull. Vol. 65 No. 1 (January 2017)
1080	U.S. Patent No. 7,831,739 B2 to Lambeth
1081	U.S. Publication No. 2010/0169948 A1 to Budko
1084	U.S. Patent No. 9,749,349 B2 to Czarny
1085	U.S. Publication No. 2017/0048266 A1 to Hovor
1090	Archived copy of “Vulnerability Details: CVE-2006-2923,” published by CVE Details and archived on July 29, 2014, by the Internet Archive’s “Wayback Machine” https://web.archive.org/web/20140729171850/https://www.cvedetails.com/cve/CVE-2006-2923/
1091	Archived copy of “Vulnerability Details: CVE-2007-0018,” published by CVE Details and archived on August 9, 2014, by the Internet Archive’s “Wayback Machine”

Exhibit No.	Description
	https://web.archive.org/web/20140809144720/https://www.cvedetails.com/cve/CVE-2007-0018/
1095	U.S. Publication No. 2017/0372070 A1 to Burdett
1100	Archived copy of “Vulnerability Details: CVE-2018-6954,” published by CVE Details and archived on December 11, 2018, by the Internet Archive’s “Wayback Machine” https://web.archive.org/web/20181211090742/http://www.cvedetails.com/cve/CVE-2018-6954
1101	Archived copy of “CVE-ID: CVE-2018-15687,” published by MITRE Corp. and archived on December 11, 2018, by the Internet Archive’s “Wayback Machine” https://web.archive.org/web/20181211082546/https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-15687
1102	Archived copy of “CVE-ID: CVE-2018-1312,” published by MITRE Corp. and archived on January 5, 2019, by the Internet Archive’s “Wayback Machine” https://web.archive.org/web/20190105194958/https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-1312
1103	Archived copy of “Vulnerability Details: CVE-2013-2765,” published by CVE Details and archived on July 20, 2014, by the Internet Archive’s “Wayback Machine” https://web.archive.org/web/20140720123338/https://www.cvedetails.com/cve/CVE-2013-2765
1105	Archived copy of “Defense Privacy, Civil Liberties, and Transparency Division, U.S. Department of Defense, FAQs,” archived on May 17, 2018, by the Internet Archive’s “Wayback Machine” at https://web.archive.org/web/20180517201830/http://dpcl.d.defense.gov/Privacy/About-the-Office/FAQs .
1111	U.S. Patent No. 10,498,845 B2 to Kapoor

Exhibit No.	Description
1112	Archived copy of “Linux Programmer’s Manual: proc–process information pseudo-filesytem,” published by Man7 and archived on December 15, 2018, by the Internet Archive’s “Wayback Machine” at https://web.archive.org/web/20181215072140/http://www.man7.org:80/linux/man-pages/man5/proc.5.html
1113	Archived copy of “PID Definition,” published by The Linux Information Project and archived on November 1, 2018, by the Internet Archive’s “Wayback Machine” at https://web.archive.org/web/20181101094536/https://www.linfo.org/pid.html
1114	Archived copy of “Unix/Linux–Processes Management,” published by TutorialsPoint and archived on December 17, 2018, by the Internet Archive’s “Wayback Machine” at https://web.archive.org/web/20181217170426/https://www.tutorialspoint.com/unix/unix-processes.htm
1116	U.S. Patent No. 9,524,389 B1 to Roth
1117	Archived copy of “20 Log Files that are located under /var/log Directory,” archived on October 13, 2018, by the Internet Archive’s “Wayback Machine” at https://web.archive.org/web/20181013220819/https://www.thegeekstuff.com/2011/08/linux-var-log-files/ .
1118	Archived copy of “Linux YUM Tutorial,” archived on November 10, 2017, by the Internet Archive’s “Wayback Machine” at https://web.archive.org/web/20171110120546/https://linuxmoz.com/linux-yum-tutorial/ .
1119	U.S. Patent No. 11,093,629 B2 to Hiltgen
1120	U.S. Publication No. 2020/0026527 A1 to XUE
1121	U.S. Patent No. 10,579,355 B2 to Bo