

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

WIZ, INC.,
Petitioner,

v.

ORCA SECURITY LTD,
Patent Owner.

Case IPR2024-00220
Patent No. 11,431,735

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	8
	A. Virtualization	9
	B. Snapshots	19
	C. Cloud Computing.....	24
	D. Cyber Security	27
	1. Types of Threats and Vulnerabilities	27
	2. Detecting Threats and Vulnerabilities	35
	3. Responding to Threats and Vulnerabilities	44
IX.	OVERVIEW OF THE '735 PATENT	49
	A. Challenged Claims.....	52
	B. Prosecution History of the '735 Patent.....	52
X.	CLAIM CONSTRUCTION	53
	A. Determining/Determine a “Location” of a Snapshot.....	53
	B. “Analyzing the Snapshot”	54

XI.	OVERVIEW OF THE PRIOR ART.....	56
A.	Veselov (U.S. Patent No. 11,216,563, EX1007)	56
B.	Basavapatna (U.S. Pub. No. 2013/0191919, EX1008).....	60
XII.	GROUND 1: CLAIMS 1-7, 9-17, AND 19 WERE OBVIOUS OVER VESELOV AND BASAVAPATNA	62
A.	Independent Claims	63
1.	Independent Claim 1.....	63
2.	Independent Claim 10.....	101
3.	Independent Claim 11.....	104
4.	Reasons Why a Person of Ordinary Skill in the Art Would Have Combined Veselov and Basavapatna	109
B.	Dependent Claims.....	115
1.	Dependent Claims 2 and 12.....	115
2.	Dependent Claims 3 and 13.....	122
3.	Dependent Claims 4 and 14.....	124
4.	Dependent Claims 5 and 15.....	134
5.	Dependent Claims 6 and 16.....	145
6.	Dependent Claims 7 and 17.....	152
7.	Dependent Claims 9 and 19.....	157
XIII.	CONCLUDING STATEMENTS	159
XIV.	APPENDIX A – MATERIALS CITED	161

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. 11,431,735 (the “’735 patent”) and on the patentability of claims 1-7, 9-17, and 19 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. 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 '735 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 '735 patent and the '735 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 '735 patent issued from U.S. Application No. 16/585,967 ("the '967 Application"), which claimed priority to Provisional Application No. 62/797,718, filed on January 28, 2019. EX1001, Face. The '735 patent does not claim priority to any other patent applications. *Id.* I have been asked to assume that the effective filing date for the claims at issue here is January 28, 2019.

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 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 person of ordinary skill in the art as of 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

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.

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 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

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>.

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>.

physical hardware such as a CPU or motherboard in order to run as independent entities. Instead, they can run inside a virtual machine (VM).”); EX1012 (Microsoft Computer Dictionary), 556 (“VM n. Acronym for Virtual Machine. An operating system for IBM mainframes that provides virtual-machine capability. VM was developed by IBM customers and later taken over by IBM itself under the name OS/VM.”).⁴ A virtual machine 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 different virtualization solutions and products, spanning both closed and open-source, available for desktop, server, and cloud applications.

⁴ EX1012 includes excerpts from Microsoft Computer Dictionary, 5th edition.

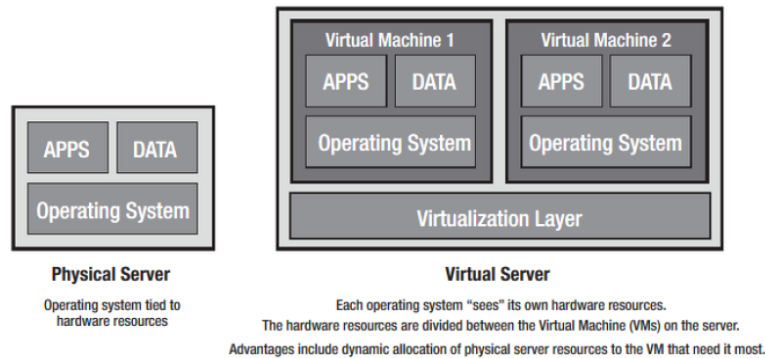
To avoid overburdening the record, since the dictionary is quite large, EX1012 only includes the front matter and the pages to which I cite.

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), 229.⁵

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



EX1009 (Wolf), 505, Figure 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

⁵ 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.

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 in-creasing the resources allocated to a virtual server").

30. Computer data was 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, 242, 246-

57; EX1010 (Waldspurger), 3-4 (discussing exemplary mappings between virtual and non-virtual locations). One common way of referencing virtual data was a “virtual address.” Virtual machines used “virtual addresses” to map the location of data as it appeared inside a virtual machine to a physical location on a disk. EX1012 (Microsoft Computer Dictionary), 553 (“virtual address n. In a virtual memory system, the address that the application uses to reference memory. The memory management unit (MMU) translates this address into a physical address before the memory is actually read or written to.”); 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 address and a physical location on disk, mapped together by the virtualization layer software, as illustrated in the following example:

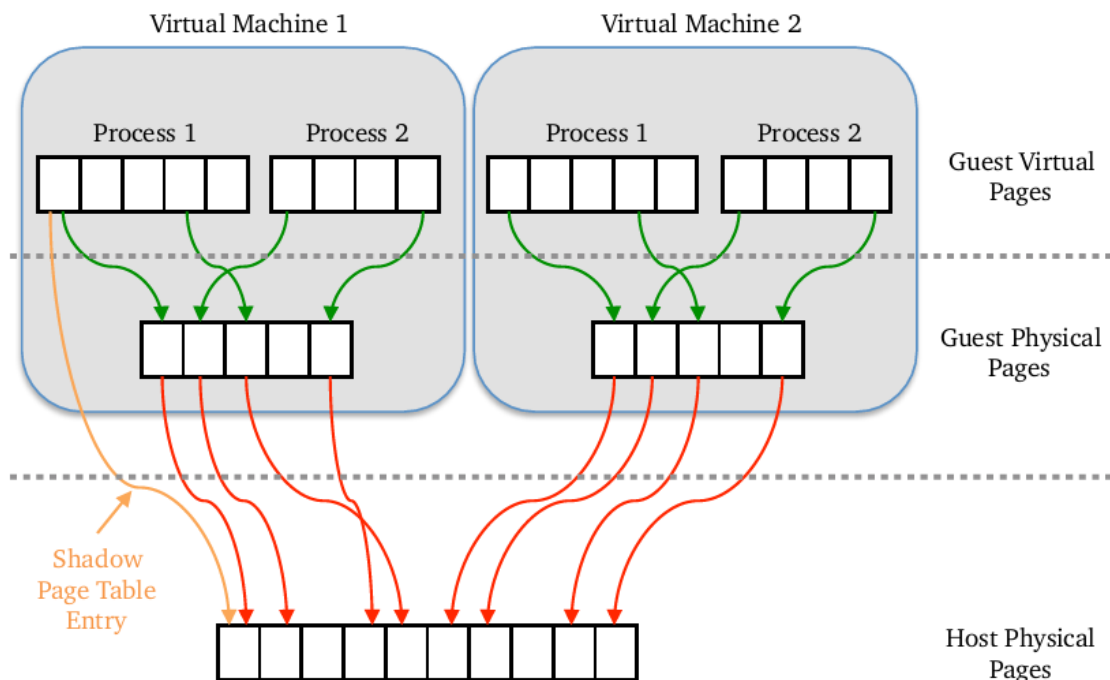


Figure 2.1: Depiction of two-level page mapping for two virtual machine guests, their processes on the same host, and shadow page table mappings. Figure borrowed with modification from VMware Performance Evaluation of Intel EPT Hardware Assist. http://protect.kern+.2222em/relax/www.vmware.com/pdf/Perf_ESX_Intel-EPT-eval.pdf

EX1014 (Klempereer), 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

⁶ EX1014 is a copy of “Efficient Hypervisor Based Malware Detection”, by P. Klempereer, 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 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

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>.

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* EX1017 (Rosendahl), 1:20-35 (discussing operation and benefits of containers).

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

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>.

machine since at least 2005, well before the '735 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

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

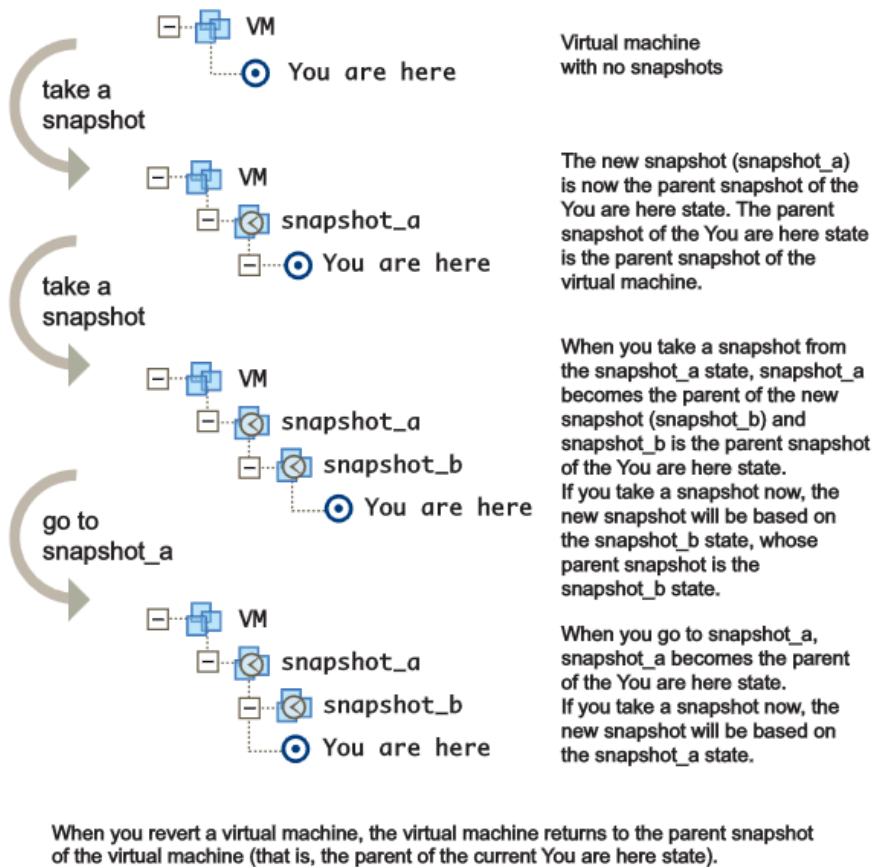
generation, storage, indexing, and import of snapshots. EX1019 (Rawat), Abstract; *see also* EX1020 (Sancheti), Abstract, 21:42-22:58 (describing use of API calls to interact with snapshots).

33. Snapshots could contain important and sensitive data that was 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 virtual machine they were taken from (including the machine’s memory), they could be used to provide insight into vulnerabilities of the virtual machine.

34. A snapshot could be comprised of one or more files, depending on the mechanism that was used to generate the snapshot. 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 in 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).

35. 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

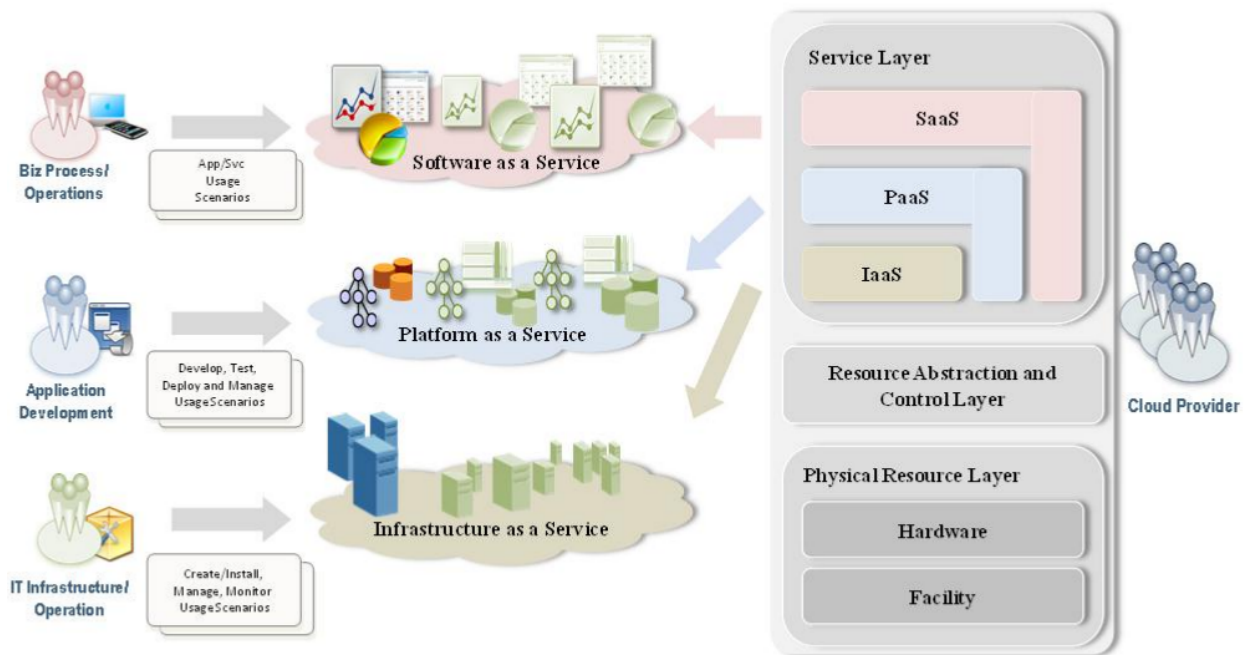
accompanied the virtual machine snapshot when the snapshot was generated. EX1015 (VMware vSphere), 164-66.

C. Cloud Computing

36. 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.”).¹⁰

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

¹⁰ EX1021 is a copy of a July 2013 NIST publication obtained from <http://dx.doi.org/10.6028/NIST.SP.500-291r2>.



EX1021 (NIST Cloud Computing), 18.

38. “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;¹¹ EX1015 (VMware vSphere), 55-58, 62-66; EX1021 (NIST Cloud Computing), 94-95. Through APIs, GUIs, or other consoles, a user could control and manage their cloud resources, for example, by sending queries and commands to a hypervisor. EX1015 (VMware vSphere), 164-66.

39. “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.

¹¹ EX1022 is a copy of “Public Cloud Service Definition,” by A. Roberts, published in January 2018 by VMware, Inc. This copy was obtained from <https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/vcat/vmware-public-cloud-service-definition.pdf>.

40. “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 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.”).

D. Cyber Security

41. Cyber security is 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 are detected and the common tools used to do so. Finally, I will discuss responses to threats and vulnerabilities once they are detected.

1. Types of Threats and Vulnerabilities

42. It was well known that computer systems were subjected to various types of cyber threats, 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.”).¹² 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), 1.¹³ Old, outdated software often contained bugs that, if not patched, could be exploited. EX1023 (NIST Security 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

¹² EX1023 is a copy of an October 2002 NIST publication obtained from https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=150702.

¹³ EX1024 is a copy of a December 2005 NIST publication obtained from <https://csrc.nist.gov/csrf/media/publications/shared/documents/itl-bulletin/itlbul2005-12.pdf>

often releases a piece of software to correct the bug. This software is often called a patch, hotfix, or service pack.”).

43. 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.’’).¹⁴ 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

¹⁴ EX1025 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), 9.¹⁵

44. 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 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

¹⁵ EX1026 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.

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.”).

45. 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:

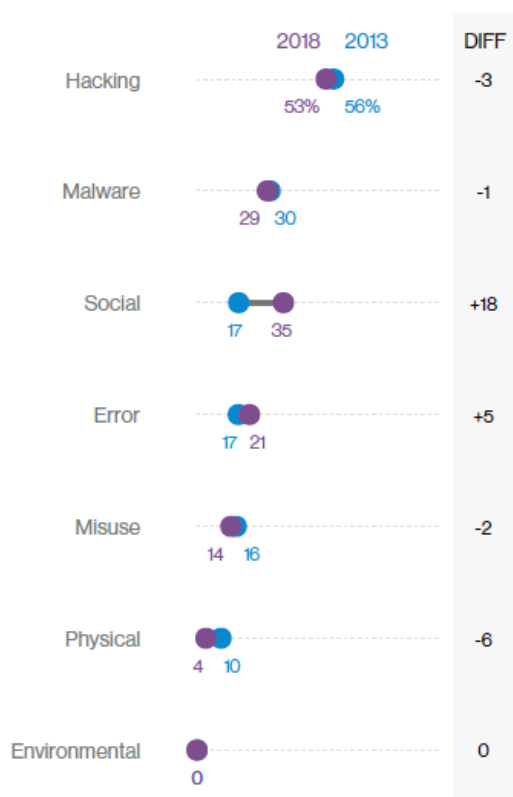


Figure 9. Threat actions in data breaches over time
n=2,501 (2013), n=1,638 (2018)

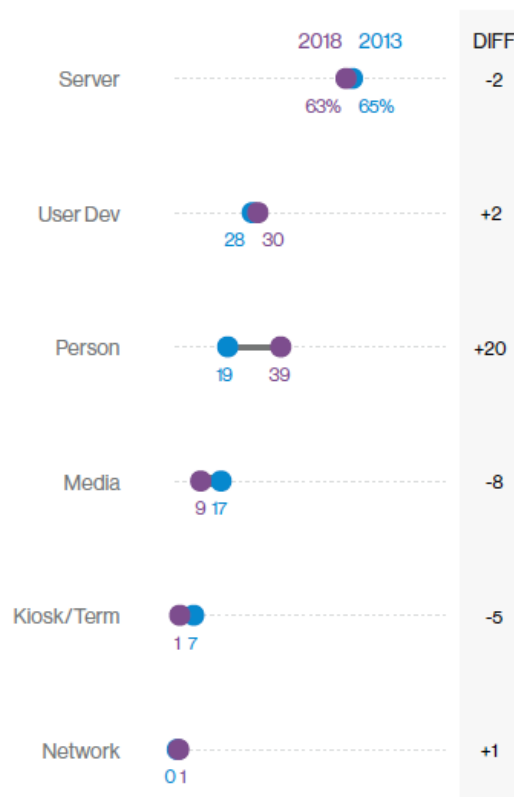


Figure 10. Asset categories in data breaches over time
n=2,294 (2013), n=1,513 (2018)

EX1027 (Verizon DBIR), 8.¹⁶ Cloud computing systems were known to have many of the same vulnerabilities as non-cloud systems, while also presenting additional

¹⁶ EX1027 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), 6-7.¹⁷ 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.

¹⁷ EX1028 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), 1 (“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 Threats and Vulnerabilities

46. Users and administrators routinely sought to identify the types of threats and vulnerabilities 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 threats. *See* EX1031 (NIST IT Asset Mgmt), 22;¹⁸ EX1032 (Li), ¶¶27, 43; EX1033 (Garfinkel), 2;¹⁹ EX1030 (Bairwa), 116.

47. 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

¹⁸ EX1031 is a copy of a September 2018 NIST publication obtained from <http://doi.org/10.6028/NIST.SP.1800-5>.

¹⁹ EX1033 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/>.

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.

48. 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), 2 (“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), 2 (discussing hybrid approach); EX1035 (Lengyel), 389 (distinguishing its

techniques from “prior agentless VMI approaches”).²⁰ 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) (surveying 64 different out-of-VM approaches); EX1038 (Suneja) (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

²⁰ EX1035 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>.

security monitoring, intrusion detection, anti-malware, firewall, virus scanning, cloud-wide information flow tracking, and policy enforcement. EX1038 (Suneja).²¹ 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:

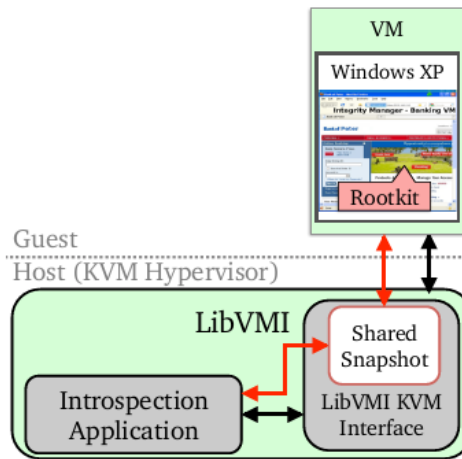


Figure 4.1: This block diagram illustrates how the shared snapshot interface provides the Introspection Application with a view into the memory of the VM Guest.

²¹ EX1038 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>.

EX1014 (Klemperer), 25.

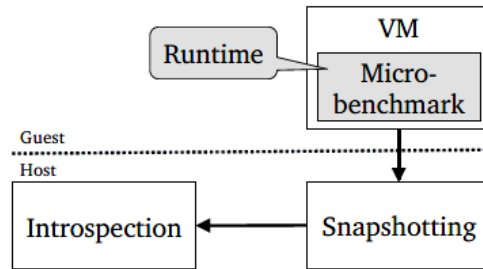
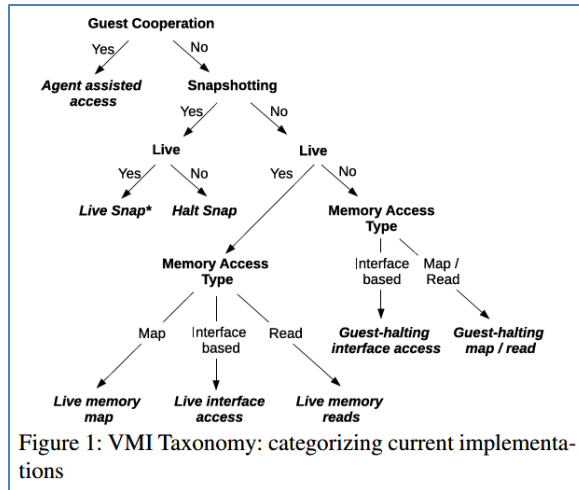


Figure 6.1: Microbenchmarks are used to quickly evaluate the effect of snapshotting over various snapshotting regimes, guest loads, and introspection loads.

EX1014 (Klemperer), 55; *see also* EX1039 (Payne), 9 (“LibVMI programs can also use a static memory snapshot as a data source.”);²² EX1038 (Suneja), 2-3 (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:

²² EX1039 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), 2; *see also* EX1040 (Ammons), Abstract, ¶43 (describing use of snapshots to perform security analysis of virtual machines).

49. 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.”).

50. 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.”).

51. Scanners also routinely looked for signatures of known malicious code. *Id.*, 5:28-32 (“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.”).

52. 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.”).

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

3. Responding to Threats and Vulnerabilities

54. When a scanner detected a threat or vulnerability, 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.”).

55. The 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.”).

56. 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.").

57. 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....”).

58. Alerts were commonly prioritized in a variety of ways. 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.”).

59. 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.”).

60. 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. EX1025 (NIST Incident Handling), 24 (“All hosts should be hardened appropriately using standard configurations. In addition to keeping each host properly patched, hosts should be configured to follow the principle of least privilege—granting users only the privileges necessary for performing their authorized tasks.”); 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.”).

IX. OVERVIEW OF THE '735 PATENT

61. The '735 Patent is entitled “Techniques For Securing Virtual Machines.” EX1001, Face. The '735 Patent discloses techniques “for securing virtual cloud assets in a cloud computing environment against cyber threats.” EX1001, Abstract. As part of these techniques, it describes determining “a location of a snapshot of at least one virtual disk of a protected virtual cloud asset.” *Id.* This snapshot is then analyzed to “detect potential cyber threats.” *Id.* Once any threats

are detected, they are prioritized and the system alerts the user of “detected potential cyber threats based on a determined priority.” *Id.* Priority is determined using various factors such as, the use of vulnerable applications or the sensitivity of vulnerable data. *Id.*, 5:43-47 (“In yet another embodiment, the security system 140 is configured to verify whether the vulnerability is relevant to the VM 119. For example, if there is a vulnerable version or module not in use, the priority of that issue is reduced dramatically”), 6:46-53 (“In an embodiment, any detected cyber threats related to sensitive data (including personally identifiable information, PII) is reported at a higher priority. In an embodiment, such data is determined by searching for the PII, analyzing the so application logs to determine whether the machine accessed PII/PII containing servers, or whether the logs themselves contain PII, and searching the machine memory, as reflected in the page file, for PII.”).

62. Figures 1A and 1B of the ’735 patent show network diagrams used to describe the various embodiments. EX1001, Figs. 1A and 1B (below). “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.” EX1001, 3:45-51. The figures sometimes depict a single

VM in a single cloud environment for simplicity, but multiple VMs may be deployed without departing from the scope of the disclosure. EX1001, 6:66-7:10.

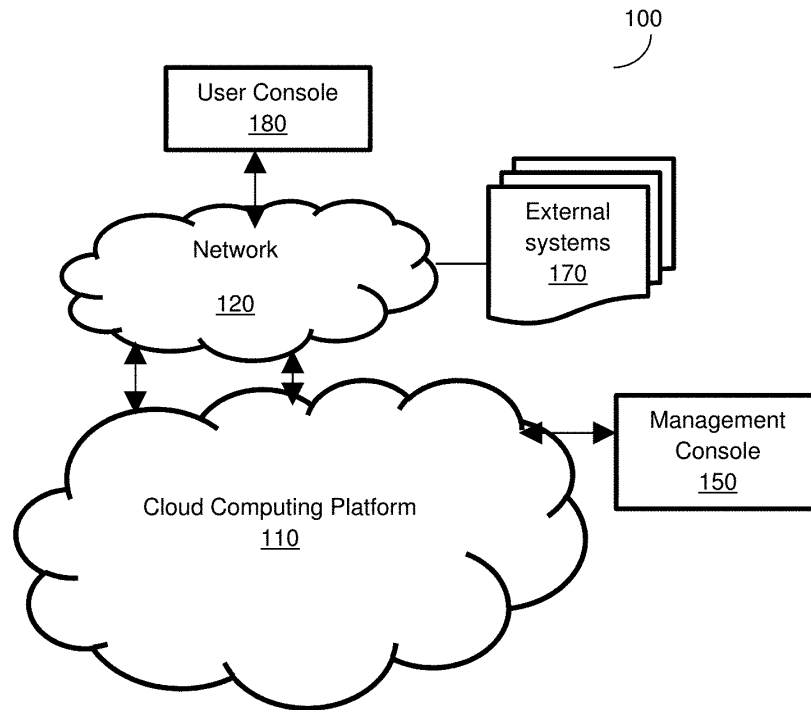


FIG. 1A

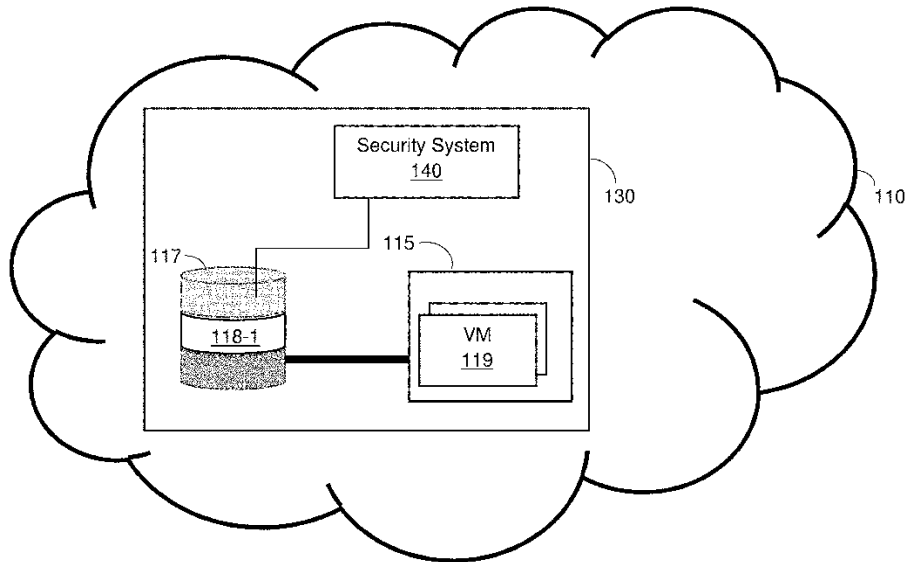


FIG. 1B

Id., Figs. 1A-B.

A. Challenged Claims

63. The '735 patent has 19 claims, with claims 1, 10, and 11 being independent. I understand that each of claims 1-7, 9-17, and 19 is being challenged in the *inter partes* review proceedings related to this declaration. Below, I reproduce the text of each of the challenged claims, with numbering added to certain claim elements for ease of reference.

B. Prosecution History of the '735 Patent

64. I understand that the application that issued as the '735 patent never received a rejection based on anticipation under §102 or obviousness under §103. I also understand that the first office action provisionally rejected the claims based on statutory double patenting but indicated that the claims were otherwise allowable.

EX1004 ('735 File History), 79-80. The '735 patent issued after the applicant amended two claims “for clarity,” canceled two other claims, and filed a terminal disclaimer. *Id.*, 115, 127, 139. I understand that the Notice of Allowance does not identify any particular limitation as supposedly missing in the art considered by the Examiner; instead, it lists several references identified as the closest art and states that these references do not teach the independent claims as a whole. *See* EX1004 ('735 File History), 139-41.

X. CLAIM CONSTRUCTION

65. 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 person of ordinary skill in the art 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.

A. Determining/Determine a “Location” of a Snapshot

66. Independent claims 1 and 10 each recite “determining a location of a snapshot.” Independent claim 11 similarly recites “determine a location of a snapshot.”

67. In my opinion, a person of ordinary skill in the art reading the claims in view of the specification and prosecution history would have understood that determining a “location” of a snapshot broadly encompasses determining a virtual

location of a snapshot and determining a non-virtual location of a snapshot. This understanding is consistent with the ordinary and customary meaning of “location” in this context (*supra*, §VIII.A-B) and confirmed by the specification of the ’735 patent. *See, e.g.*, EX1009 (Wolf), 242, 246-57; EX1010 (Waldspurger), 3-4; EX1015 (VMWare vSphere), 56; EX1021 (NIST Cloud Computing), 8. While discussing Figure 1, the specification of the ’735 patent states that “the management console 150 may be queried, by the security system 140, about as **the location (e.g., virtual address)** of the virtual disk 118-1 in the storage 117.” EX1001, 4:27-30 (emphasis added). This parenthetical would have made it clear to a person of ordinary skill that “location” in this context expressly encompasses a “virtual address,” and the use of “e.g.” here also would have made it clear that the named inventors did not intend to limit “location” in this context to virtual addresses exclusively.

68. Accordingly, in my opinion a person of ordinary skill in the art would have understood determining “a location” of a snapshot, as recited in independent claims 1, 10, and 11, as at least encompassing both determining a virtual location of a snapshot and determining a non-virtual location of a snapshot.

B. “Analyzing the Snapshot”

69. Independent claims 1 and 10 each recite “analyzing the snapshot,” and independent claim 11 similarly recites “analyze the snapshot.”

70. In my opinion, a person of ordinary skill in the art would have understood the ordinary and customary meaning of “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 ’735 specification. *See, e.g.*, EX1001, 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 (“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”).

71. I have also been asked to assume that “analyzing the snapshot” encompasses analysis of a VM instantiated from the snapshot.

72. Accordingly, my discussion of Ground 1 below applies an understanding of “analyzing the snapshot” in claims 1 and 10, as well as “analyze the snapshot” in claim 11, as encompassing both direct analysis of the snapshot data and analysis of a VM instantiated from snapshot. *Infra*, §§XII.A.1.d, XII.A.2.d, XII.A.3.f.

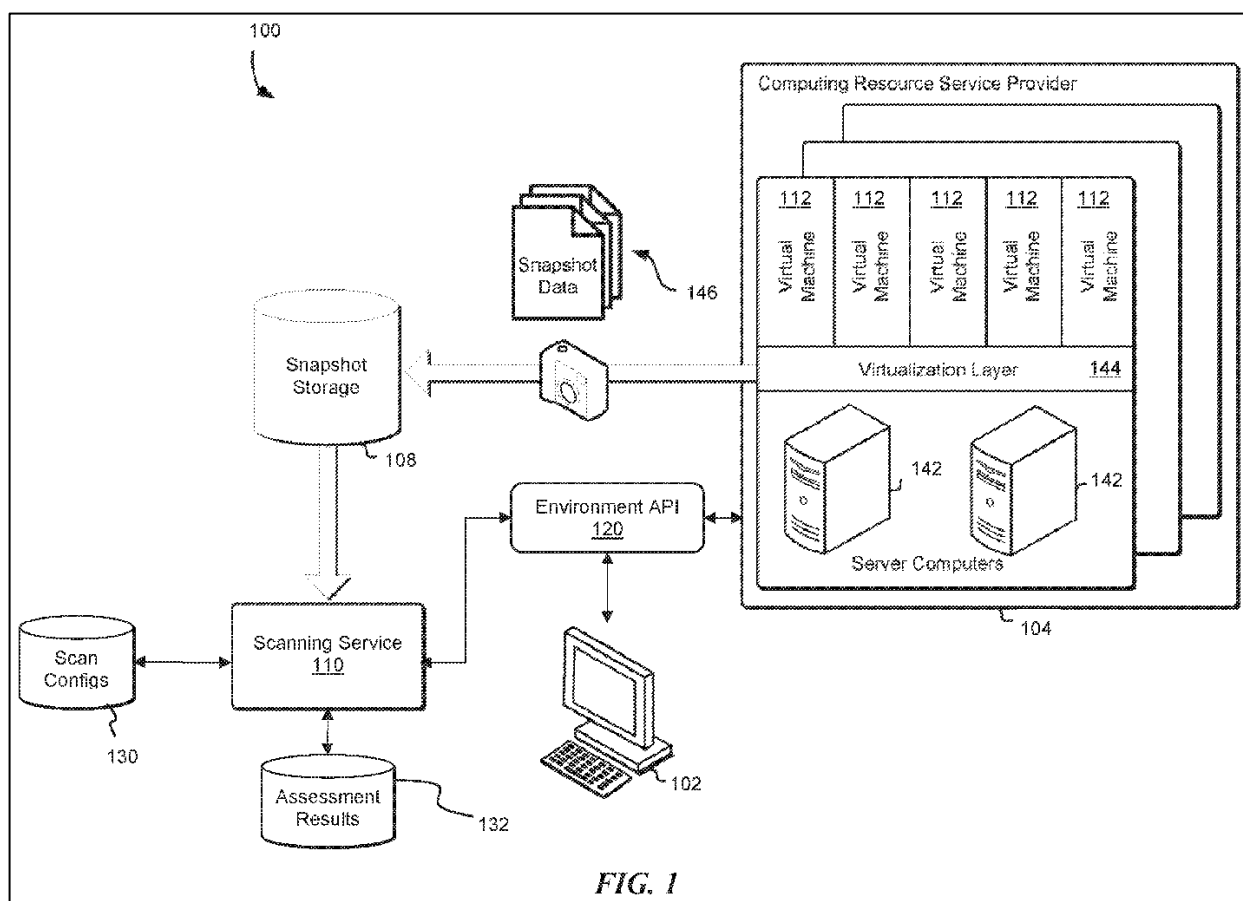
XI. OVERVIEW OF THE PRIOR ART

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

73. Veselov, a patent titled “Security Assessment of Virtual Computing Environment Using Logical Volume Image,” describes methods and “[s]ystems for performing a security assessment of a target computing resource, such as a virtual machine or an instance of a virtual machine, includ[ing] a scanning service that facilitates duplication of all or a portion of the target computing resource, and then performs the security assessment on the duplicate computing resource ...” EX1007 (Veselov), Abstract. This “duplicate” can be generated from a snapshot of a resource such as a logical volume (e.g., a virtual disk allocated to a VM). *Id.*, Abstract, 3:20-40, 11:28-33, Figs. 2, 4, 6. As Veselov explains, performing the security assessment on a snapshot, rather than the target computing resource itself, “avoid[s] consuming processing time, processing power, and storage space of the target computing resource.” *Id.*, Abstract. Veselov uses its security assessment to identify various types of threats and vulnerabilities. *Id.*, 1:41-49 (known security assessment services “analyze the behavior of computing resources to identify vulnerabilities, bad configurations, and the like”), 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

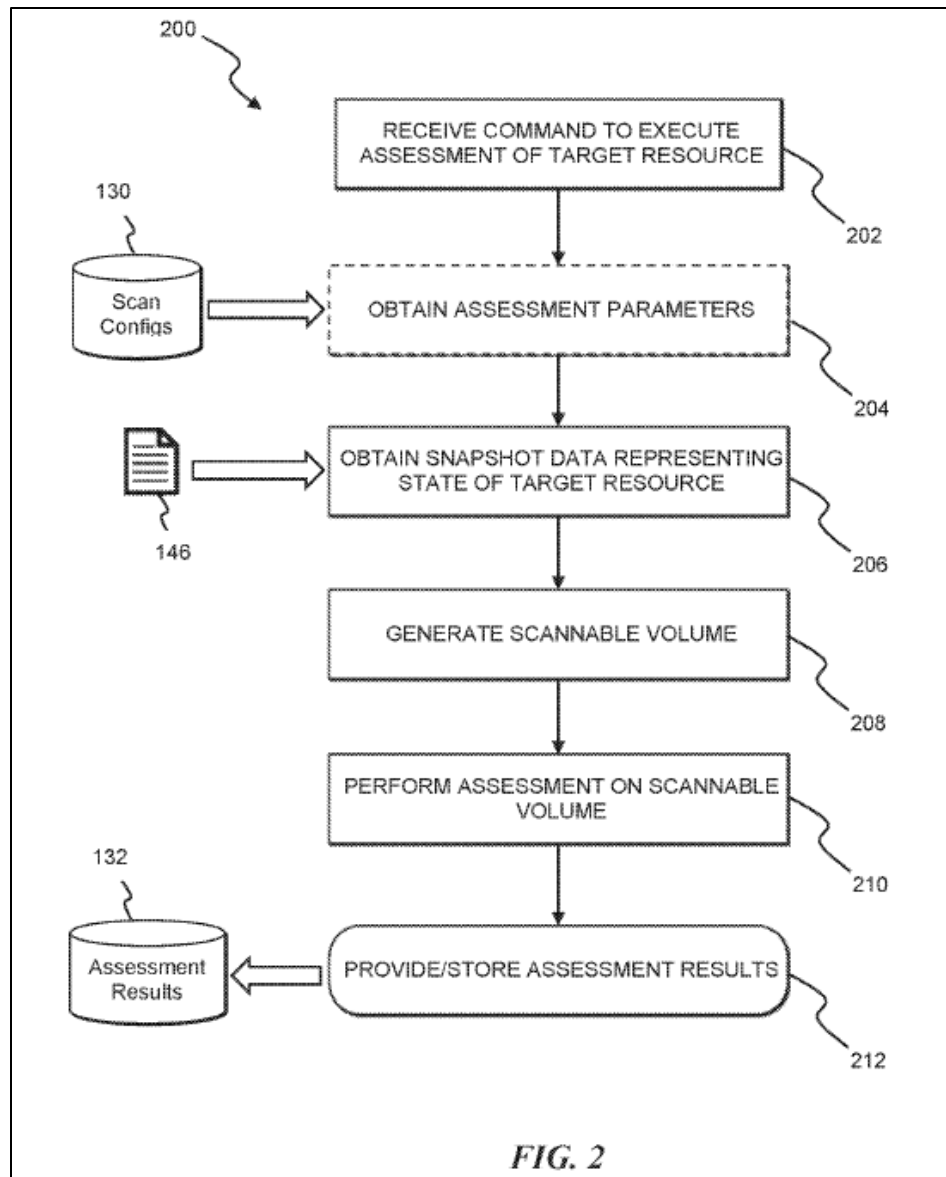
(CVEs), Center for Internet Security (CIS) benchmarks, ‘best practices’ packages, static or runtime behavior analysis, host configuration assessments, and the like.”).

74. Veselov notes that its techniques are particularly useful in cloud computing environments such as data centers. EX1007 (Veselov), 2:22-3:19. It also describes applying its techniques to distributed, virtualized systems commonly for cloud computing:



Id., Fig. 1; *see also id.*, 1:5-50 (discussing virtualization in data centers); *supra*, §§VIII.A, VIII.B, VIII.C (discussing cloud computing and virtualization).

75. Veselov describes several types of security analysis systems. 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. EX1007 (Veselov), 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:



EX1007 (Veselov), Fig. 2; *see also id.*, 8:52-10:36 (describing Fig. 2), Figs. 4, 6.

76. 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),

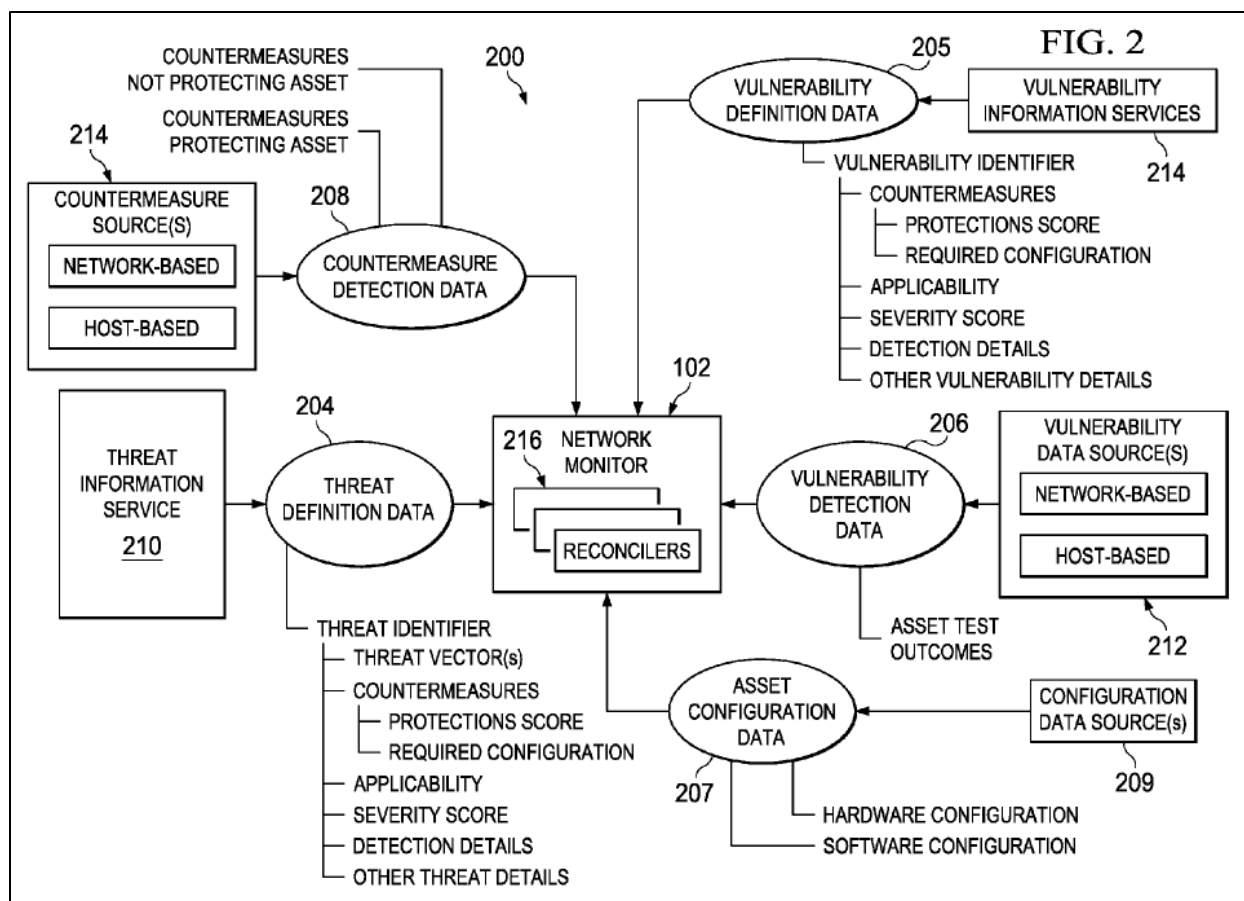
claim 2 (same), Fig. 3B (depicting security agent 330). Veselov also describes analysis techniques that do not rely on such agents, including agentless analysis of the assessment VM and analysis of the snapshot itself as a data file (without instantiating an assessment VM). *See, e.g., id.*, 3:61-67 (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).

77. 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.” EX1007 (Veselov), 10:32-36.

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

78. Basavapatna describes examples of the commonplace technique of prioritizing security alerts. EX1008 (Basavapatna), Abstract; *supra*, §VIII.D.3 (discussing alert prioritization). For example, Basavapatna discusses prioritizing

alerts with risk metrics calculated from vulnerability scores. EX1008 (Basavapatna), Abstract, ¶¶138-40. These scores are determined by assessing, for example, the severity of a vulnerability, the probability that the vulnerability affects a particular asset, and the probability that a countermeasure would mitigate the risk to the asset posed by the vulnerability. *Id.*, claim 1. “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.” *Id.*, ¶138. Basavapatna also describes alerting users regarding the identified, prioritized risks in a variety of ways, including reporting the ten greatest risks, only reporting items above a minimum risk threshold, and only reporting risks for a particular asset. *Id.*, ¶¶138-39, Fig. 6. The following figure depicts exemplary system components and various factors that can be evaluated as part of the security analysis and alerting process:



EX1008 (Basavapatna), Fig. 2.

XII. GROUND 1: CLAIMS 1-7, 9-17, AND 19 WERE OBVIOUS OVER VESELOV AND BASAVAPATNA

79. As I explain below, it is my opinion that claims 1-7, 9-17, and 19 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.

80. Veselov teaches all aspects of independent claims 1, 10, and 11 except that it does not expressly discuss the “determined priority” aspect of elements 1.4,

10.4, and 11.4. Veselov generally teaches locating or generating a snapshot of a virtual cloud asset, including its virtual disks. Next, Veselov then performs a security assessment by scanning the snapshot data. Finally, Veselov reports assessment results to the system's user.

81. While Veselov does not expressly state that detected potential cyber threats should be alerted “based on a determined priority” as recited in elements 1.4, 10.4, and 11.4, prioritized alerts were well known and routinely implemented at the time. For example, Basavapatna describes a set of scores calculating the risk of a type of threat or vulnerability in general, the risk of the threat or vulnerability being present in the protected system, and the potential for a known countermeasure or remediation technique to alleviate the risk posed, then combines these scores into an overall priority score for each alert. EX1008 (Basavapatna), Abstract. As I explain below in §XII.A.4, a person of ordinary skill in the art had reason to combine the teachings of Veselov and Basavapatna to arrive at the claims as recited.

A. Independent Claims

1. Independent Claim 1

a. Claim 1 Preamble

82. The preamble of claim 1 recites “[a] method for securing virtual cloud assets in a cloud computing environment against cyber threats, comprising....” As I explain below, Veselov teaches this preamble.

83. Veselov teaches securing several types of computing assets against cyber threats, including “virtual” assets. Veselov discloses systems and methods 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 (emphasis added). A person of ordinary skill would have understood such virtual machines to be “virtual cloud assets” as recited. Veselov also explains that such assessments identify various types of threats and vulnerabilities associated with these assets. *Id.*, 2:37-41 (security assessment “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”), 4:1-8 (“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.”), 7:1-5 (scanning service may search for changes to settings “that exposed or repaired a security vulnerability”). For example, rules packages such as the CVEs mentioned in Veselov typically included a set of specific threats and vulnerabilities that might comprise the security of computer system. A person of ordinary skill would have understood that a computer security system that identified vulnerabilities would also protect “against cyber threats,” as recited, since such threats might exploit those

vulnerabilities. Veselov also describes providing the results of the assessments to users, as well as identifying and providing remedial actions for the identified vulnerabilities. *Id.*, 10:17-36 (example security process includes “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”). Veselov thus teaches securing virtual assets against cyber threats.

84. Veselov also teaches 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), 2:42-63 (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 (“[I]n some embodiments it may be suitable or advantageous to perform the security assessment on a different but similar virtual machine than the customer’s virtual machine because the assessment can be performed without impacting the customer’s virtual machine, which may be servicing requests from its customers or performing other

useful work. For example, the virtual computing environment 304A may be a production environment in which allocating virtual computing resources to the security assessment may degrade operation of the virtual machine 312 and negatively impact functionality for end users.”). This comports with the understanding of a person of ordinary skill in the art that cyber security techniques were important and routinely used to secure virtual assets in cloud computing environments. *Supra*, §VIII.D.1.

85. Veselov also describes applying its security techniques to typical computing resources of cloud environments. 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); *see also id.*, 20:46-56 (describing 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”). As I previously explained, a person of ordinary skill in the art

would have understood that cloud computing systems were commonly hosted on these types of resources—e.g., virtual machines 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.”

86. Accordingly, Veselov teaches a method for “securing virtual cloud assets in a cloud computing environment against cyber threats,” as recited in the preamble of claim 1.

b. Element 1.1

87. Element 1.1 recites “determining a location of 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 this element. I first explain the various ways Veselov teaches scanning a “snapshot of at least one virtual disk of a protected virtual cloud asset.” Next, I explain how Veselov’s scanning of snapshots involves “determining a location” of such snapshots. Lastly, I explain why the protected virtual cloud asset can be “instantiated in the cloud computing environment.”

88. Veselov teaches performing a security assessment on “a snapshot of at least one virtual disk of a protected virtual cloud asset,” as recited. Rather than simply scanning a given resource (e.g., the current instance of a virtual machine or

associated virtual disks), Veselov describes scanning a “snapshot” of the resource. For example, Veselov states that “[i]n various embodiments, 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.” EX1007 (Veselov), 3:20-27 (emphasis added); *see also id.*, 4:8-14 (“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.”), 4:53-54 (“In some embodiments, the scanning service 110 may obtain and analyze snapshot data 146.”), 6:25-27 (“The scanning service 110 enables the customers and other services of the computing resource service provider 104 to manage and operate analysis of various snapshot data 146 generated based at least in part on computing resources of the computing resource service provider 104, such as one or more virtual machines 112.”). Performing the security scans on snapshots, rather than directly on the protected resources, enhances the ability of the systems and methods in Veselov to “overcome drawbacks of existing security assessment systems, such as allocation and overuse of processing power.” *Id.*, 2:42-46; *see also id.*, 2:56-63 (“Performing a robust security assessment on a target computing device, such as a server, may reduce the target computing device’s capabilities.”).

89. Veselov also specifically teaches performing its assessments on a snapshot “of at least one virtual disk of a protected virtual cloud asset.” Discussing the embodiments of Figures 3A-B, 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 person of ordinary skill in the art 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.”), 8:52-60 (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 virtual machine may be stored on logical volume, which “may store data implementing and / or used by the virtual machine”), 11:34-41 (virtual machine 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”), 18:16-21 (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.

90. Veselov also teaches scanning snapshots of “virtual disks” with its discussion of scanning virtual storage elements of virtual machines. 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 person of ordinary skill

in the art 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 person of ordinary skill in the art 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.*

91. Moreover, for each of these disclosures regarding scanning a snapshot of a virtual disk, it would have been clear to a person of ordinary skill in the art that Veselov is describing, or at least suggesting, a virtual disk of “a protected virtual cloud asset” such as a virtual machine. Veselov notes that the snapshot may be of a logical volume—i.e., a virtual disk, as I explained above—implementing a “target computing resource, such as a virtual machine or an instance of a virtual machine.” *Id.*, Abstract; *see also id.*, 2:42-54 (“A security assessment may involve sending

requests and/or other information to one or more targets, such as a virtual machine....”), 3:29-40 (“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.”). Veselov further explains that these may be virtual machines in a data center, which, as I previously explained, were routinely used for cloud systems. *Id.*, 2:22-35 (security assessment performed in data center at any level of abstraction, including “virtual machines and virtual environments implementing virtual machines”); *see also supra*, §VIII.C (discussing relevant state of the art regarding cloud computing). I also previously explained why Veselov teaches performing its security assessments on virtual cloud assets such as virtual machines in a cloud computing environment (*supra*, §XII.A.1.a), and a person of ordinary skill in the art would have viewed such an asset to be a “protected” virtual cloud asset because it is being protected by the security assessment.

92. Thus, Veselov teaches that the security assessment is performed on a “snapshot of at least one virtual disk of a protected virtual cloud asset,” as recited.

93. Veselov further teaches “determining a location” of such snapshots. As I explained above in §X.A, a person of ordinary skill in the art would have understood this limitation as encompassing the determination of both a virtual

location and a non-virtual storage location. As I explain below, Veselov teaches both.

94. Veselov explains that, prior to or as part of the security assessment, “the scanning system obtains, or obtains access to” the snapshot. EX1007 (Veselov), 3:20-27; *see also id.*, 4:53-54 (“In some embodiments, the scanning service 110 may obtain and analyze snapshot data 146.”), 6:40-43 (“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.”); *see also id.*, 15:18-34, 16:21-26, 18:16-21, Figs. 2, 4, 6 (depicting obtaining steps). As Veselov explains, obtaining the snapshot can include bringing a copy of the snapshot into a virtual testing environment from various sources, whereas obtaining *access* to the snapshot can include accessing the snapshot in the virtual resource’s own computing environment. *See id.*, 9:14-25; *see also id.*, 3:41-51 (“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.”), Figs. 3A (depicting exemplary virtual computing environment 304A, where the target virtual machine resides, and

a separate test environment 305, where the snapshot of the virtual machine can be analyzed), 3B (depicting virtual machine and analyzed snapshot in same virtual computing environment 304B). Both of these approaches involve, or at least suggest, determining a “location” of the snapshot. For example, regardless of whether the scanning system is obtaining the snapshot or obtaining *access* to the snapshot, Veselov teaches determining a general location of the snapshot. *See supra*, §VIII.A (discussing commonly referenced types of locations). This can be seen, for example, in Veselov’s explanation that the scanning service may “determine the corresponding virtualization layer 144 for the virtual machine 112” before transmitting a command to provide snapshot data (or provide access to the snapshot data), the virtualization layer in which the snapshot resides being one type of general location of the snapshot. *Id.*, 6:25-49. Though mentioned in the context of Figure 1, a person of ordinary skill in the art would have understood that this discussion applies to the techniques discussed in later embodiments. *See, e.g., id.*, 4:19-29 (“Referring to FIG. 1, embodiments of the present disclosure may operate within or upon a computing environment 100....”); *see also 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.”), 10:37-41 (Figs. 3A-B describe security assessments “in accordance with various embodiments”), 16:15-

18 (Fig. 4 illustrates security-assessment method “in a computing environment like the example computing environments 300, 350 of FIGS. 3A-B), 16:56-67 (Figs. 5A-B illustrate “additional embodiments of the present security assessment systems,” and depicted components “as well as their implementations and interactions, are the same as described above” for associated components of Figs. 3A-B), 18:16-18 (“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.”), 18:56-59 (“FIG. 7 illustrates aspects of an example environment 700 for implementing aspects in accordance with various embodiments.”), 24:54-25:2 (noting that phrases such as “in an embodiment” is “not necessarily intended to refer to the same embodiment or to any one particular embodiment described.”).

95. With respect to obtaining “access” to the snapshot, Veselov states that the scanning service “may obtain access to the snapshot data, such **as at a storage location** of the snapshot data.” *Id.*, 9:17-25 (emphasis added). A person of ordinary skill would have understood that this teaching, presented in the context of process 200 (Fig. 2), applies to Veselov’s other embodiments involving obtaining access to the snapshot data, since Veselov states that “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” (*id.*, 8:60-62), and as I previously explained, Veselov

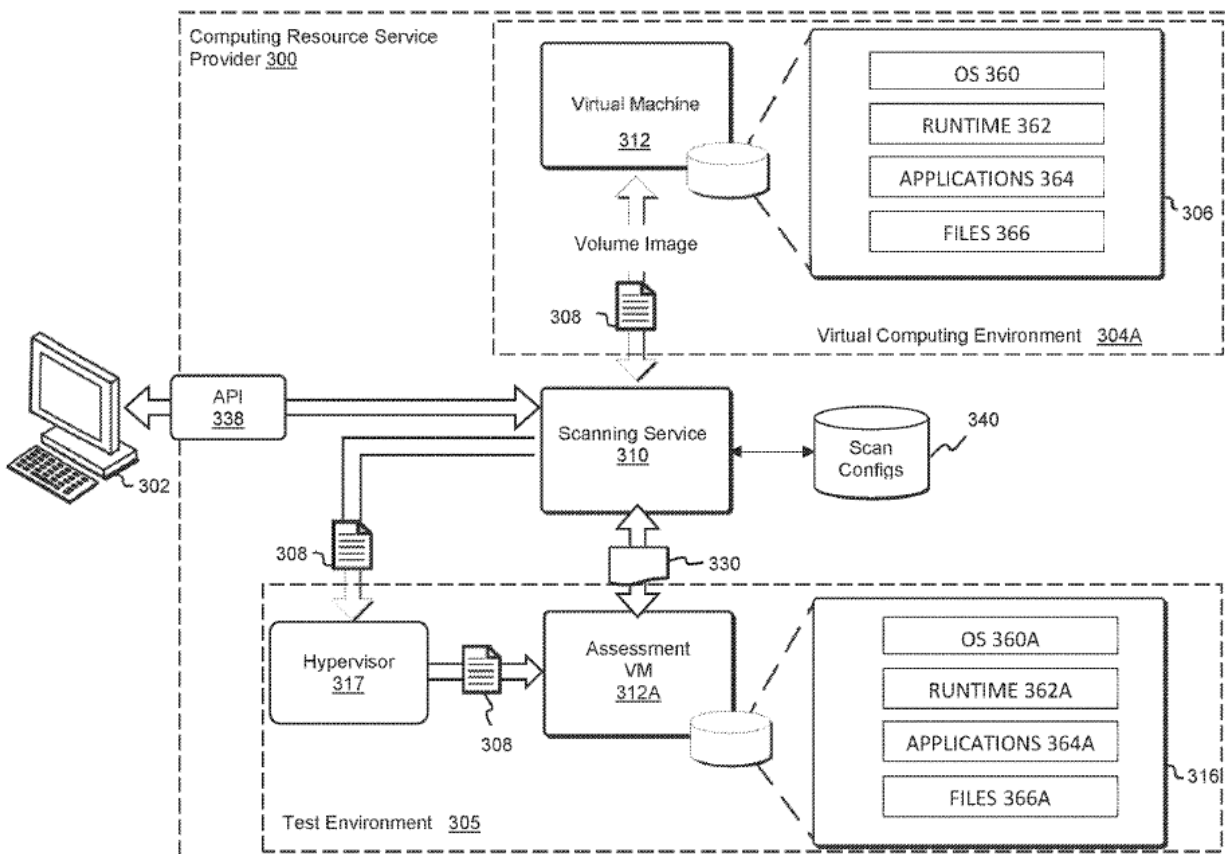
states that Figure 1 applies to the other embodiments. Veselov’s broad disclosure of the snapshot’s “storage location” at least encompasses the general location of the snapshot where the scanning service accesses the snapshot data, including the common types of general locations that I discussed above in paragraphs 30 (which apply to computer data generally and snapshot data more specifically, as I noted in paragraph 34). A person of ordinary skill in the art would have understood that Veselov’s broad disclosure of the snapshot’s storage “location” would also encompass accessing the snapshot data at a virtual location such as a virtual address—particularly since this was a well-known, commonly used way of accessing virtual assets (*supra*, §VIII.B)—or at a specific non-virtual storage location. *See also* EX1007 (Veselov), 10:63-11:4 (“[T]he virtual machine 312 may be associated with a virtual network address...”), 15:13-23 (request to perform security scan may include “identification of the target virtual machine”), Figs. 3A (depicting snapshot in original virtual computing environment 304A and virtual test environment 305), 3B (depicting snapshot accessed in original virtual computing environment 304B), 5A (depicting snapshot in original computing environment 504A and test environment 505), 5B (depicting snapshot accessed in original virtual computing environment 504B). In the methods in which Veselov does not obtain the snapshot, but instead obtains access to the snapshot, Veselov’s scanning service

operates on the snapshot while it remains in its original virtual computing environment or in a third-party snapshot storage service. *See* EX1007 (Veselov), 12:64-13:52 (describing embodiments in which snapshot data remains within protected virtual computing environment), 17:30-43 (“[T]he scanning service 510 may communicate with a provisioning service of the virtual computing environment 504B, such as a hypervisor 518, in order to allocate a logical volume 516 within the virtual computing environment 504B for the security assessment.”), Figs. 3B, 5B (depicting embodiments in which snapshot data never leaves the protected virtual computing environment). A person of ordinary skill in the art would have understood that accessing the snapshot at a given location would involve, or at least suggest, “determining” that location, since determining the location of data was a typical precursor for accessing data at that location.

96. At a minimum, determining the location of a snapshot would have been obvious in view of Veselov’s teachings. As I previously explained, obtaining or accessing a computing resource, such as Veselov’s snapshot, routinely involved determining where a snapshot was located (both physical disk location and virtual address) and which snapshot corresponded to which virtual disks or virtual machines. *Supra*, §VIII.B; *infra*, §XII.A.1.c; *see also* EX1009 (Wolf), 257; EX1015 (VMWare vSphere), 56; EX1019 (Rawat), Abstract; EX1020 (Sancheti), Abstract,

21:42-22:58. For example, hypervisors routinely stored snapshots (e.g., as files in the virtualization layer’s storage), and programs routinely determined the location of such snapshots by, for example, querying a hypervisor’s API. EX1015 (VMware vSphere), 164-66. In some implementations, the location data was also stored in metadata accompanying the snapshot data. *Id.* Moreover, as I discuss below regarding claim 4 (*infra*, §XII.B.3), utilities such as cloud management consoles were commonly queried to determine both the location of snapshots and the locations of corresponding virtual disk(s). With a person of ordinary skill in the art having been familiar with commonplace techniques for obtaining and accessing a remote resource by determining its location, such person would have found “determining a location” of the snapshot, as recited, to be obvious in view of Veselov’s teachings regarding obtaining (or accessing) snapshots from the target VM’s original computing environment or a third-party storage service. *See, e.g., supra*, §§VIII.A-B.

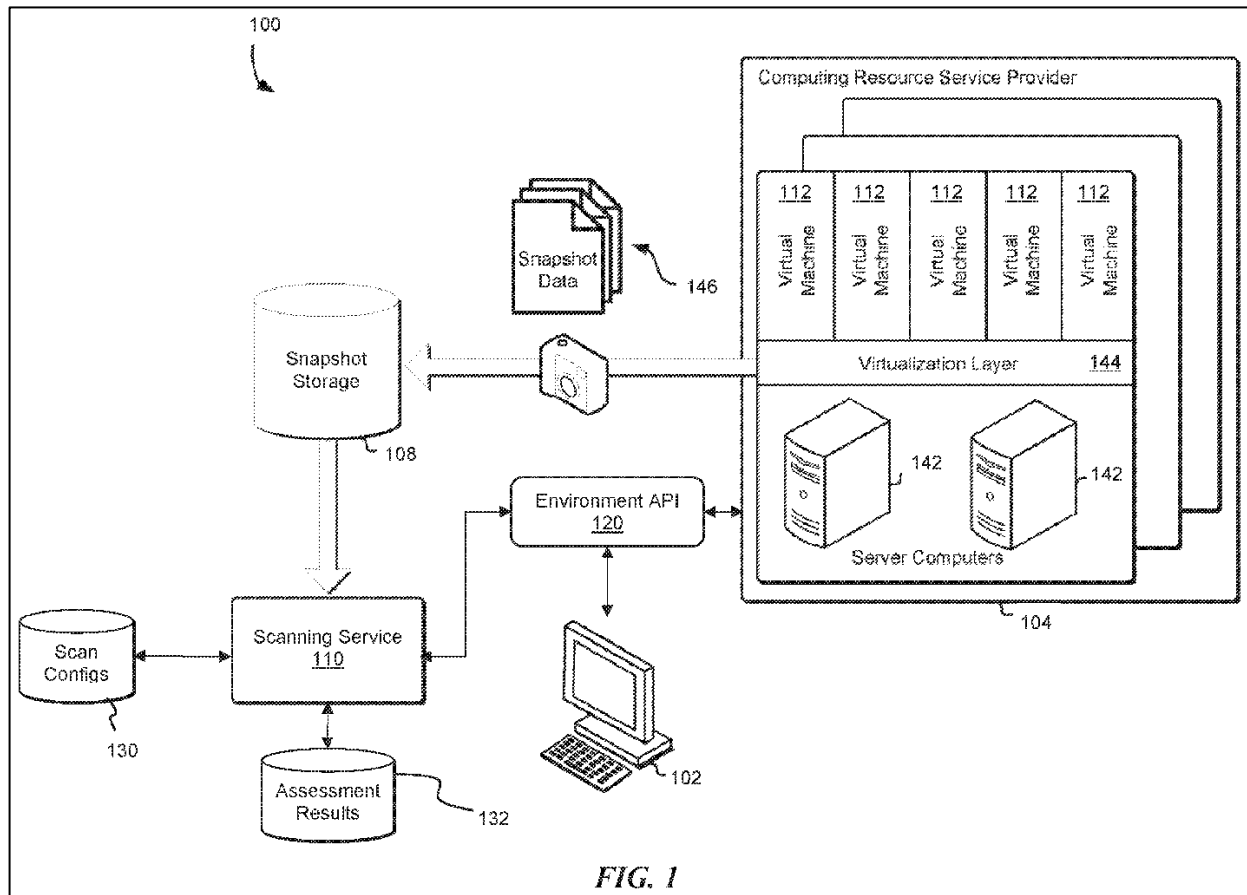
97. Obtaining the snapshot (i.e., bringing a copy of the snapshot into a virtual testing environment) would similarly encompass both determining a virtual address and, alternatively, determining a physical storage location. As I noted above, Veselov describes analyzing the snapshot within a virtual test environment. A virtualized test environment is depicted, for example, in Figure 3A:



EX1007 (Veselov), Fig. 3A (depicting test environment 305); *see also id.*, Fig. 5A (depicting analogous test environment 505). To obtain a copy of the snapshot data from the protected virtual computing environment and place the copy in the outside test environment, the scanning service interacts with the virtualization layer, or hypervisor, of the protected environment. As I discussed above, the scanning service “may determine the corresponding virtualization layer 144 for the virtual machine 112” identified in a request from the user 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,” at which point the

virtualization layer of the protected cloud environment may “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.” *Id.*, 6:25-49. Here again, a person of ordinary skill in the art would have understood that obtaining a copy of a snapshot from its original host environment would encompass determining a virtual location or a non-virtual location of that snapshot. At a minimum, a person of ordinary skill in the art would have found it obvious to determine the snapshot’s location in view of Veselov’s disclosure of API calls to a hypervisor to obtain snapshot data. *Supra*, §§VIII.A-B; *see also, e.g.*, EX1009 (Wolf), 221, 257; EX1015 (VMWare vSphere), 56, 124, 164-66); EX1031 (NIST IT Asset Mgmt), 1 (Executive Summary).

98. Second, Veselov discloses obtaining snapshot data from a third-party storage service outside of the protected virtual computing environment, for example, as shown in Figure 1:



EX1007 (Veslov), Fig. 1 (depicting snapshot storage 108); *see also id.*, 9:14-25 (“[T]he scanning service may obtain snapshot data representing a state of the target resource By accessing a data store (e.g., the snapshot storage service) containing the snapshot.”), 7:34-8:21 (describing example snapshot storage services). As an alternative to querying a hypervisor to determine the location of a snapshot located within a protected virtual computing environment, Veslov discloses querying a data store containing snapshots, which encompasses determining the location of both the data store and the snapshot within it. *Id.*, 7:34-8:21. For example:

the snapshot storage service 108 may include a database, such as a Structured Query Language (SQL) database, which enables the customer, the scanning service 110, or other services of the computing resource service provider 104 to query the snapshot storage service 108 for information contained in or associated with the snapshot data 146 and stored by the snapshot storage service 108. For example, **the snapshot data 146 may be stored with its own unique identifier as well as the identifier of the virtual machine instance from which the snapshot data 146 was derived; the snapshot storage service 108 may be queryable by either or both identifiers.**

Id., 7:43-55 (emphasis added); *see also id.*, 7:62-8:15 (“The snapshot storage service 108 may operate as a key value store that associates data objects with identifiers of the data objects that may be used by the customer to retrieve or perform other operations in connection with the data objects stored by the snapshot storage service 108.”), 19:21-37 (describing a data store in the context of a virtualized system). At a minimum, for the same reasons I discussed above, a person of ordinary skill in the art would have found it obvious to determine the snapshot’s virtual location (or non-virtual location) in view of Veselov’s disclosure of obtaining or accessing such data.

99. Third, Veselov discloses that after a copy of a snapshot has been brought into the test environment, the scanning service may interact with the snapshot via the hypervisor of the test environment. *See* EX1007 (Veselov), Figs. 3A (depicting test environment 305 with hypervisor 317), 5A (analogous

arrangement for test environment 505 with block device provisioning service 517), 12:21-63 (discussing test environment with hypervisor), 16:56-67 (Fig. 3A-B disclosures apply to Figs. 5A-B), 17:9-29 (discussing test environment with block device provisioning service). A person of ordinary skill in the art would have understood that determining the location of a snapshot would occur in at least at two different circumstances within Veselov's described use of a test VM within a test environment. First, Veselov describes the test environment hypervisor, at the direction of the scanning service, copying the snapshot's files into a test environment storage 316. EX1007 (Veselov), 12:21-63. A person of ordinary skill in the art would have understood the action of placing snapshot data into a storage location to involve determining the location in which the data would be placed. Second, during the scan, Veselov discloses maintaining an index of snapshot data locations within the test environment to enable interaction between the test VM and the snapshot data. *Id.* Veselov explains that within the test environment:

the hypervisor 317 may create and/or maintain an **index associating data locations of the second logical volume 316 with the actual locations of the data within the physical storage device**. The hypervisor 317 may receive a read request from the virtual machine instance, the read request indicating that the instance wants to read data from one of its virtual hard disks, the second logical volume 316; to respond to the read request, the hypervisor 317 uses the index to

identify which data is being read by the instance, and **where on the physical storage device the data is stored**. The hypervisor 317 may retrieve the data from the physical storage device and deliver the data to the requesting virtual machine instance.

Id., 12:51:63 (emphasis added). At a minimum, a person of ordinary skill in the art would have understood that, in view of Veselov’s disclosure of querying a hypervisor that mapped virtual addresses of snapshots to physical locations of snapshot data on a storage device, determining the location of such a snapshot was obvious.

100. Thus, Veselov teaches that the security assessment encompasses “determining a location” of a snapshot as recited.

101. Lastly, Veselov teaches that “the virtual cloud asset is instantiated in the cloud computing environment” as recited in the wherein clause of element 1.1. *See* EX1007 (Veselov), Abstract (systems for “performing a security assessment of a target computing resource, such as a virtual machine or an **instance** of a virtual machine” (emphasis added)), 6:9-20 (“These virtual computer components, and other physical and virtual resources, may be discretized into **instances** of one or more virtual machines 112. The virtual machine 112 or components thereof may be provided to the customers, end users, and/or other services inside or outside of the service provider 104. For example, a server computer 142 may host a first virtual

machine 112 **instantiated** from a first volume image and operated by a first customer and may host a second virtual machine 112 **instantiated** from a second volume image that is operated by a second customer.” (emphasis added)). As I discussed above with respect to the preamble of claim 1, Veselov teaches that the virtual cloud asset can be a virtual disk/machine in a cloud computing environment. *Supra*, §XII.A.1.a. Given Veselov’s focus on analyzing a *snapshot* of this resource, a person of ordinary skill in the art would have expected the target resource itself to still be instantiated in its cloud computing environment in many instances. Indeed, Veselov notes benefits of its snapshot-based analysis, including avoiding making the target resource unavailable during the assessment, which at least suggests that the target resource may be instantiated during the assessment. EX1007 (Veselov), 2:42-46 (“The present systems and methods overcome drawbacks of existing security assessment systems, such as allocation and overuse of processing power, unavailability of resources services that are part of the assessment, and complexity of installation and monitoring of the assessment system.”).

102. Accordingly, Veselov teaches “determining a location of 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 recited by element 1.1.

c. Element 1.2

103. Element 1.2 recites “accessing the snapshot of the virtual disk based on the determined location.” As I explain below, Veselov teaches this element.

104. In the previous section, I discussed Veselov’s teachings regarding performing a security assessment based on a snapshot of a virtual disk of a protected virtual cloud asset. *Supra*, §XII.A.1.b. I also explained how Veselov teaches or suggests multiple methods of determining a location (virtual and non-virtual) of a snapshot, including a location within the original computing environment, a location within a third-party snapshot storage service, and a location within a local testing environment. *Supra*, §XII.A.1.b. Below, I explain why, for each such “location,” Veselov teaches accessing the snapshot based on the determined location.

105. First, Veselov discloses both accessing (without copying) and obtaining (a copy of) a snapshot based on its location within the protected virtual computing environment. For example, in discussing Figure 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.”), 5:18-22 (“[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.”), 6:43-49 (“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”).

106. At a minimum, accessing a snapshot based on its location in the protected virtual computing environment would have been obvious. As I discussed

above, cloud computing environments commonly included utilities with GUIs and corresponding consoles and APIs to manage cloud resources such as virtual machines. *Supra*, §VIII.C. These utilities not only routinely provided users with information about a snapshot (such as its location and its corresponding virtual machines and virtual disks) but also facilitated gaining access to the snapshot to, for example, make copies of the snapshot. *Supra*, §§VIII.B, VIII.C. Veselov discloses use of such an API to not only “determine the corresponding virtualization layer 144 for the virtual machine 112,” but also “**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.” EX1007 (Veselov), 6:25-49 (emphasis added). A person of ordinary skill in the art, in view of Veselov’s disclosure of making an API call to ascertain the correct hypervisor and snapshot data for a particular virtual machine, would have found it obvious to use commonplace techniques for locating snapshots and accessing them based on their locations. *See, e.g., supra*, §VIII.A-B.

107. Second, Veselov discloses both accessing (without copying) and obtaining a snapshot based on its location within a third-party snapshot storage service. Veselov discusses how its scanning service may obtain the snapshot data “**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), 9:14-25 (emphasis added); *see also id.*, 5:9-17 (“data storage service (e.g., snapshot storage service 108) accessible by the scanning service 110”), 5:18-22 (“[T]he computing resource provider ... may store the snapshot data 146”), 6:43-49 (“The virtualization layer 144 may be configured to ... store the snapshot data 146 via a snapshot storage service 108.”), 8:16-21 (scanning service “may obtain the snapshot data 146 from the snapshot storage service 146”). Veslov discloses that the snapshot storage service may be “queryable,” that is, where “the data storage system and/or data storage device enable[s] requestors to submit data queries and receive response[s] to the submitted data queries.” *Id.*, 7:34-61. Veslov discloses that a queryable snapshot storage service may be “configured to store snapshot data 146 and **make the snapshot data 146 accessible** to one or more other computer systems, such as the scanning service 110.” *Id.*, 7:34-37; *see also id.*, 7:76:64 (“The snapshot storage service 108 may be a collection of computing resources configured to synchronously process requests to store and/or access data.”), 8:16-34 (discussing interaction between scanning service and snapshot storage service). At a minimum, accessing a snapshot based on its location in a snapshot storage service would have been

obvious, since Veselov describes querying a snapshot storage service to access a snapshot of a virtual disk, and accessing remotely stored data based on its determined location was a straightforward, common way of accessing such data. *See, e.g., supra*, §VIII.A-B.

108. Third, Veselov discloses accessing a snapshot based on its location within a local test environment after obtaining a copy of the snapshot. As I previously discussed, Veselov indicates that obtaining the snapshot involves bringing a copy of the snapshot into the local testing environment, whereas obtaining *access* to a snapshot involves getting access prior to analyzing the snapshot in the target resource’s own computing environment or in a third-party storage service. Veselov discloses that within the test environment, the hypervisor may maintain an “index associating data locations of the second logical volume 316 with the actual locations of the data within the physical storage device.” *Id.*, 12:37:63 (emphasis added); *see also id.*, Fig. 3A (depicting the test environment hypervisor interacting with the scanning service and volume image). When discussing a similar test environment with a block device provisioning service, Veselov states:

[t]he 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.

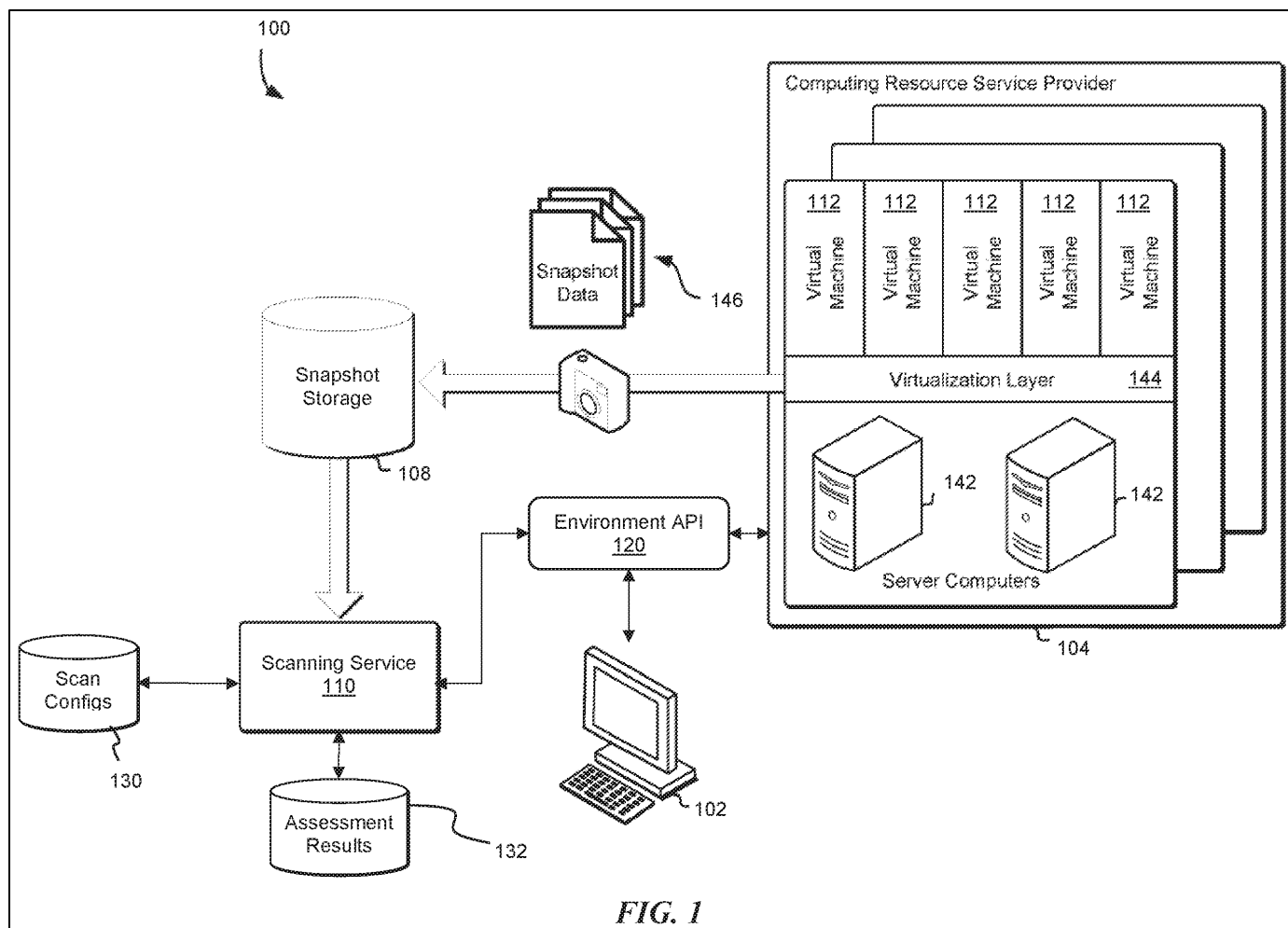
Id., 17:9-29 (emphasis added). At a minimum, a person of ordinary skill in the art would have found it obvious to access the snapshot based on its determined location (both virtual and non-virtual), since Veselov describes accessing snapshot data in a virtualized test environment while also noting that hypervisors inside such test environments index data's location (i.e., virtual location) with the actual location of the data in physical storage.

109. Accordingly, Veselov teaches “accessing the snapshot of the virtual disk based on the determined location,” as recited by element 1.2.

d. Element 1.3

110. Element 1.3 recites “analyzing the snapshot of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset.” As I explain below, Veselov teaches this element.

111. Veselov teaches “analyzing the snapshot of the protected virtual cloud asset” via its scanning service. The scanning service is shown, for example, as scanning service 110 in Figure 1 below:



EX1007 (Veslov), Fig. 1; *see also id.*, Figs. 3A-B (depicting scanning service 310), 5A-B (depicting scanning service 510). Veslov explains that the scanning service “may obtain and **analyze snapshot data** 146.” *Id.*, 4:49-54 (emphasis added); *see also id.*, 8:16-51 (describing variety of analysis techniques), Figs. 2 (depicting “Perform Assessment” step), 4 (same), 6 (same), 10:1-16 (describing types of analyses).

112. Veselov teaches several different ways that this analysis can be performed. For example, discussing the example process of Figure 2, Veselov explains:

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.

113. In one approach, the scanning service may use the snapshot to launch an active instance of a targeted virtual machine. 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 VM), Figs. 3A-B (depicting assessment VM).

114. The scanning service may also analyze the snapshot data without instantiating a live instance of the targeted virtual machine. 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.*, 17:10-29 (“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). Veselov’s methods of analyzing a snapshot as a data file are “agentless,” that is, they do not require a scanning application running inside an assessment VM. In these methods, the snapshot is not spun up into an instantiated assessment VM, and thus no security scanning applications could be installed. *See infra*, §§XII.B.4.a, XII.B.4.b (elements 5.1 and 5.2).

115. Veselov also teaches that this analysis is done “to detect potential cyber threats risking the protected virtual cloud asset.” As I discussed above in §§VIII.D.2, VIII.D.3, cyber security scanning tools were routinely used to generate information for users, administrators, or engineers to better protect against cyber threats, and Veselov is no different. Veselov also states that the scanning service may be used to “provide security risk information to customers and/or other services of the computing resource service provider 104.” EX1007 (Veselov), 4:57-59. Veselov accomplishes this with the scanning service that is configured to “perform security assessments and produce assessment results based at least in part on snapshot data.” *Id.*, 5:9-11. These assessment results, and the security risk information they provide, can identify a variety of threats risking the protected asset. *See, e.g., id.*, 1:46-49 (“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.”), 2:35-41 (Veselov’s system “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”), 4:1-4 (“[T]he 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.”).

116. Accordingly, Veselov teaches “analyzing the snapshot of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset” as recited in element 1.3.

e. Element 1.4

117. Element 1.4 recites “alerting detected potential cyber threats based on a determined priority.” As I explain below, the combination of Veselov and Basavapatna teaches this element.

118. First, Veselov teaches “alerting detected potential cyber threats.” In the previous section, I explained how Veselov teaches analyzing the snapshot to “detect potential cyber threats.” Veselov further teaches “alerting” these detected potential cyber threats by having the scanning service generate and communicate “assessment

results.” EX1007 (Veselov), 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 (“[T]he scanning service 110 may obtain the snapshot data 146 from the snapshot storage service 146 and may process the snapshot data 146 to generate one or more assessment results.”).

119. Veselov explains that the assessment results identify the vulnerabilities detected by the scanning system. *See id.*, 20:57-21:13 (assessment results “describ[e] a security vulnerability of the first virtual machine instance”), 23:19-47 (assessment results “describ[e] one or more security vulnerabilities of the first computing resource”), claims 1, 16-17 (same); *see also id.*, 10:17-24 (“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**; examples depending on the method of generating the scannable volume and the rules of the assessment are described below.” (emphasis added)), 10:24-36 (at step 212, the scanning system identifies remedial actions “if vulnerabilities are identified in the assessment results”).

120. Veselov then describes “alerting” these vulnerabilities by communicating the assessment results (which identify the vulnerabilities) to a user,

owner, or administrator of the targeted resource. For example, Veselov describes “providing the assessment results in a user-readable format to a user interface to be displayed on the user device.” *Id.*, 10:30-36. An example of this alerting technique is shown in Figure 6 at step 616:

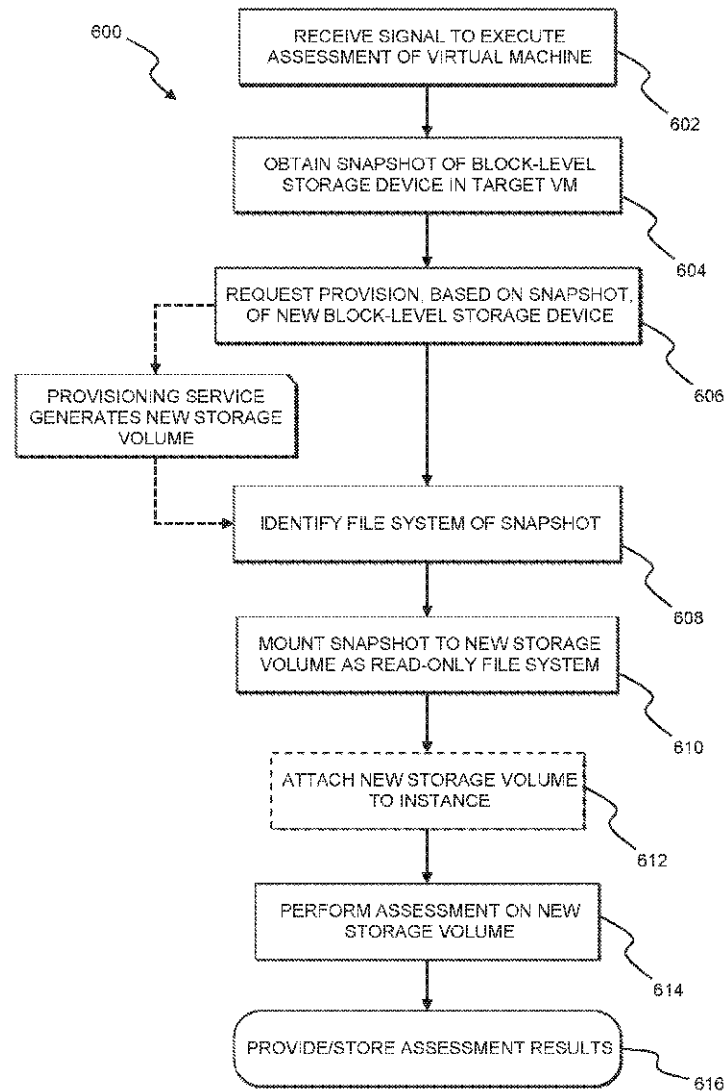


FIG. 6

Id., Fig. 6; *see also id.*, Figs. 2 (results provided in step 212), 4 (results provided at step 416). Veselov states: “At step 616, the scanning service may perform actions

on the assessment results, such as storing or delivering the results, as described above.” *Id.*, 18:53-56. A person of ordinary skill in the art would have understood delivering the results to be a way of “alerting” detected potential cyber threats. Thus, Veselov teaches “alerting detected potential cyber threats” as recited.

121. While Veselov does not expressly state that the alerting is “based on a determined priority,” this type of alerting was commonplace well before 2019. Basavapatna describes an example of the common technique of prioritizing potential cyber threats as part of an analogous security assessment. Basavapatna describes multiple algorithms for determining priorities, based on weighting factors including the “risk associated with the vulnerability relative to other vulnerabilities” and the “criticality of the particular asset within the particular system.” EX1008 (Basavapatna), ¶¶14-15; *see also infra* (element 2.1). Basavapatna describes scoring each vulnerability such that “if an aggregate risk metric for any vulnerability or threat rises above a specified threshold, the user should be alerted.” EX1008 (Basavapatna), ¶140. For example, after establishing a risk metric, the system “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.” *Id.*, ¶138. The risks are then displayed to the user in sorted order. *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.”). Basavapatna also explains the well-known importance of such 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. Accordingly, Basavapatna teaches “alerting detected potential cyber threats based on a determined priority” as recited. I explain the reasons for combining the teachings Veselov and Basavapatna to arrive at element 1.4 in more detail below in §XII.A.4.

122. Accordingly, the combination of Veselov and Basavapatna teaches “alerting detected potential cyber threats based on a determined priority” as recited by element 1.4. It is therefore my opinion that claim 1 was obvious over Veselov and Basavapatna.

2. Independent Claim 10

123. In my opinion, claim 10 was obvious for substantially the same reasons as claim 1. Independent claim 10 is substantially similar to independent claim 1, except that claim 10 recites “a non-transitory computer readable medium with instructions to cause a processing circuitry to execute a process” instead of a method. Claim 10 recites the following:

A non-transitory computer readable medium having stored thereon instructions for causing a processing circuitry to execute a

process for securing virtual cloud assets in a cloud computing environment against cyber threats, the process comprising:

determining a location of 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;

accessing the snapshot of the virtual disk based on the determined location; analyzing the snapshot of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset; and

alerting detected potential cyber threats based on a determined priority.

a. Claim 10 Preamble

124. The preamble of claim 10 recites a “non-transitory computer readable medium having stored thereon instructions for causing a processing circuitry to execute a process for securing virtual cloud assets in a cloud computing environment against cyber threats, the process comprising:” As I explain below, Veselov teaches this preamble.

125. The preamble of claim 10 is substantially similar to the preamble of claim 1 except that the preamble of claim 10 recites a “non-transitory computer readable medium having stored thereon instructions for causing a processing circuitry to execute a process...” with the subsequent description of that process mirroring the method recited in the preamble of claim 1. I previously explained why

Veselov teaches the preamble of claim 1, and that discussion applies here as well. *Supra*, §XII.A.1.a.

126. Veselov also teaches that instructions stored on a non-transitory computer readable medium cause processing circuitry to execute this process. EX1007 (Veselov), 20:57-60 (“Thus, in one aspect, the present disclosure provides a system, including a memory and a processor, the memory including computer-executable instructions....”); *see also id.*, 20:60-23:18 (describing scanning system’s actions caused by executing instructions), claim 1 (reciting a “system, including a memory and a processor, the memory including computer-executable instructions”). It would have been clear to a person of ordinary skill, given that person’s basic understanding of computer operations, that the system taught by Veselov and Basavapatna was configured to perform the relevant actions—i.e., the actions/functionality I discussed with respect to elements 1.1 to 1.4—by the processing circuitry’s execution of instructions contained in such a memory.

127. Accordingly, Veselov teaches the preamble of claim 10.

b. Element 10.1

128. Element 10.1 recites “determining a location of 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 explained above with respect to element 1.1, Veselov teaches this element. *Supra*, §XII.A.1.b.

c. Element 10.2

129. Element 10.2 recites “accessing the snapshot of the virtual disk based on the determined location.” As I explained above with respect to element 1.2, Veselov teaches this element. *Supra*, §XII.A.1.c.

d. Element 10.3

130. Element 10.3 recites “analyzing the snapshot of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset;” As I explained above with respect to element 1.3, Veselov teaches this element. *Supra*, §XII.A.1.d.

e. Element 10.4

131. Element 10.4 recites “alerting detected potential cyber threats based on a determined priority.” As I explained above with respect to element 1.4, the combination of Veselov and Basavapatna teaches this element. *Supra*, §XII.A.1.e.

3. Independent Claim 11

132. In my opinion, claim 11 was obvious for substantially the same reasons as claim 1. Independent claim 11 is substantially similar to independent claim 1, except that claim 11 recites a “system for securing virtual cloud assets in a cloud computing environment against cyber threats, comprising: ...” instead of a method. Claim 11 recites the following:

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

a processing circuitry; and

a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to:

determine a location of 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;

access the snapshot of the virtual disk based on the determined location;

analyze the snapshot of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset; and

alert detected potential cyber threats based on a determined priority.

a. Claim 11 Preamble

133. The preamble of claim 11 recites “a system for securing virtual cloud assets in a cloud computing environment against cyber threats, comprising:”

134. Veselov teaches this preamble. As I explained above with respect to the preamble of claim 1, Veselov teaches “securing virtual cloud assets in a cloud computing environment against cyber threats.” *Supra*, §XII.A.1.a. A person of ordinary skill would have understood Veselov to teach a system to perform all

aspects of its disclosure. *See, e.g.*, EX1007 (Veslov), Abstract (“Systems for performing a security assessment of a target computing resource....”).

135. Accordingly, Veslov teaches the preamble of claim 11.

b. Element 11.i

136. Element 11.i recites “a processing circuitry.” As I explain below, Veslov teaches this element.

137. Veslov explains that its scanning system can be implemented using one or more servers, which include a central processing unit (i.e., “processing circuitry”):

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. A server computer (e.g., server computers 142 implementing the virtual machine instances 112) may be any device or equipment configured to execute instructions for performing data computation, manipulation, or storage tasks, such as a computer or a server. A server computer may be equipped with any needed processing capability including **one or more processors**.

EX1007 (Veslov), 5:23-37; *see also id.*, claim 1 (reciting a “system, including a memory and a **processor**”); *supra*, §XII.A.2.a (“processing circuitry” is also described in the preamble of claim 10).

138. Accordingly, Veselov teaches “a processing circuitry” as recited by element 11.i.

c. Element 11.ii

139. Element 11.ii recites “a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to:” As I explain below, Veselov teaches this element.

140. Veselov teaches “a memory” with instructions that, when executed, performs the actions of the described scanning system. EX1007 (Veselov), 20:57-60 (“Thus, in one aspect, the present disclosure provides a system, including a memory and a processor, the memory including computer-executable instructions....”); *see also id.*, 20:60-23:18 (describing scanning system’s actions caused by executing instructions), claim 1 (reciting a “system, including a memory and a processor, the memory including computer-executable instructions”). It would have been clear to a person of ordinary skill, given that person’s basic understanding of computer operations, that the system taught by Veselov and Basavapatna was configured to perform the relevant actions—i.e., the actions/functionality I discussed with respect to elements 1.1 to 1.4—by the processing circuitry’s execution of instructions contained in such a memory.

141. Accordingly Veselov teaches “a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to:” as recited by element 11.ii.

d. Element 11.1

142. Element 11.1 recites “determine a location of 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 explained above with respect to element 1.1, Veselov teaches this element. *Supra*, §XII.A.1.b.

e. Element 11.2

143. Element 11.2 recites “access the snapshot of the virtual disk based on the determined location.” As I explained above with respect to element 1.2, Veselov teaches this element. *Supra*, §XII.A.1.c.

f. Element 11.3

144. Element 11.3 recites “analyze the snapshot of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset;” As I explained above with respect to element 1.3, Veselov teaches this element. *Supra*, §XII.A.1.d.

g. Element 11.4

145. Element 11.4 recites “alert detected potential cyber threats based on a determined priority.” As I explained above with respect to element 1.4, the combination of Veselov and Basavapatna teaches this element. *Supra*, §XII.A.1.e.

4. Reasons Why a Person of Ordinary Skill in the Art Would Have Combined Veselov and Basavapatna

146. A person of ordinary skill had numerous reasons to combine the teachings of Veselov and Basavapatna, as I discussed above in §XII.A.1.e, to arrive at the claimed invention.

147. As an initial matter, Veselov and Basavapatna are analogous art to one another and to the '735 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 and reporting threats. *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 (methods involve “identifying a standardized vulnerability score for a particular vulnerability in a plurality of known vulnerabilities, the standardized vulnerability score indicating a relative level of risk associated with the particular vulnerability relative other vulnerabilities”). The references are also in the same field of endeavor as the ’735 patent and are reasonably pertinent to the problem addressed in the ’735 patent. *See, e.g.*, EX1001, 1:10-12 (“This disclosure relates generally to cyber-security systems and, more specifically, to techniques for securing virtual machines.”).

148. Moreover, arriving at the claimed invention would have involved nothing more than the predictable use of known hardware and software elements according to their known functions. For example, securing cloud systems by accessing and analyzing snapshots of virtual disks to identify threats, as taught by Veselov, was a predictable, common technique. *Supra*, §VIII.D.2; EX1014 (Klemperer), 24-25. Alerting identified cyber threats based on a determined priority, as taught by Basavapatna, was similarly predictable and commonplace. *Supra*,

§VIII.D.3; EX1044 (Giakouminakis), 6:9-18, 11:51-67. Combining Basavapatna’s teachings with those of Veselov to arrive at the claims likewise would have been nothing more than the use of a known technique—i.e., alerting cyber threats based on a determined priority—that was commonly used to improve similar cyber security systems in the same way. As I explained in §VIII.D.3, it was well known that prioritized alerts such as those taught in Basavapatna were commonly used to facilitate remediation of more critical threats as soon as possible. A person of ordinary skill would have recognized that such techniques were equally useful in the similar context of security assessments based on snapshots of virtual disks such as those taught by Veselov. Indeed, virtual machines operate in a virtualization layer but largely contain the same types of files, configurations, and functionality as physical machines. *Supra*, §VIII.A. A person of ordinary skill in the art thus would have recognized that virtual resources (and snapshots of such resources) have many of the same threats and vulnerabilities as non-virtual resources (*supra*, §VIII.D.1), meaning alert prioritization was similarly implemented—and similarly useful—in Veselov’s context.

149. Multiple considerations also would have motivated a person of ordinary skill in the art to combine Veselov and Basavapatna to arrive at the claims.

150. First, a person of ordinary skill in the art would have recognized that Basavapatna's teachings regarding alerting based on a determined priority were directly applicable to Veselov's general teachings regarding providing assessment results to a user. As I explained in §XI.A, Veselov broadly teaches alerting users about the results of its threat assessment but does not provide many specifics as to content or structure of the assessment results. Basavapatna then provides a more detailed discussion of how identified threats are processed, including the prioritization of certain threats and the corresponding alerts. *See supra*, §XI.B. A person of ordinary skill in the art would have recognized that Basavapatna's more detailed teachings regarding alert prioritization were a natural fit with Veselov's more high-level description of providing assessment results to users. Basavapatna simply provides a more detailed discussion than Veselov with respect to the alerting aspect of the security assessment, including a more robust description of the common, well-understood technique of prioritizing certain threats and alerts. *See supra*, §VIII.D.3; *see also* EX1046 (Crabtree), 11:52-12:14 (need for dynamic threat ranking "obvious and immediate"). That technique was equally appropriate and useful in the types of security assessments described in Veselov.

151. Second, a person of ordinary skill in the art also had reason to combine the teachings of Veselov and Basavapatna as discussed herein in order to obtain

known benefits associated with prioritized alerts, including benefits expressly discussed in Basavapatna itself. For example, as I previously explained, it was well known at the time that a threat analysis would typically identify numerous threats, some of which were more important to address (and address quickly) than others. *Supra*, §VIII.D.3. It was also well known that prioritizing certain threats and associated alerts—e.g., ranking, categorizing, or scoring threats and alerts—was helpful to facilitate more immediate mitigation of more pressing threats. *Id.* Basavapatna describes this well-known benefit, explaining 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.” EX1008 (Basavapatna), ¶138.

152. A person of ordinary skill in the art would have understood that this commonplace technique was beneficial in the type of security assessments taught by Veselov, including security assessments of snapshots associated with virtual machines in a cloud computing context. Snapshot-based analysis of virtual cloud assets, like other types of cyber-threat analysis, would have been expected to identify multiple threats of varying importance. *Supra*, §§VIII.D.2-3; *see also, e.g.*, EX1025 (NIST Incident Handling), 32; EX1044 (Giakouminakis), 6:9-18; EX1046 (Crabtree), 11:52-12:14. Likewise, it would have been clear that prioritizing certain

alerts based on threats to virtual cloud systems was just as important (if not more important) for virtual cloud systems as for other types of computer systems. *Supra*, §VIII.D.1; EX1025 (NIST Incident Handling), 3 (“Incidents should be prioritized based on the relevant factors, such as the functional impact of the incident (e.g., current and likely future negative impact to business functions), the information impact of the incident (e.g., effect on the confidentiality, integrity, and availability of the organization’s information), and the recoverability from the incident (e.g., the time and types of resources that must be spent on recovering from the incident).”); EX1045 (Canzanese), ¶7. A person of ordinary skill therefore would have expected Basavapatna’s teachings regarding prioritized alerts to beneficially “allow[] a user to quickly identify which assets are most at risk, or which vulnerabilities and threats are most dangerous for a system” in Veselov’s context. EX1008 (Basavapatna), ¶138.

153. Lastly, a person of ordinary skill in the art would have had a reasonable expectation of success in combining Veselov and Basavapatna to arrive at the claimed invention. Based on my experience in this field, providing a computer system that secures virtual cloud assets by determining the location of a virtual disk’s snapshot, accessing the snapshot based on that location, analyzing the snapshot to detect threats, and providing alerts—as taught by Veselov—would have been routine

for a person of ordinary skill. Further providing a threat-response process that alerts threats based on a determined priority, as taught by Basavapatna, also would have been a routine task for a person of ordinary skill. *See, e.g.*, EX1042 (Rubin), 2:10-27. Implementing each aspect of this technique, and the technique as a whole, would have been well within the ability of a person of ordinary skill. *Supra*, §VII (discussing level of ordinary skill). As I explained in §VIII.D.3, prioritizing certain threats/vulnerabilities and associated alerts was common practice at the time. For example, the implementation of alert prioritization—including such implementation in the context of virtual machines in a cloud computing environment such as those taught by Veselov—was well understood by those of ordinary skill in the art. Accordingly, given the well-understood nature of virtual machines, cloud computing systems, threat analysis, and alert prioritization, the combination of Veselov’s and Basavapatna’s teachings to arrive at the claimed invention would not have presented any significant technical challenges or led to any unexpected results.

B. Dependent Claims

1. Dependent Claims 2 and 12

154. Dependent claims 2 and 12 depend from independent claims 1 and 11, respectively, but recite substantially similar limitations. The preamble of claim 2 recites the “method of claim 1, further comprising...” and the preamble of claim 12 recites the “system of claim 11, wherein the system is further configured to...” Both

claims then recite analogous limitations. Given these similarities, I discuss the analogous limitations of claims 2 and 12 together in the following sections.

a. Elements 2.1 and 12.1

155. Element 2.1 recites “prioritizing each of the detected potential cyber threats based on their respective risk to the protected virtual cloud asset.” In element 12.1, the system is similarly configured to “prioritize each detected of potential cyber threats based on their respective risk to the protected virtual cloud asset.” As I explain below, the combination of Veselov and Basavapatna teaches these elements.

156. As I explained above with respect to element 1.4, the combination of Veselov and Basavapatna teaches alerting detected potential cyber threats based on a determined priority. In particular, Veselov generally teaches alerting detected potential cyber threats, and Basavapatna specifically teaches prioritized alerts. In addition to what I previously discussed, Basavapatna teaches “prioritizing each of the detected potential cyber threats based on their respective risk to the” protected asset. In combination with Veselov—which, as I discussed above with respect to elements 1.1, 1.3, and 1.4, teaches detecting and alerting threats to protected “virtual cloud” assets—Basavapatna’s discussion of prioritizing threats based on their respective risk to the asset teaches, or at a minimum suggests, the limitation as a whole.

157. As an initial matter, both Veselov and Basavapatna teach that alerts are provided for “each” of the detected potential cyber threats. Veselov discusses providing an “assessment result” that describes “one or more security vulnerabilities of the first computing resource; and associating the first assessment result with the first computing resource.” EX1007 (Veselov), 23:19-47. Basavapatna similarly discusses determining a risk metric “for each of the one or more assets for each of the one or more vulnerabilities.” EX1008 (Basavapatna), ¶14.

158. Basavapatna further teaches prioritizing each threat “based on their respective risk to the” protected asset. Basavapatna discloses aggregate risk metrics that are made up of multiple scores. One of these scores is a “standardized vulnerability score indicating a relative level of risk associated with the particular vulnerability.” *Id.*, ¶12. Another is a composite vulnerability score that 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”). Basavapatna combines this with a countermeasure component score that estimates the probability that “a countermeasure will mitigate [a] risk associated with the particular vulnerability.” *Id.*, ¶12. Basavapatna uses these scores to determine a risk metric for the particular

asset and particular vulnerability, with this risk metric being used to trigger certain actions, such as alerts. *Id.*, ¶140. (“[A] user can specify that if an aggregate risk metric for any vulnerability or threat rises above a specified threshold, the user should be alerted. Similarly, a user can specify that if an aggregate risk metric for any asset rises above a specified threshold, the user should be alerted.”); *see also id.*, ¶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.” (emphasis added)). Accordingly, Basavapatna teaches prioritizing each threat “based on their respective risk to the” protected asset.

159. 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.A.1.a (discussing Veselov’s teachings regarding protecting virtual cloud assets).

160. Accordingly, the combination of Veselov and Basavapatna teaches “prioritizing each of the detected potential cyber threats based on their respective risk to the protected virtual cloud asset” as recited by elements 2.1 and 12.1.

b. Elements 2.2 and 12.2

161. Element 2.2 recites “mitigating a potential cyber threat posing a risk to the protected virtual cloud asset.” In element 12.2, the system is similarly configured to “mitigate a potential cyber threat posing a risk to the protected virtual cloud asset.” As I explain below, Veselov and Basavapatna teach these elements.

162. Veselov teaches a variety of responses to the detection of cyber threats, including several ways of mitigating the threats to the protected asset. *See, e.g.*, EX1007 (Veselov), 10:17-36. For example, Veselov explains that the scanning service “may take various actions on the assessment results” including “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.” *Id.*, 10:24-36. Thus, in addition to simply notifying users or administrators of the threats, Veselov can mitigate the threats by identifying and/or taking remedial actions.

163. Basavapatna similarly teaches mitigating potential cyber threats by providing countermeasures to detected vulnerabilities. EX1008 (Basavapatna), Abstract (“A countermeasure component score is identified that indicates an

estimated probability that a countermeasure will mitigate risk associated with the particular vulnerability on the particular asset.”). Basavapatna describes both active countermeasures, such as installing software patches or making changes to asset configuration, and passive countermeasures, such as blocking network traffic from an attack source using a fire wall.

The host-based sensors 108 and the network-based sensors 110 can include one or more passive **countermeasures that are part of the sensor**. These passive countermeasures are software programs and/or hardware that protect assets from various threats. Each passive countermeasure reduces the risk that a threat will affect an asset. A passive countermeasure protects against a threat by detecting and stopping an attack associated with the threat, by detecting and stopping activities associated with the attack, or by **mitigating damage caused by an attack**. For example, a passive countermeasure may be configured to detect data having a signature associated with a particular attack, and block data with that signature. As another example, a passive countermeasure may generate back-up copies of particular files targeted by an attack, so that even if the attack attacks the files, the files can be restored. Example passive countermeasures include, but are not limited to, hardware firewalls, software firewalls, data loss prevention systems, web proxies, mail filters, host-based intrusion prevention systems, network-based intrusion prevention systems, rate-based intrusion prevention systems, content based

intrusion prevention systems, intrusion detection systems, and virus detection software.

EX1008 (Basavapatna), ¶24; *see also id.*, ¶¶20, 29. A person having ordinary skill in the art would have understood that these suggested countermeasures are used to mitigate a potential cyber threat. Moreover, as I explained in the previous section, the combination of Basavapatna’s teachings with Veselov would involve addressing threats “to the protected virtual asset.”

164. Accordingly, Veselov and Basavapatna independently teach “mitigating a potential cyber threat posing a risk to the protected virtual cloud asset,” as recited. While Veselov’s teachings are sufficient on their own to satisfy the limitation in element 2.2, a person having ordinary skill in the art also had reason to apply Basavapatna’s teachings in virtualized cloud systems as disclosed in Veselov. I previously explained reasons to combine Veselov’s teachings with Basavapatna’s teachings regarding alerting detected threats (*supra*, §XII.A.4), and that reasoning applies equally to mitigating actions following the detection of threats. The types of remedial and mitigating actions described in Veselov are comparable to those in Basavapatna. One of the primary purposes of detecting threats and vulnerabilities is to facilitate remediation and mitigation, and it would have been clear to a person of ordinary skill in the art to apply well-known and common automated or semi-

automated remedies such as installing patches and implementing firewalls. *Supra*, §VIII.D.3; *see also, e.g.*, EX1025 (NIST Incident Handling), 24.

165. Accordingly, Veselov and Basavapatna teach “mitigating a potential cyber threat posing a risk to the protected virtual cloud asset,” both individually and in combination, as recited by elements 2.2 and 12.2. It is therefore my opinion that claims 2 and 12 were obvious over Veselov and Basavapatna.

2. Dependent Claims 3 and 13

166. Claim 3 recites the “method of claim 1, wherein determining the location of the snapshot of at least one virtual disk further comprises: determining a virtual disk allocated to the protected virtual cloud asset.” Claim 13 similarly recites the “system of claim 11, wherein determining the location of the snapshot of at least one virtual disk further comprises: determining a virtual disk allocated to the protected virtual cloud asset.” Given the analogous nature of the additional limitations recited in these claims, I discuss them together in this section. As I explain below, Veselov teaches these additional limitations.

167. Veselov teaches “determining a virtual disk allocated to the protected virtual cloud asset.” As I explained above with respect to element 1.1 (*supra*, §XII.A.1.b), Veselov discloses that the protected virtual cloud asset may be a virtual machine that has one or more associated virtual disks allocated to it. *See, e.g.*, EX10X (Veselov), 3:29-40 (Veselov’s “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.” (emphasis added)); 6:7-10 (“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.”); 11:36-41 (“[T]he snapshot data of the virtual machine 312 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.”). A person of ordinary skill would have understood that obtaining a snapshot of a virtual machine that has an allocated virtual disk, as described in Veselov, would involve “determining” this virtual disk.

168. Determining this virtual disk would have been obvious at a minimum. For example, as I described earlier with respect to element 1.1 (*supra*, §XII.A.1.b), hypervisors commonly provided utilities with GUIs, as well as corresponding APIs, which maintained data that mapped snapshots to particular virtual disks and virtual machines. EX1015 (VMware vSphere), 164-66. Users commonly and easily discovered which areas of virtual disk storage, either volatile or pre-allocated to a particular virtual machine, corresponded to both each other and to a particular

snapshot, by using these utilities. *Id.* A person of ordinary skill in the art reading Veselov, which discloses security scans of snapshots of virtual machines with virtual disk storage allocated to the virtual machine, would have found it obvious to determine the virtual disk allocated to the protected virtual machine.

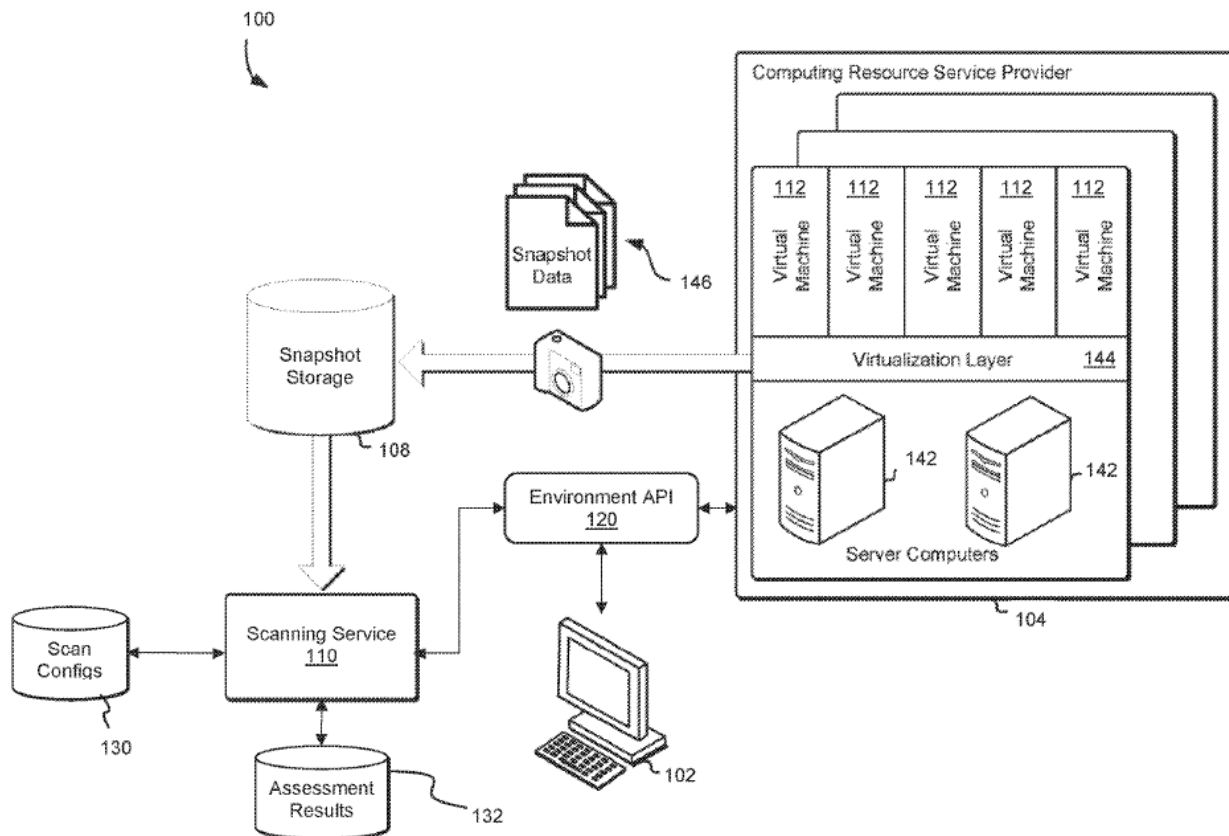
169. Accordingly, Veselov teaches “determining a virtual disk allocated to the protected virtual cloud asset” as recited by claims 3 and 13. It is therefore my opinion that claims 3 and 13 were obvious over Veselov and Basavapatna.

3. Dependent Claims 4 and 14

170. Claim 4 recites the method of claim 2, further comprising “querying a cloud management console of the cloud computing environment to determine the location of the snapshot and the location of the virtual disk.” Claim 14 similarly recites the system of claim 12, wherein the system is further configured to “query a cloud management console of the cloud computing environment to determine the location of the snapshot and the location of the virtual disk.” Given the analogous nature of the additional limitations recited in these claims, I discuss them together in this section. As I explain below, Veselov teaches these additional limitations.

171. Veselov teaches that determining the snapshot’s location can be accomplished by “querying a cloud management console of the cloud computing environment,” as recited. Veselov describes accessing the snapshot by using an environment API or web interface to communicate with the computing resource

service provider hosting the virtual machine. EX1007 (Veselov), 4:19-48 (“In some embodiments, the computing resource service provider 104 may provide, or otherwise be compatible with, an environment application programming interface (API) 120 through which a user device 102 can connect to one or more virtual computing environments of the computing resource service provider 104.”), 5:1-8 (“The scanning service 110 may access the computing resource service provider 104, or any associated virtual computing environments and/or computing resources, via the API 120 or another API.”). An exemplary environment API is depicted in Figure 1:



Id., Fig. 1 (depicting environment API 120); *see also id.*, Figs. 3A-3B (depicting API 338). Although some figures in Veselov do not explicitly show an API mediating the interaction between the scanning service and the hypervisor (*see id.*, Figs. 5A-B), Veselov clarifies that “the user device 502, virtual computing environment 504A, ... as well as their implementation and interactions, are the same as described above for the associated components ... of FIGS. 3A-B.” *Id.*, 16:56-17:8. Also, a person of ordinary skill in the art would have recognized that querying an API was an obvious, efficient, and commonly practiced technique for obtaining snapshot data from a virtual computing environment. *See, e.g., supra*, §§VIII.B-C; EX1015 (VMware vSphere), 164-66; EX1020 (Sancheti), Abstract, 21:42-22:58 (describing use of API calls to interact with snapshots).

172. A person of ordinary skill in the art would have understood this API and its associated user interfaces to be, or at least suggest, a “cloud management console” (or at least suggesting the use of such a console) since they provide various mechanisms for managing the cloud system. Veselov describes examples of user interaction:

For example, the API 120 may be a web-based interface implemented on a web server of the computing resource service provider 104 as described further below; **one or more user interfaces may be transmitted to the user device 102 and displayed thereon**, enabling

the user of the user device 102 to provide settings, commands, software packages, and other user input, to the computing resource service provider 104. Such user input may be used to configure virtual computing environments of the computing resource service provider 104 that are associated with the user, such as via a user account; the user may be required to provide credentials and be authenticated and authorized to modify its virtual computing environments and virtual resource allocations via the API 120.

Id., 4:19-48 (emphasis added). Veselov describes managing the system via a user interface associated with the environment API. *Id.*, 4:49-5:8 (“In some embodiments, the scanning service 110 may use the API 120 to provide user interfaces to the user device 102, enabling the user to configure settings of the scanning service 110 as described further below.”), 5:55-60 (“The virtualization layer 144 may also receive and process API calls from external devices or services.”), 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, such as by using the user device 302 to access the scanning service 310 via one or more APIs 338, web interfaces, and the like.”). At a minimum, using a cloud management console in this context would have been obvious in view of Veselov’s disclosure of remotely managing the system via a user interface associated with the environment API, since such consoles were a well-known and routinely

used mechanism for remote management of cloud systems such as those described in Veselov. *See, e.g., supra*, §§VIII.B-C; EX1009 (Wolf), 221 (“VirtualCenter uses a central console so you can access and manage all your VMs from one location.”), 257-60 (describing generation of snapshot via “VMware management UI”); EX1015 (VMware vSphere), 164-66.

173. Veselov also teaches, or at least suggests, querying this cloud management console “to determine the location of the snapshot and the location of the virtual disk,” as recited. I will first address querying the cloud management console “to determine the location of the snapshot.” I discussed Veselov’s teachings of determining the snapshot’s location with respect to element 1.1. *Supra*, §XII.A.1.b. Veselov further teaches querying the cloud management console to make such a determination, since it teaches that location determinations such as those I discussed with respect to element 1.1 may be triggered by querying the cloud management console. Veselov explains that the scanning service “may access the computing resource service provider 104, or any associated virtual computing environments and/or computing resources, via the API 120 or another API.” EX1007 (Veselov), 5:1-5; *see also id.*, Fig. 1 (depicting scanning service 110 accessing the computing resource service provider 104 via the environment API 120). Likewise, the virtualization layer on the host system can “receive and process

API calls from external devices or services.” *Id.*, 5:59-60. Via a series of messages initiated by the user via the UI/API (i.e., querying the cloud management console as I discussed above), the scanning service “may determine the corresponding virtualization layer 144 for the virtual machine 112” identified in a request from the user 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,” at which point the virtualization layer on the host system may “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.” *Id.*, 6:25-49. As I explained above in section XII.A.1.b (discussing element 1.1), this virtualization layer is one example of a general “location” of the snapshot. Veselov’s querying of the cloud management console thus determines the location of the snapshot, or at a minimum suggests querying the console to make such a determination since such querying was a common feature of VM management tools such as those discussed in Veselov. *See, e.g.*, EX1009 (Wolf), 221 (discussing managing VMs through both console and scripting API), 257 (discussing taking snapshots in VMWare management GUI); EX1015 (VMWare vSphere), 56 (depicting server inventory hierarchy), 124 (discussing datastore management), 164-66 (discussing snapshot management); EX1031 (NIST IT Asset Mgmt), 1 (Executive Summary) (“To effectively manage,

use, and secure each of those assets, you need to know their locations and functions.”), ii (“An effective IT asset management (ITAM) solution can tie together physical and virtual assets and provide management with a complete picture of what, where, and how assets are being used. ITAM enhances visibility for security analysts, which leads to better asset utilization and security.”), 1-4 (describing benefits of security solution that provides API and enables management of virtual and physical assets).

174. I will now turn to querying the cloud management console to determine “the location of the virtual disk.” As an initial matter, I note that claims 4 and 14 recites “the” location of the virtual disk, while neither these claims nor the claims from which they depend expressly introduce “a location of the virtual disk” (e.g., claim 1 merely recites “a location of a snapshot of at least one virtual disk”). Nevertheless, in my opinion, querying a cloud management to determine the snapshot’s location and the virtual disk’s location was still obvious. First, Veselov’s disclosures that I discussed above with respect to querying the cloud management console to determine the snapshot’s location at a minimum would have suggested that such a query would also determine the virtual disk’s location. For example, a person of ordinary skill would have understood that the snapshot of a virtual disk and the underlying virtual disk itself would sometimes have the same general

location (e.g., being located in the same computing environment). This is further supported by Figures 3A-B and 5A-B of Veselov, which depict the snapshot and its associated virtual disk (shown as image 308/508 and logical volume 306/506) in the same virtualized computing environment, which also indicates that they may be in the same virtualization layer. *See also supra*, §XII.B.2 (discussing determination of virtual disk with respect to claims 3 and 13). Second, Veselov teaches using a UI/API (i.e., a cloud management console, as I discussed above) to display and control configuration parameters of the security assessment, such as choice of the target, including target virtual disks. EX1007 (Veselov), 4:19-48 (Displayed UIs allow user to “provide settings, commands, software packages, and other user input, to the computing resource service provider 104. Such user input may be used to configure virtual computing environments of the computing resource service provider 104 that are associated with the user, such as via a user account; the user may be required to provide credentials and be authenticated and authorized to modify its virtual computing environments and virtual resource allocations via the API 120.”), 8:52-9:8 (target may be a “logical volume, physical or virtual storage device”), 11:28-33 (logical volume “also known as a ‘logical drive’ or ‘virtual disk drive’ or ‘virtual storage device’ (e.g., the logical volumes 306, 316)”), 11:34-67 (“the snapshot data of the virtual machine 312 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”), 14:37-63 (“In various embodiments, an owner or administrator of the target computing resources to be assessed may configure one or more security assessments, such as by using the user device 302 to access the scanning service 310 via one or more APIs 338, web interfaces, and the like. 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)...”), 16:27-44 (“[T]he scanning service may provide one or more parameters of a logical volume (e.g., size, file system) to the hypervisor...”). Veselov thus teaches, or at least suggests, querying a cloud management console to perform a wide variety of tasks including selection of a target virtual disk, which typically involved determination of the virtual disk’s location. At a minimum, in view of Veselov’s use of a user interface associated with an environment API to obtain snapshot data, it would have been obvious to a person of ordinary skill in the art to query a cloud management console to locate both snapshots and virtual disks. Identifying such locations by querying a cloud management console was a common, routine task for a person of ordinary skill in the art, and it would have been a natural initial step when performing security assessments on snapshots of virtual disks (such as Veselov’s

security assessments that I discussed above). *See, e.g.*, EX1015 (VMWare vSphere), 16-23 (discussing remote client API), 55-58 (discussing datacenter inventory object hierarchy), 115-20 (discussing datastore management), 124-26 (discussing datastores for VM files, including virtual disks, and querying), 150-52 (discussing setting and accessing VM configuration and location), 155-56 (discussing file locations); EX1031 (NIST IT Asset Mgmt), 1 (Executive Summary) (“Users can now query one system and gain insight into their entire IT asset portfolio”), ii (, 1-4; *see also supra*, §VIII.A (discussing virtual asset locations). Querying a cloud management console to determine the virtual disk’s location also would have been obvious as part of the remediation process that I discussed earlier in relation to claims 2 and 12, since these assessments may identify vulnerabilities associated with the virtual disk, and a person of ordinary skill would have expected the console discussed above to provide query-based determination of the virtual disk’s location so that those vulnerabilities can be resolved. *Supra*, XII.B.1.b; EX1007, 9:9-41 (“[T]he scanning service may cause the snapshot data to include metadata, such as a tag, that identifies the target resource embodied in the snapshot data.”), 10:17-36 (“[T]he scanning service may associate the assessment results with the target resource, such as by identifying the target resource from a metadata tag included in the snapshot data”). Accordingly, Veselov teaches or at least suggests “querying a

cloud management console of the cloud computing environment to determine the location of the snapshot and the location of the virtual disk” as recited by claims 4 and 14. It is therefore my opinion that claims 4 and 14 were obvious over Veselov and Basavapatna.

4. Dependent Claims 5 and 15

175. Dependent claims 5 and 15 depend from independent claims 1 and 11, respectively, but recite substantially similar limitations. The preambles of claims 5 and 15 recite the “method of claim 1” and the “system of claim 11,” respectively, “wherein analyzing the snapshot of the protected virtual cloud asset further comprises....” Both claims then recite identical limitations. Given these similarities, I discuss the additional limitations of claims 5 and 15 together in the following sections.

a. Elements 5.1 and 15.1

176. Elements 5.1 and 15.1 recite “parsing a copy of the snapshot.” As I explain below, Veselov teaches these elements.

177. 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:

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), 9:16-25 (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.A.1.b, XII.A.1.c (elements 1.1 and 1.2). A person of ordinary skill 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.

178. Veselov also describes “parsing” this copy of the snapshot as part of the analysis. As I discussed above in §VIII.D.2, parsing was a well-known technique that included, for example, processing data to transform it into a more structured format. *See* EX1035 (Lengyel), 389; EX1039 (Payne), 14; EX1041 (Kornfeld), 3:1-9; EX1042 (Rubin), 10:45-55, 12:54-60; EX1043 (Jiang), 12:10. Veselov teaches such parsing in several ways. First, Veselov 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 (Veselov), 8:21-23 (emphasis added). Veselov 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.”). Veselov 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.” EX1007 (Veselov), 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)). A person of ordinary skill would have viewed this type of processing to transform the data into a scannable format as “parsing” the copy of the snapshot as recited, or at a minimum suggesting such parsing.

179. 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 virtual machine data as a flat file to detect cybersecurity risks. These VMI techniques, for example, commonly included parsing data files to prepare them for security scans (without an agent installed in the inspected VM). *Supra*, §VIII.D.2 (background); *see also* EX1033 (Garfinkel), 1; EX1034 (Tang), 2; EX1035 (Lengyel), 389; EX1037 (Bauman), 11-15; EX1038 (Suneja), 2; EX1039 (Payne), 14; EX1042 (Rubin), 10:45-55, 12:54-60; EX1043 (Jiang), 12:10. 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. *Id.* A person of ordinary skill in the art, 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.

180. Accordingly, Veselov teaches “parsing a copy of the snapshot” as recited by elements 5.1 and 15.1.

b. Elements 5.2 and 15.2

181. Elements 5.2 and 15.2 recite “scanning the parsed copy to detect the potential cyber threats.” As I explain below, Veselov teaches these elements.

182. Veselov teaches “scanning the parsed copy.” As discussed above with respect to elements 5.1 and 15.1, Veselov teaches various ways of parsing the copy of the snapshot such that the copy 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.*, 3:61-67 (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 Figure 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., 18:26-44 (emphasis added). 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.A.1.e (element 1.4). Accordingly, Veselov teaches "scanning the parsed copy to detect the potential cyber threats" as recited by elements 5.2 and 15.2.

183. At a minimum, scanning a parsed copy of a snapshot to detect cyber threats would have been obvious. As discussed above with respect to elements 5.1 and 15.1, 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 person of ordinary skill in the art, 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 of the snapshot as taught by Veselov. EX1007 (Veselov), 18:26-44; *see also id.*, 24:46-53 (“In other instances, methods, apparatuses, or systems that would be known by one of ordinary skill have not been described in detail so as not to obscure the claimed subject matter.”).

c. Elements 5.3 and 15.3

184. Elements 5.3 and 15.3 both recite “wherein the potential cyber threats include known and unknown vulnerabilities.” As I explain below, Veselov and Basavapatna teach these elements.

185. Veselov and Basavapatna each describe detecting potential cyber threats that include “known” vulnerabilities. As an initial matter, a person of ordinary skill 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, “unknown” 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.

186. 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 ’735 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)).

187. 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, bad configurations, 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.

188. 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.”).

189. 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 person having ordinary skill in the art also had reason to apply Basavapatna’s teachings in the context of virtual cloud systems such as those taught by Veselov. At a minimum, security scanning for both known and unknown vulnerabilities would have been obvious. I previously discussed reasons for combining Veselov’s teachings with Basavapatna’s teachings regarding alerting detected threats (*supra*, §XII.A.4), and that reasoning applies equally to all types of threats discussed in Basavapatna, including unknown vulnerabilities. Both known and unknown

vulnerabilities were routinely addressed by security scans (*supra*, §VIII.D.1), and it would have been clear to a person of ordinary skill 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.

190. Accordingly, Veselov and Basavapatna (individually or in combination) teach “wherein the potential cyber threats include known and unknown vulnerabilities” as recited by elements 5.3 and 15.3.

d. Elements 5.4 and 15.4

191. Elements 5.4 and 15.4 recite “wherein the detection is based on a type of vulnerability.” As I explain below, Veselov teaches these elements.

192. Veselov teaches detecting vulnerabilities “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). Veselov also explains that the security assessment can apply rules packages as CVEs, which a person of ordinary skill would have understood apply detection techniques based on various types of vulnerabilities. *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.”); *see also supra*, §XII.A.1.d.

193. Accordingly, Veselov teaches “the detection is based on a type of vulnerability” as recited by elements 5.4 and 15.4. It is therefore my opinion that claims 5 and 15 were obvious over Veselov and Basavapatna.

5. Dependent Claims 6 and 16

194. Dependent claims 6 and 16 depend from claims 5 and 15, respectively, but recite substantially similar limitations. The preamble of claim 6 recites the “method of claim 5, wherein scanning the parsed copy further comprises any one of ...” and the preamble of claim 16 recites the “system of claim 15, wherein scanning the parsed copy further comprises any one of...” The claims then list the same four steps, which I will label 6.1 to 6.4 (or 16.1 to 16.4) to simplify further discussion. Given these similarities in the limitations of claims 6 and 16, I discuss them together below. These steps are listed below with the labels for claim 6:

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

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

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

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

195. 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 6 and 16 are satisfied if even one of these steps is met. As I explain below, Veselov further teaches elements 6.1 and 16.1 and elements 6.4 and 16.4, either one of which is sufficient to satisfy claims 6 and 16. It is therefore my opinion that claims 6 and 16 were obvious over Veselov and Basavapatna.

a. Elements 6.1 and 16.1

196. Veselov teaches that scanning the parsed copy further comprises “checking configuration files of applications and operating system installed in the protected virtual cloud asset” as recited elements 6.1 and 16.1.

197. Veselov explains that the snapshot—a copy of which is parsed as I explained when discussing claims 5 and 15—includes configuration files associated with the virtual machine. 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.”).

198. Veselov also specifically 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 cloud asset” 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 Figure 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.

199. Veselov further teaches analyzing such configurations 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 (same). It 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”); 18:32-50 (describing scanning a parsed copy of a 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 person of ordinary skill in the art that Veselov’s assessment of the snapshot—which includes parsing a copy of a snapshot containing “operating system information” and

configurations of the “software packages installed” and then scanning that parsed copy—including “checking configuration files of applications and operating systems installed in the protected virtual cloud asset” as recited. *See id.*, 7:16-22; *supra*, §§XII.B.4.a, XII.B.4.b (discussing parsing). At a minimum, Veselov suggests such checking in view of the teachings I discussed above.

200. Accordingly, Veselov teaches the additional limitations of elements 6.1 and 16.1.

b. Elements 6.4 and 16.4

201. Veselov teaches that scanning the parsed copy further comprises “analyzing machine memory stored in the snapshot to deduce what applications and modules executed in the protected virtual cloud asset” as recited elements 6.4 and 16.4 for reasons similar to those I discussed in the previous section.

202. First, Veselov teaches “analyzing machine memory stored in the snapshot.” It explains that the snapshot of a virtual machine, 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.A.1.c, XII.A.1.d) and claims 5 and 15 (*supra*, §§XII.B.4.a, XII.B.4.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. Veselov's security assessment, which can involve parsing of a copy of the snapshot, includes "analyzing machine memory stored in the snapshot."

203. Veselov also teaches performing this analysis "to deduce what applications and modules executed in the protected virtual cloud asset," as recited. As I explained in the previous section, the scanned data can include information indicating which software packages were on the 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”). A person of ordinary skill would have understood these references to 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 virtual machines. Moreover, given Veselov’s discussion of performing CVE assessments based on a list of “all software packages installed on the virtual machine,” Veselov teaches, or at least suggests, that the security analysis includes deducing which of these applications and modules are executing in the protected virtual cloud asset (e.g., as part of a determination of whether that list includes applications or modules addressed in the CVE).

204. Veselov therefore teaches that scanning the parsed copy further comprises “analyzing machine memory stored in the snapshot to deduce what applications and modules executed in the protected virtual cloud asset” as recited elements 6.4 and 16.4.

6. Dependent Claims 7 and 17

205. Dependent claims 7 and 17 depend from independent claims 5 and 15, respectively, but recite substantially similar limitations. The preamble of claim 7 recites the “method of claim 5, further comprising...” and the preamble of claim 17 recites the “system of claim 15, wherein the system is further configured to...” Both

claims then recite analogous limitations. Given these similarities, I discuss the analogous limitations of claims 7 and 17 together in the following sections

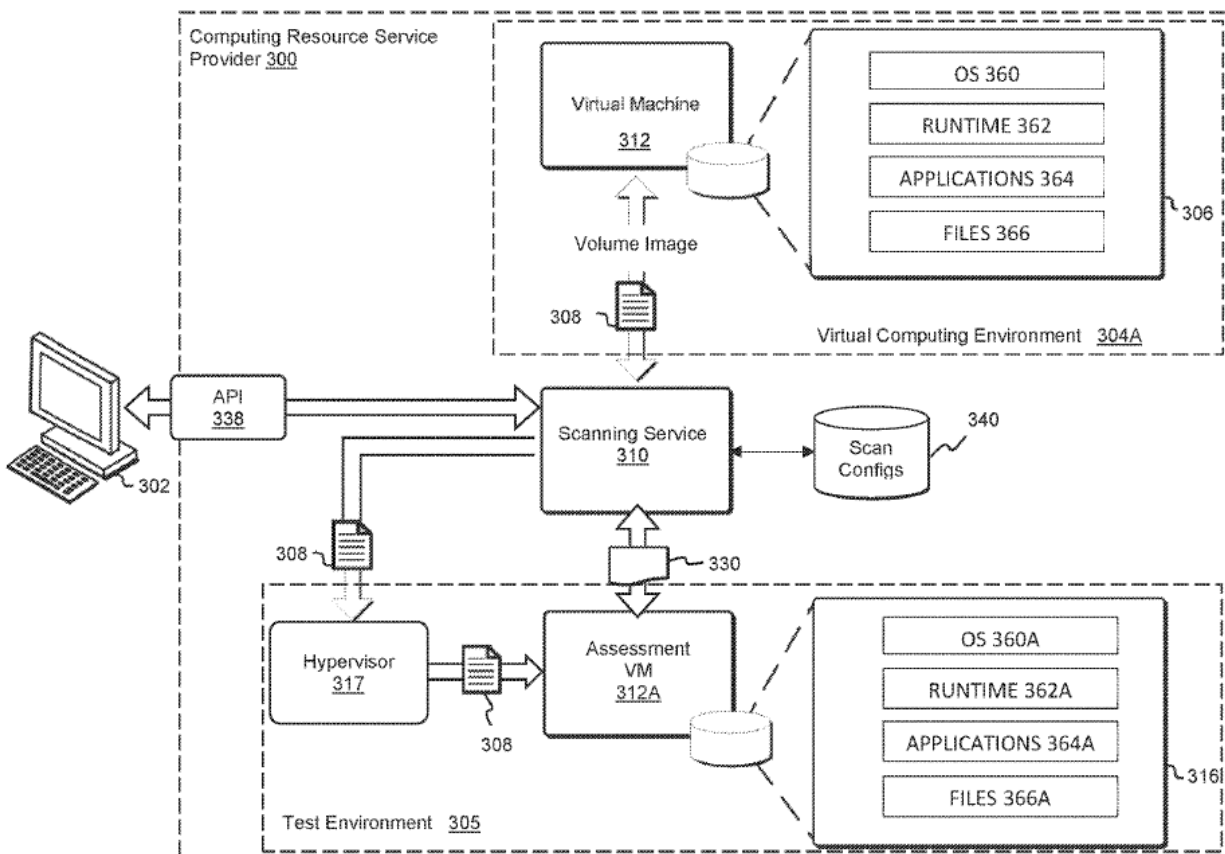
a. Elements 7.1 and 17.1

206. Element 7.1 recites “instantiating an instance copy of the protected virtual cloud asset from the snapshot.” Likewise, in element 17.1 the system is further configured to “instantiate an instance copy of the protected virtual cloud asset from the snapshot.” Veselov teaches these elements. 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 5 and 15) 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.B.5) and the VM-instantiation techniques I discuss below.

207. 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 virtual machine. 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.”).

208. Veselov also depicts such instantiation of an instance 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).

209. Accordingly, Veselov teaches “instantiating an instance copy of the protected virtual cloud asset from the snapshot” as recited by elements 7.1 and 17.1.

b. Elements 7.2 and 17.2

210. Element 7.2 recites “monitoring all activity performed by the instance copy of the protected virtual cloud asset.” Likewise, in element 17.2, the system is further configured to “monitor all activity performed by the instance copy of the protected virtual cloud asset.” Veselov teaches these elements.

211. Veselov teaches monitoring “all activity performed by the instance copy of the protected virtual cloud asset.” Discussing the virtual machine 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.*, 15:60-16:1; *see also id.*, 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)).

212. Accordingly, Veselov teaches “monitoring all activity performed by the instance copy of the protected virtual cloud asset” as recited by elements 7.2 and 17.2. It is therefore my opinion that claims 7 and 17 were obvious over Veselov and Basavapatna.

7. Dependent Claims 9 and 19

213. Claim 9 recites the “method of claim 1, wherein the protected virtual cloud asset includes any one of: a virtual machine, a software container, a micro-service.” Claim 19 similarly recites the “system of claim 11, wherein the protected

virtual cloud asset includes any one of: a virtual machine, a software container, 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 9 and 19 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.

214. 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.”), 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)). I also explained why Veselov teaches that the protected virtual cloud asset can be a virtual machine when discussing element 1.1. *Supra*, §XII.A.1.b. Veselov also contemplates protecting software containers as part of the same cloud computing infrastructure. *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)). Because containers are simply partitions of virtual machines, a person of ordinary skill would have understood a security system that protects virtual machines while also discussing containers would be usefully applied to protect containers within the virtual machines.

215. Accordingly, Veselov teaches “the protected virtual cloud asset includes any one of: a virtual machine, a software container, a micro-service” as recited by claims 9 and 19. It is therefore my opinion that claims 9 and 19 were obvious over Veselov and Basavapatna.

XIII. CONCLUDING STATEMENTS

216. 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.

217. 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: 01/04/2024

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

XIV. APPENDIX A – MATERIALS CITED

Exhibit No.	Description
1001	U.S. Patent No. 11,431,735 to Shua
1003	<i>Curriculum Vitae</i> of Dr. Angelos Stavrou
1004	File History of U.S. Patent Application No. 16/585,967 ('735 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 et al., "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	Microsoft Computer Dictionary, 5 th ed., (2002)
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)

Exhibit No.	Description
	https://doi.org/10.6028/NIST.SP.800-190
1017	U.S. Patent No. 11,106,784 B2 to Rosendahl et al.
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

Exhibit No.	Description
	Switzerland AG (2018) https://doi.org/10.1007/978-3-030-04834-1_9
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.20171.026
1035	Lengyel, “Scalability, Fidelity and Stealth in the DRAKVUF Dynamic Malware Analysis System (2014) https://doi.org/10.1145/2664243.2664252
1036	U.S. Patent No. 8k613,080 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) http://dx.doi.org/10.1145/2731186.2731196

Exhibit No.	Description
1039	Payne, “Simplifying Virtual Machine Introspection Using LibVMI”, SANDIA Report-SAND2012-7818, Sandia National Labs. (Sep. 2012)
1040	U.S. Publication No. 2013/0111018 A1, Published May 2, 2013, 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, Published December 12, 2019, to Canzanese Jr.
1046	U.S. Patent 10,783,241 B2 to Crabtree