

Filed on behalf of: Orca Security Ltd.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

WIZ, INC.,
Petitioner,

v.

ORCA SECURITY LTD.,
Patent Owner.

Case IPR2024-00864
Patent 11,663,032

**DECLARATION OF DR. DAVID KAEI
IN SUPPORT OF
PATENT OWNER'S RESPONSE**

Table of Contents

I.	Introduction.....	3
II.	Qualifications.....	4
III.	Summary Of Materials Reviewed And Considered	7
IV.	Legal Standards.....	8
V.	Level Of Skill In The Art And Perspective Applied In This Declaration.....	12
VI.	Background.....	15
	A. Overview of the '032 Patent.....	15
	B. Claim Construction.....	20
	1. “Analyzing the Snapshot”	20
VII.	Overview of the Asserted References.....	23
	A. Overview of Veselov	23
	B. Overview of Hufsmith	30
	C. Overview of Hutchins.....	43
VIII.	The Claims Of The '032 Patent Would Not Have Been Obvious ...	45
	A. Ground 1: Claims 1-11 and 13-25 Would Not Have Been Obvious Over Veselov and Hufsmith.....	45
	1. A Person of Ordinary Skill in the Art Would Not Have Been Motivated to Combine Veselov and Hufsmith and Would Not Have Had a Reasonable Expectation of Success in Doing So.....	45
	2. Even If Combined, Veselov and Hufsmith Do Not Disclose or Suggest Multiple Limitations in the Independent Claims.....	55
	B. Ground 2: Claim 12 Would Not Have Been Obvious Over Veselov, Hufsmith, and Hutchins.....	81
	1. A Person of Ordinary Skill in the Art Would Not Have Been Motivated to Combine Veselov, Hufsmith, and Hutchins and Would Not Have Had a Reasonable Expectation of Success in Doing So	82

2. Even If Combined, Veselov, Hufsmith, and Hutchins
Do Not Render Claim 12 Obvious 84

IX. Objective Indicia of Non-obviousness..... 84

A. Orca’s SideScanning™ Uses The ’032 Patent Inventions 84

B. Long-Felt, Unresolved Need 85

C. Orca’s claimed inventions overcame industry skepticism to
become widely accepted--and widely praised—in the
industry. 90

D. Extensive Industry Praise Supports Non-Obviousness of the
Claims..... 94

E. Evidence Shows Wiz Copied Orca’s SideScanning™..... 101

1. Evidence shows Wiz had access to Orca’s invention 102

2. Evidence shows Wiz’s Workload Scanning is
substantially similar to Orca’s SideScanning™ 119

F. Commercial Success..... 141

X. CONCLUSION..... 143

I. Introduction

1. I, David Kaeli, submit this declaration on behalf of Orca Security Ltd. (“Patent Owner”) to provide my opinions concerning the validity of claims 1-25 of U.S. Patent No. 11,663,032 (the “’032 patent,” Ex.1001), in support of Patent Owner’s Response to the Petition for *Inter Partes* review of the ’032 patent, which I understand was filed by Wiz, Inc. (“Petitioner”) and accompanied by a Declaration of Dr. Angelos Stavrou (the “Stavrou Declaration”), among other exhibits. I further understand that on December 9, 2024, this *Inter Partes* review (IPR) was instituted (Paper 8).

2. I have previously provided my opinions in a Declaration (Ex.2001) in support of the Patent Owner’s Preliminary Response to the Petition. Although I believe that declaration is accurate in all respects, this new Declaration represents my considered opinion and supersedes my previous Declaration.

3. I am more than eighteen years of age, and I am a citizen of the United States, currently residing in the Commonwealth of Massachusetts.

4. I am being compensated for the time I spent preparing this Declaration at my customary rate. The opinions I express in this Declaration are my own. My compensation does not depend upon the outcome of this IPR, the contents of this Declaration, or any testimony I may provide in this IPR.

5. This Declaration summarizes the opinions I have formed to date. I reserve the right to modify or update my opinions, if necessary, based on my review and analysis of information or materials that I may receive after this Declaration is submitted. I further reserve the right to respond to any new or updated positions Wiz or its expert(s) may take in this IPR.

II. Qualifications

6. In formulating my opinions, I have relied upon my knowledge, training, and experience in the relevant art. My qualifications, along with a more detailed summary of my background and publications, are stated more fully in my *curriculum vitae*, which has been provided as Exhibit 2002. Here, I provide a brief summary of my qualifications.

7. I am currently the Director of Northeastern University's Computer Architecture Research Laboratory and a Distinguished Professor of Electrical and Computer Engineering. I have been a faculty member at Northeastern since September 1993, in the Electrical and Computer Engineering Department.

8. For over thirty years, I have pursued research across a broad range of computer architectures and technologies, including memory systems, storage systems, virtualization, and secure computing systems. I have published over 400 refereed journal and conference papers in the areas of computer architecture and computer security. I also have supervised over forty Ph.D. theses in these areas.

9. I am a Fellow of the Institute of Electrical and Electronic Engineers (IEEE) and a Fellow of the Association of Computing Machinery (ACM). I have served two terms as the Chair of the Executive Committee of the IEEE Technical Committee on Computer Architecture. I am serving in my second terms as the Editor-in-Chief of the ACM Transactions on Architecture and Code Optimization. I am also an Associate Editor for Elsevier's Journal of Future Generation Computer Systems.

10. From 2011-2014, I served as Chief Science Advisor of Apposable Software. From 2007-2011, I served as Chief Technology Officer of NUIC Technologies, where I worked on designing secure virtual appliances. From 2005-2011, I consulted for Akorri Networks, where, among other things, I helped develop intellectual property relating to storage virtualization. In 2000, I served as a consultant for InCert Corporation.

11. Before I joined Northeastern's faculty in 1993, I spent twelve years working for IBM. From 1981-1986, I worked on the design and delivery of the IBM System/370 3090 mainframe system. From 1986-1993, I worked at the IBM T.J. Watson Research Center, designing, and analyzing IBM Personal Systems 2 computers and associated software. During that time, I also worked on the design of a new Application System/400 multi-processor system.

12. I am an active educator and researcher and have extensive experience in the broader field of computer architecture. I am a co-inventor on patents targeting virtualization security. *See, e.g.,* Moffie, et al., *VMM-Based Intrusion Detection System*, (US Patent 8,719,936, May 6, 2014). I am a co-inventor on patents relating to virtual machine management. *See, e.g.,* Smirnov, et al., *Systems and Methods for Determining Placement of Virtual Machines*, (US Patent 8,099,487, January 17, 2012). And, I have conducted academic research on, and published multiple papers relating to, hardware and software security, including papers discussing security on virtualized infrastructure, computer system vulnerabilities and protections, and monitoring virtualized computing environments. *See, e.g.,* Azmandian et al., *Securing Virtualized Execution Environments Through Machine Learning-based Intrusion Detection*, IEEE International Workshop on Machine Learning for Signal Processing (2015); Azmandian et al., *Virtual Machine Monitor-based Lightweight Intrusion Detection*, ACM SIGOPS Operating Systems Review (2011); Sabbagh et al., *Overdrive Fault Attacks on GPU*, IEEE/IFIP International Conference on Dependable Systems (2021); Ahn et al., *Trident: A Hybrid Correlation-Collison GPU Cache Timing Attack for AES Key Recovery*, IEEE International Symposium on High Performance Computer Architecture (2024); Arzuaga, et al., *Increasing Power/Performance Resource Efficiency on Virtualized Enterprise Servers*, IEEE

Computing Frontiers (2011); Azmandian, et al., *Workload Characterization at the Virtualization Layer*, IEEE Annual International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, (2011).

13. I earned a Bachelor of Science degree in Electrical Engineering from Rutgers University in 1981. I earned a Master of Science degree in Computer Engineering from Syracuse University in 1985. I earned a Ph.D. in Electrical Engineering from Rutgers University in 1992.

III. Summary Of Materials Reviewed And Considered

14. The opinions contained in this Declaration are based on the documents I reviewed and my knowledge and professional judgment. In forming the opinions expressed in this Declaration, while drawing on my experience, I reviewed all of the documents cited in this declaration, as well as the Petition and the Institution Decision.

15. My opinions are additionally guided by my appreciation of how a person of ordinary skill in the art would have understood the claims of the '032 patent at the time of the invention, which I have been asked to assume is January 28, 2019. I will sometimes refer to this time as the "relevant time."

IV. Legal Standards

16. I am not an attorney. However, I have been informed by counsel of certain legal standards and have applied those standards, described below, in formulating my opinions.

17. I understand that the claims of a patent define the limits of a patentee's exclusive rights. I understand that to determine the scope of the claimed invention, courts may construe claim terms, the meaning of which the parties may dispute. I understand that claim terms should generally be given their ordinary and customary meaning as understood by a person of ordinary skill in the art at the time of filing of the patent application.

18. I understand that claims must be construed in light of and consistent with the intrinsic evidence. In this context, I understand that intrinsic evidence includes the claims themselves, the written disclosure in the specification, and the patent's prosecution history. I understand that the specification is always highly relevant to the claim construction analysis and often is the single best guide to the meaning of a disputed term. I understand that extrinsic evidence may also be considered when construing claims and may include, for example, technical dictionaries, technical publications and books, treatises, and expert testimony, though the intrinsic record will be more persuasive as to the meaning of a particular term.

19. I understand that one or more prior art references can render a patent claim obvious to a person of ordinary skill in the art if the differences between the subject matter set forth in the patent claim and the prior art are such that the subject matter of the claim would have been obvious at the time the claimed invention was made.

20. I understand that for a reference to qualify as a relevant prior art reference in an obvious analysis, the reference must be “analogous”—meaning it is from the same field of endeavor as the patented technology, and if not, then the reference must be reasonable pertinent to the particular problem facing the inventors of the patent at issue. I also understand that a reference is reasonably pertinent if a person of ordinary skill in the art would have reasonably consulted those references and applied their teachings in seeking a solution to the problem facing the inventors.

21. In analyzing obviousness, I understand that it is important to consider the scope of the claims, the level of skill of a person of ordinary skill in the relevant art, the scope and content of the prior art, and the differences between the prior art and the claims. I understand that claimed subject matter can be obvious when a person of ordinary skill in the art would have been motivated to combine pre-existing disclosures in the prior art, and that combination yields no more than what a person of ordinary skill in the art would have expected. However, I further

understand that obviousness cannot be based on the hindsight combination of components selectively culled from the prior art.

22. I also understand from counsel that to guide against hindsight, obviousness must be considered in view of any evidence demonstrating certain objective factors that may show that an invention was not obvious from the prior art, even if the differences between the prior art and the claimed invention seem slight. I have been informed by counsel that “objective indicia” of non-obviousness, can include, for example, evidence of commercial success of the invention, evidence of a long-felt need that was solved by an invention, evidence that others copied an invention, evidence that an invention achieved an unexpected result, evidence of industry praise of an invention, evidence of skepticism by others, evidence of licensing an invention, or evidence that others failed to solve the same problem that an invention solved.

23. I have been informed that in order to have relevance and weight in an obviousness analysis, the patentee must show that there is a “nexus” between the objective indicia and the merits of the claimed inventions. I understand that the focus of the nexus requirement looks to the patent claims as a whole, and that nexus does *not* have to be based only on limitations that are allegedly not shown in a primary prior art reference. For example, even if every limitation of a claim is shown to be *individually* known in the prior art, that does not preclude nexus where

the objective indicia evidence is tied to the combination of features as a whole and the combination was not previously known.

24. I have been informed that commercial success is relevant as an indication of nonobviousness because the law presumes an idea would successfully have been brought to market sooner, in response to market forces, had the idea been obvious to persons skilled in the art. Thus, the law deems evidence of (1) commercial success, and (2) some causal relation or “nexus” between an invention and commercial success of a product embodying that invention, helpful in determining whether an invention was nonobvious.

25. Similarly, I have been informed that recognition by others of skill in the art that the inventor’s contribution was an invention may support a finding of nonobviousness. This is because the law recognizes that industry participants, especially competitors, are not likely to praise an obvious advance over the known art. Accordingly, the fact that a product which embodies the patent claims garnered awards or otherwise received praise will weigh against an assertion that the same claim would have been obvious.

26. I understand that copying of the claimed inventions can also support an inference of nonobviousness, because it can both show a type of “praising” of the invention and suggesting that the prior art was insufficient. I have been informed that, like all other objective indicia, copying requires a showing of a

nexus between the alleged copying and the claimed invention. I also understand that not every competing product that arguably falls within the scope of a patent is evidence of copying, because otherwise every infringement suit would automatically confirm the nonobviousness of the patent. Rather, a nexus can be shown for copying based on, for example, that a competitor had access to the patentee's product or a disclosed embodiment in the patent and then created a product having substantial similarity to the patented product.

V. Level Of Skill In The Art And Perspective Applied In This Declaration

27. I understand that certain issues relating to validity must be judged from the perspective of a person of ordinary skill in the art. I further understand that factors that may be considered in determining the level of skill of a person of ordinary skill in the art include, but are not limited to, (1) the type of problems encountered in the art; (2) prior art solutions to those problems; (3) rapidity with which innovations are made; (4) sophistication of the technology; and (5) the educational level of active workers in the field.

28. I have been informed and understand that a person of ordinary skill in the art is a hypothetical person who is presumed to be aware of all pertinent prior art, thinks along conventional wisdom in the art, and is a person of ordinary creativity. Importantly, this hypothetical person is of ordinary skill in the art to which the claimed subject matter pertains.

29. With respect to the '032 patent, a person of ordinary skill in the art at the time of the earliest priority date, January 28, 2019, would have been familiar with techniques for securing virtual cloud assets against cyber threats, and underlying virtualization technologies used with such techniques.

30. A person of ordinary skill in the art would have had at least a Bachelor's degree in computer science, computer engineering, or a related field, and two years of industry experience or academic research experience in cyber security and virtualization, including in cloud computing cybersecurity. In my view, more education can supplement relevant experience and vice versa.

31. At the time of the earliest priority date for the '032 patent (January 28, 2019), I met the qualifications to be a person of ordinary skill in the art, and I believe that I also have a sufficient level of knowledge, experience, and education to provide expert opinions in the field of the '032 patent.

32. While my own level of skill exceeded that of the ordinary level of skill as of January 2019, I am well-acquainted with the actual performance and capabilities of a person of ordinary skill in the art as defined above. For example, during the relevant timeframe, I was supervising and evaluating the performance of such individuals, such as my graduate students who were performing research in this area. Furthermore, my participation in professional organizations and in

conferences and other events exposes me to the state of the industry and the level of experience of individuals who, like I did, had at least the ordinary level of skill.

33. I have applied the above-defined perspective of a person of ordinary skill in the art, as of January 2019, in forming and expressing my opinions. Unless otherwise stated, my testimony below refers to the knowledge of one of ordinary skill in the art at that relevant time.

34. In the Stavrou Declaration (Ex.1002), Dr. Stavrou sets out the following articulation of the level of ordinary skill:

In my opinion, a person of ordinary skill in the art (“POSA”) at the relevant time would have had (1) at least a bachelor’s degree in computer science, computer engineering, electrical engineering, or a related degree, and (2) would also have 2-3 years of professional experience with cyber security analysis and virtualization. Such experience includes, for example, malware analysis, security analysis of cloud computing systems, and security analysis of virtual machines. ... Additional experience can compensate for less education and vice versa.

Ex.1002 (Stavrou Decl.), ¶22. At the time of the earliest priority date for the ’032 patent (January 28, 2019), I met Dr. Stavrou’s proposed qualifications to be a person of ordinary skill in the art, and my opinions in this Declaration would

not change if the Board were to adopt and apply that proposed level of ordinary skill.

VI. Background

A. Overview of the '032 Patent

35. The '032 patent is titled “Techniques for Securing Virtual Machines by Applicant Use and Analysis” and generally relates to innovative systems and methods for securing virtual cloud assets, such as VMs. Ex.1001 ('032 patent), Abstract, cls. 1, 18, 22.

36. The '032 patent explains that, around the time of its priority date (January 2019), public cloud service providers, like Amazon, Microsoft, and Google, had “increasingly adapted their applications to be run from multiple cloud computing platforms,” in which “[v]irtualization,” which “allow[s] multiple applications and users to share the same cloud computing infrastructure,” “plays a key role.” Ex.1001 ('032 patent), 1:20-29. The '032 patent also provides context that “virtualization can be achieved by means of virtual machines,” each of which can “emulate[] a number of ‘computers’ or instances, all within a single physical device.” Ex.1001 ('032 patent), 1:31-33.

37. The '032 patent discusses problems that existed with trends in the cybersecurity field. For example, the '032 patent discusses that “applications and virtual machines running on top of virtualization technologies are also vulnerable

to some cyber threats,” e.g., from “vulnerable software applications or infected operating systems.” Ex.1001 (’032 patent), 1:42-46. And, the ’032 patent also explains ways the industry had tried to deal with such issues. For example, the ’032 patent discusses “traffic inspection,” which is “[t]raditionally ... performed by a network device connected between a client and a server ... hosting virtual machines”; using “network scanner[s] deployed out of path” to query servers to determine if applications possessing security threats are executed; “traffic monitor[ing],” which “listens to traffic flows between clients and [a] server”; and using “vulnerability management and security assessment solutions ... based on agents installed in each server in a cloud computing platform or data center.” Ex.1001 (’032 patent), 1:46-2:14. As the ’032 patent also explains, though, these methods suffered from disadvantages: “Traffic inspection may not provide an accurate indication of the security status of [a] server due to inherent limitations, such as encryption” and lack of access to pertinent data. Ex.1001 (’032 patent), 1:52-55. With network scanners deployed out of path, servers “may not respond” or “expose [] necessary data,” and network configuration- and credential-related issues can be problematic. Ex.1001 (’032 patent), 1:60-67. Traffic monitoring “detect[s] threats only based on the monitored traffic” and thus can miss server “misconfiguration[s]” and “vulnerabilities in software executed by the server.” Ex.1001 (’032 patent), 2:4-9. And “[u]sing agents is a cumbersome

solution for a number of reasons, including IT resource management, governance, and performance.” Ex.1001 ('032 patent), 2:14-16. “For example, installing agents in a large data center may take months.” Ex.1001 ('032 patent), 2:16-17. The '032 patent's systems and method “overcome the[se] deficiencies.” Ex.1001 ('032 patent), 2:18-19.

38. The '032 patent aims to solve such problems. Ex.1001 ('032 patent), 3:35-36, 3:47-63. Figure 1B of the '032 patent, reproduced below, illustrates a diagram of a cloud computing platform to which the '032 patent's techniques for scanning for embedded vulnerabilities in machines apply:

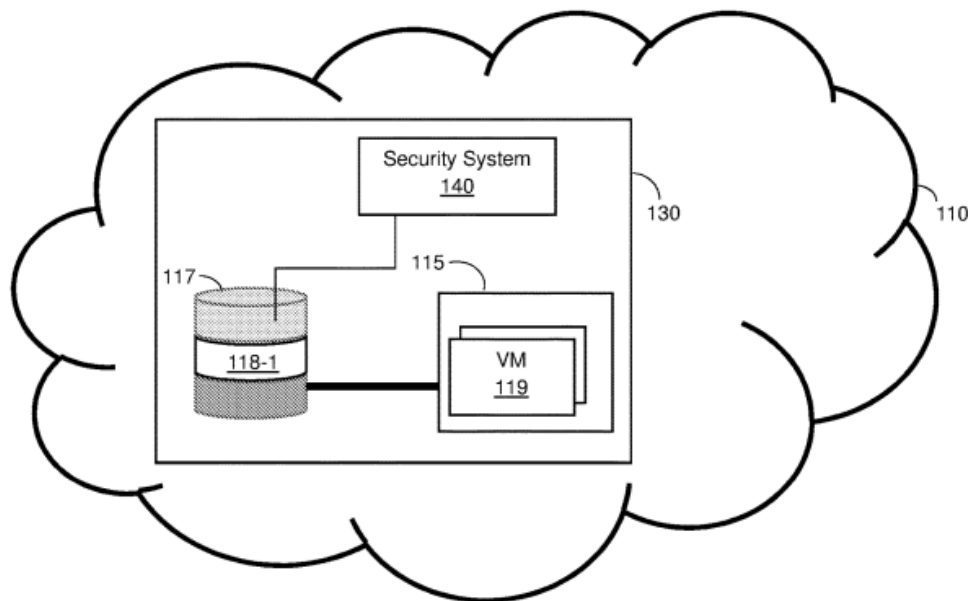


FIG. 1B

39. The cloud computing platform of the '032 patent includes a server (shown as no. 115 in Figure 1B) that hosts one or more VMs (shown as no. 119 in Figure 1B). Ex.1001 ('032 patent), 3:48-50. The cloud computing platform also has storage (shown as no. 117 in Figure 1B) which acts as storage space for the platform's server and "emulates" virtual disks (shown as no. 118-1 in Figure 1B) for the VMs. Ex.1001 ('032 patent), 3:50-51, 3:55-60. The storage may be part of the server or, as shown in Figure 1B, separated from the server. Ex.1001 ('032 patent), 3:55-60. In the example shown in Figure 1B, a "high-speed connection" such as an "optic fiber" connects the storage with the server. Ex.1001 ('032 patent), 3:55-60. Figure 1B also shows a client environment (shown as no. 130 in Figure 1B) which interfaces with a security system (shown as no. 140 in Figure 1B) using "APIs" or through "services provided by the cloud computing platform" (not shown in Figure 1B). Ex.1001 ('032 patent), 4:16-21. The security system is configured to perform the steps of the '032 patent's invention and may be separate from the client environment, as shown in Figure 1B, or part of the client environment. Ex.1001 ('032 patent), 4:1-15. The challenged claims of the '032 patent demonstrate the embodiment where the security system interfaces with the client environment through APIs. Ex.1001 ('032 patent), cls. 1, 18, 22.

40. The security system of the '032 patent interfaces with the client environment so that it can perform scanning operations and detect any "vulnerable

software, non-secure configurations, exploitation attempts, compromised assets, data leaks, data mining” and other risks or vulnerabilities. Ex.1001 ('032 patent), 4:50-56. For example, the security system can detect vulnerable code (Ex.1001 ('032 patent), 5:31-33), cyber threats that do not represent vulnerabilities such as unencrypted sensitive data (Ex.1001 ('032 patent), 6:21-32), and changes in sensitive machine areas such as “added or changed application files without installation” (Ex.1001 ('032 patent), 6:33-39). The security system is also configured to provide “incident response, anti-ransomware, and cyber-insurance” through access to the system’s “security posture” (Ex.1001 ('032 patent), 4:50-56), generate reports on detected vulnerabilities and cyber threats, including, “for example, as alerts” (Ex.1001 ('032 patent), 7:53-55), and mitigate detected vulnerabilities and cyber threats. Ex.1001 ('032 patent), 7:59-8:6. Mitigation actions may include, for example, blocking traffic from untrusted networks, quarantining an infected VM, or halting the operation of the VM. Ex.1001 ('032 patent), 8:1-6.

41. In order to accomplish the scanning and detection operations I discuss above, the '032 patent provides a series of steps. The '032 patent provides multiple techniques for analyzing snapshots to detect different types of vulnerabilities and cyber threats. Ex.1001 ('032 patent), 7:59-8:6. For example, the snapshot may be parsed and analyzed by the security system, the system may match installed

application lists to a known list of vulnerable applications, and the system may check configuration files of applications. Ex.1001 ('032 patent), 5:20-25, 5:34-36, 5:49-54. The security system then generates reports on detected vulnerabilities and cyber threats, including, “for example, as alerts” that may be filtered and prioritized. Ex.1001 ('032 patent), 7:53-55. The security systems and methods also “may cause an execution of one or more mitigation actions,” including disabling a VM from execution, updating a VM with patches, blocking traffic from untrusted networks, quarantining an infected VM, and preventing a VM from launching. Ex.1001 ('032 patent), 6:61-67, 8:1-6.

B. Claim Construction

42. In addition to the explicit constructions discussed below, I have applied the plain and ordinary meanings of the claim terms in the '032 patent, as a person of ordinary skill in the art would have understood them in the context of that patent in approximately January 2019.

1. “Analyzing the Snapshot”

43. Claims 1 and 18 recite “analyzing the snapshot of the at least one virtual disk by matching installed applications with applications on a known list of vulnerable applications.” Similarly, claim 22 recites “analyze the snapshot of the at least one virtual disk by matching installed applications with applications on a

known list of vulnerable applications.” Dr. Stavrou labels these as elements 1.3, 18.3, and 22.3. Ex.1002 (Stavrou Decl.), ¶137.

44. Dr. Stavrou states that he was “asked to assume that ‘[analyze/analyzing] the snapshot’ also encompasses analysis of a VM instantiated from the snapshot.” Ex.1002 (Stavrou Decl.), ¶82; *see also* ¶83 (“Accordingly, my discussion of Ground 1 below applies an understanding of ‘analyze the snapshot’ in claims 1 and 22,¹ as well as ‘analyzing the snapshot’ in claim 18, as encompassing both direct analysis of the snapshot data and analysis of a VM instantiated from snapshot.”). Thus, Dr. Stavrou treats limitations in the challenged claims that recite “[analyze/analyzing] the snapshot” as though they “encompass[] both [1] direct analysis of the snapshot data and [2] analysis of a VM instantiated from [the] snapshot.” Ex.1002 (Stavrou Decl.), ¶¶80-83.

45. I disagree that this claim element encompasses “[2] analysis of a VM instantiated from [the] snapshot” because that is not how a person of ordinary skill in the art would understand the claim element. Dr. Stavrou does not try to justify this claim element encompassing “analysis of a VM instantiated from a snapshot,”

¹ Contrary to this passage of Dr. Stavrou’s declaration, claim 1 recites “analyzing” not “analyze.” However, for the purpose of claim construction the distinction does not matter.

but simply states that he was “asked to assume” this claim construction. Ex.1002 (Stavrou Decl.), ¶82; *see* Ex.2077 (Stavrou Tr.) 265:17-266:4, 266:13-267:15 (“I didn’t ask why. I was asked to assume.”), 267:16-268:3, 268:4-269:20.

46. A person of ordinary skill in the art would not understand the “[analyze/analyzing] the snapshot” limitations in the ’032 patent to encompass “analysis of a VM instantiated from a snapshot.” As an initial matter, there appears to be no disagreement that an instantiated VM is not a snapshot—otherwise Dr. Stavrou would not have enumerated it as separate from “direct analysis of snapshot data” in his assumed construction. Rather, an instantiated VM is a VM, and analysis of a VM instantiated from a snapshot is an analysis of a VM—not a snapshot. Once a VM is instantiated from a snapshot, e.g., when a duplicate VM is instantiated from a snapshot for subsequent monitoring, the instantiated VM is no longer a “snapshot,” and monitoring, assessing, or evaluating it is not “analyze/analyzing” a “snapshot” as claimed. This would have been readily understood by a person of ordinary skill in the art reading the ’032 patent—an instantiated VM is **not** a snapshot including because “[a] VM’s snapshot is a copy of the machine’s virtual disk (or disk file) *at a given point in time.*” Ex.1001 (’032 patent), 4:60-61. The specification also distinguishes analyzing a snapshot by a “security system 140 to detect vulnerabilities” (Ex.1001 (’032 patent), 5:20-21), as claimed, from “another embodiment” where a system “instantiate[s] a copy of the

VM” and performs “execution of the instance of the VM... [in] an isolated sandbox” (*id.* 5:55-60, 7:42-52).

47. In fact, as I discuss further below, it appears that Dr. Stavrou “assume[d] that ‘[analyze/analyzing] the snapshot’ also encompasses analysis of a VM instantiated from the snapshot,” as he was “asked” to (Ex.1002 (Stavrou Decl.), ¶83), simply to try to buttress opinions he offers that depend on disclosures from prior art references, such as Veselov, that discuss evaluating duplicate VMs, not snapshots.

VII. Overview of the Asserted References

A. Overview of Veselov

48. Veselov is titled “Security Assessment of Virtual Computing Environment Using Logical Volume Image” and discloses systems and techniques that perform a security assessment on “virtualized reproductions of computing resources” such as VMs or instances of VMs. Ex.1007 (Veselov), Title, Abstract, 3:20-23. Figure 1 illustrates an exemplary environment and components:

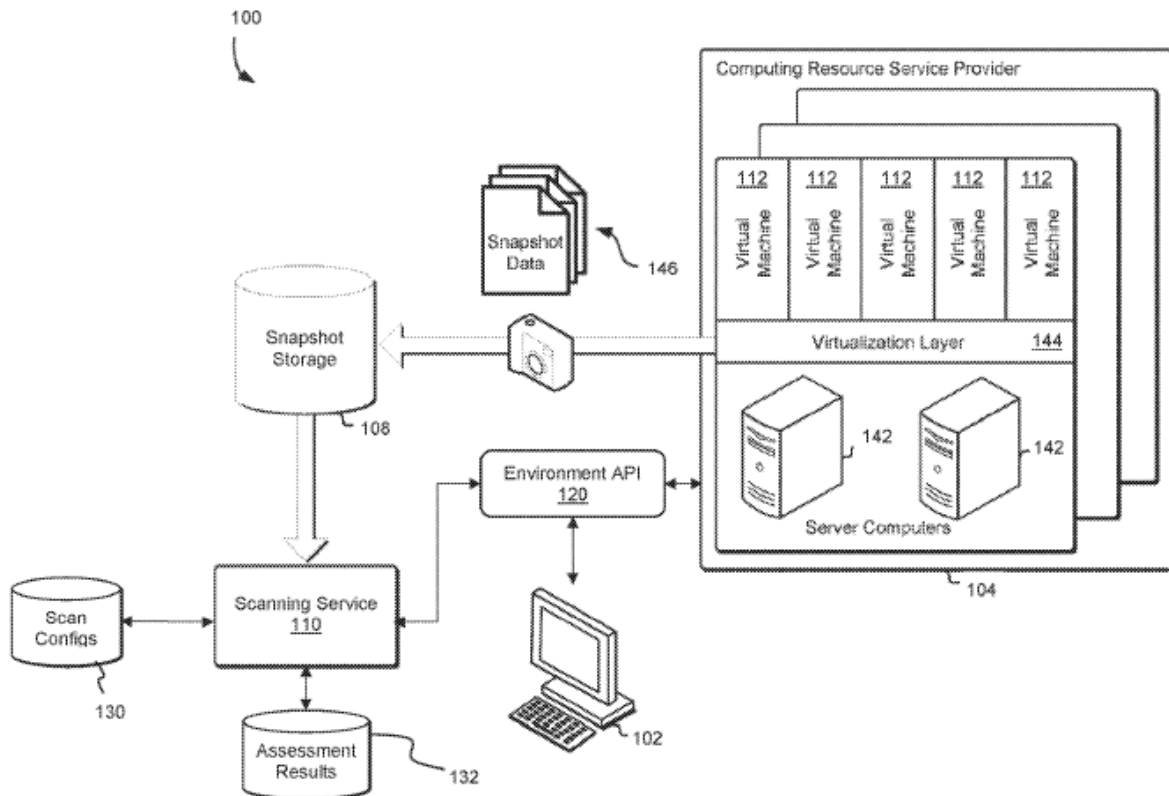


FIG. 1

49. As seen in Figure 1, Veselov discloses embodiments of a computing environment involving a computing resource service provider (shown as no. 104), in which user devices (shown as no. 102) request or manage resource allocation for server computers (shown as no. 142). Ex.1007 (Veselov), 4:19-29. These server computers are allocated as virtual computing resources. Ex.1007 (Veselov), 4:19-29. Figure 1 also depicts an API (shown as no. 120) that allows a user to connect, through, for example, a web-based application, to the computing resource service provider. Ex.1007 (Veselov), 4:29-41.

50. Veselov states that the scanning service of Figure 1 implements various disclosed steps for a security assessment. Ex.1007 (Veselov), 8:16-17. The scanning service obtains and processes snapshot data to generate assessment results. Ex.1007 (Veselov), 8:18-21. In implementing its scanning procedures, Veselov attempts to address drawbacks with resource usage during the security assessments. For example, Veselov highlights that there is a problem in the cybersecurity field where security assessments can “reduce the target computing device’s capabilities” such that the target machine “may not be capable of responding to a client as quickly as normal” or “may not be capable of providing content as quickly as normal.” Ex.1007 (Veselov), 2:56-67. Veselov also cautions that security assessments may impact machine capabilities because system components such as processors and memories are used to perform the assessment rather than to execute normal workloads. Ex.1007 (Veselov), 2:56-67. And, Veselov notes that security assessments using a target device’s computing resources “may be costly monetarily” and will require “an extended period of downtime.” Ex.1007 (Veselov), 3:8-11.

51. Veselov’s overarching goal is therefore to avoid downtime of VMs and reduce assessment complexity when performing security assessments. Throughout its disclosure, Veselov aims to preserve the activity of the original virtual machine instance. For example, Veselov provides systems that “include a

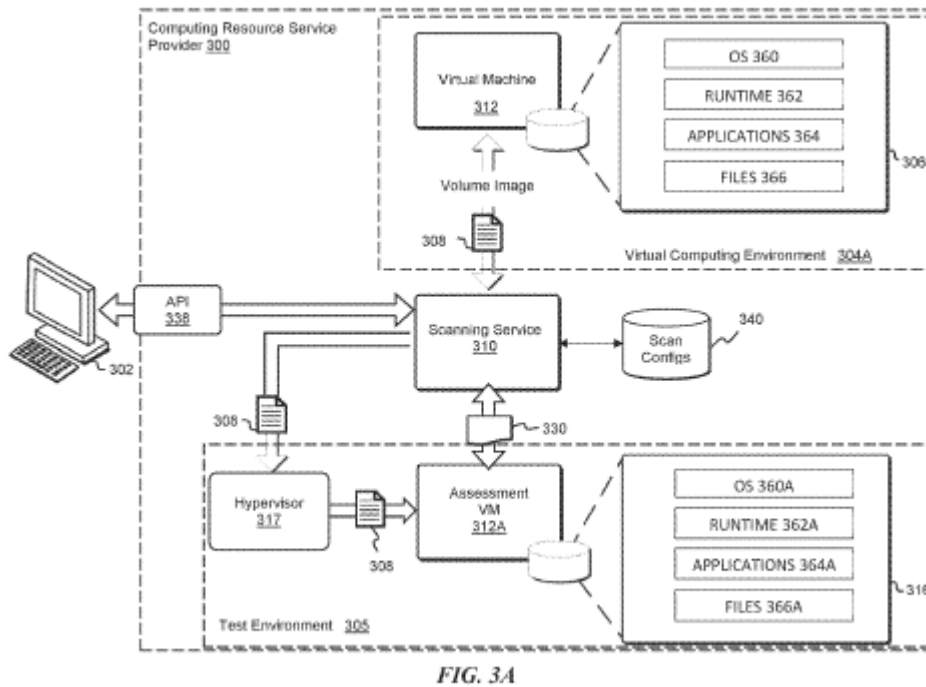
scanning service that facilitates duplication of all or a portion of the target computing resource, and then perform[] the security assessment on the duplicate computing resource”—“to avoid consuming processing time, processing power, and storage space of the target computing resource.” Ex.1007 (Veselov), Abstract; *see also* Ex.1007 (Veselov), 10:9-11 (generating a scannable volume from a snapshot includes “launching a duplicate virtual machine instance.”). And, Veselov provides that performing a robust security assessment may reduce the target computing device’s capabilities and may not be as responsive should an assessment be performed on the target machine. Ex.1007 (Veselov), 2:56-67. And Veselov cautions against “downtime.” Ex.1007 (Veselov), 3:8-11. Veselov explains that its solution allows the assessment to “be performed without impacting the customer’s virtual machine,” which can continue to “servic[e] requests from its customers or perform[] other useful work.” Ex.1007 (Veselov), 12:21-43. For the system to perform such “other useful work” Veselov must keep its VMs active during the capture of a snapshot and subsequent vulnerability analysis. Additionally, Veselov counsels solutions where the VM is configured to access one or more physical or virtual resources that are isolated in the event data dependencies need to be evaluated during an assessment. Ex.1007 (Veselov), 13:1-22, 13:33-36, 13:43-67. And the scanning service can cause “activity to occur

within the assessment virtual machine” to apply tasks in a real-time environment.

Ex.1007 (Veselov), 15:41-59, 20:57-21:13, 21:21-32.

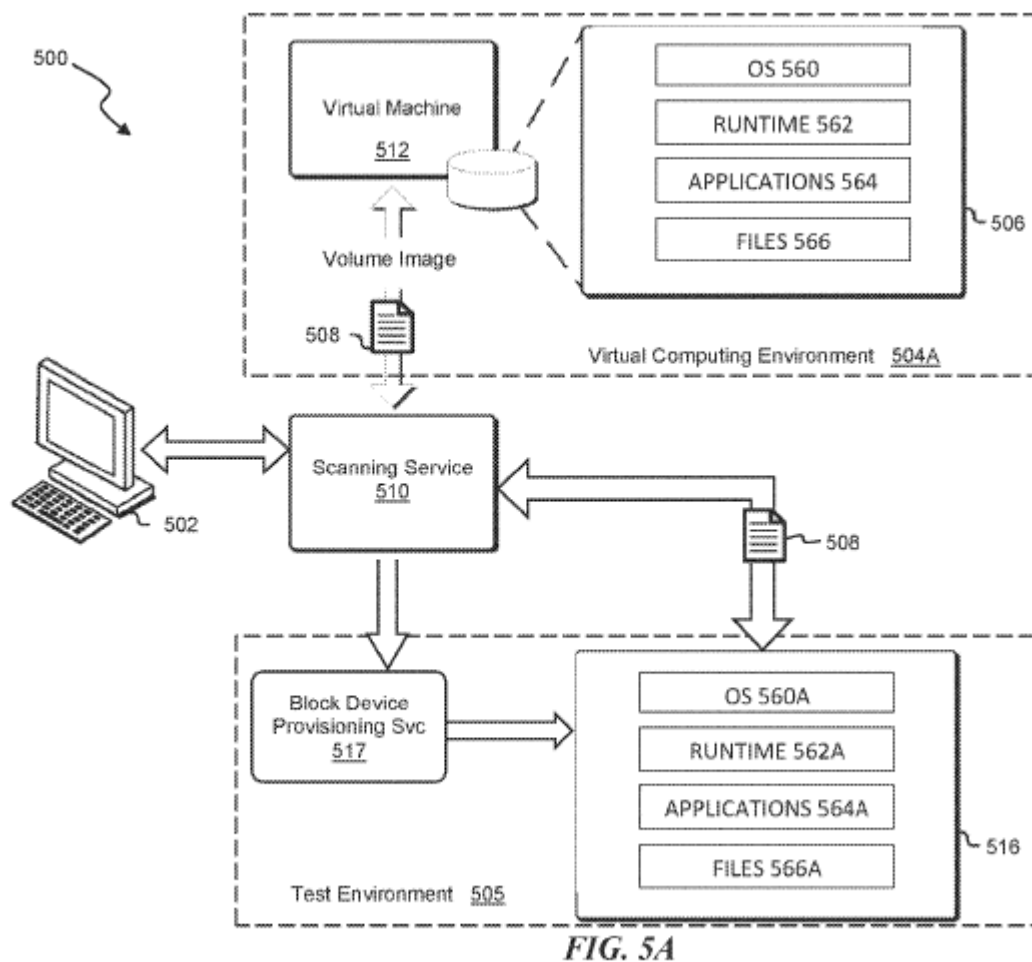
52. Veselov proposes two scanning solutions to maintain the active state of the target virtual machine. In both of Veselov’s solutions, Veselov implements snapshot technology to capture the state of the target virtual machine. *See, e.g.*, Ex.1007 (Veselov), Fig. 1, no. 146 (snapshot data). Veselov implements a virtual layer to generate or obtain access to a snapshot or snapshot data to initiate a security assessment. Ex.1007 (Veselov), 3:23-32, 6:25-49. Veselov’s snapshot includes all data necessary to “recreate the state of the computing resource” within a duplicate resource so as to preserve the active state of the machine. Ex.1007 (Veselov), 3:27-29, 3:41-34, 12:21-27.

53. In Veselov’s first solution, Veselov’s scanning service sets up and launches an active virtual machine instance (a duplicate machine for assessments). Ex.1007 (Veselov), 3:54-59. The duplicate machine has the same configuration as the original machine instance at the time of the snapshot capture. Ex.1007 (Veselov), 3:54-59. Figures 3A and 3B are informative.



54. In Figure 3A, the scanning service performs the security assessment on the assessment machine via a security agent program. Ex.1007 (Veselov), 14:13-27. The scanning service receives assessment results via the agents on the assessment machine. Ex.1007 (Veselov), 14:28-34. The assessment machine can implement processing activity in order to demonstrate real-time dynamic or static data, network connections, and communications to other computing resources. Ex.1007 (Veselov), 15:41-59. Additionally, a user or administrator may communicate instructions to the scanning service (shown inside the computing resource service provider's environment as no. 110 in Figures 3A and 3B) via the API (shown as no. 338) to configure and accomplish the security assessment. Ex.1007 (Veselov), 14:36-15:40.

55. In Veselov's second solution, Veselov provides that the scanning system may "mount" the logical volume as a hard drive with data stored thereon or may transform the snapshot into files to be read. Ex.1007 (Veselov), 3:61-66. This technique permits a security assessment on the data at the file system level. Ex.1007 (Veselov), 3:66-67. Figures 5A and 5B are informative.



56. In exemplary Figure 5A, the scanning service analyzes snapshot data in a "test environment" outside of the virtual computing environment. Ex.1007 (Veselov), 17:9-12. In both Figures 5A and 5B, the user communicates directly

with the scanning service outside the target resource computing environment. Ex.1007 (Veselov), Figs. 5A-B. 17:9-67. An API is not depicted in Figures 5A and 5B, and is not used for communication between the user and scanning service. The scanning service interfaces with VM resources through provisioning services such as a hypervisor. Ex.1007 (Veselov), Figs. 5A-B. 17:9-67.

57. In both solutions, Veselov's scanning system stores the assessment results or provides them to "an owner or administrator of the virtual machine in order to evaluate the vulnerabilities of the target resources." Ex.1007 (Veselov), 4:5-8. After storing or sharing the results, the scanning system may delete the snapshot or store it for additional analysis. Ex.1007 (Veselov), 4:8-10.

58. In both solutions, Veselov's snapshots are "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." Ex.1007 (Veselov), 4:1-4. Accordingly, as detailed in this section, Veselov's techniques purport to preserve the activity of the original virtual machine instance and focus on analyzing deployed and executing virtual machines.

B. Overview of Hufsmith

59. Hufsmith is titled "Combined Threat Score for Container Images." Ex.1078 (Hufsmith), Cover. Hufsmith discloses "a process for determining threat scores for container images or distributed applications that consider the results of

a multitude of different scanners and other factors.” Hufsmith, Abstract. Hufsmith’s “Field” is disclosed as “techniques that combine metrics of heterogeneous vulnerability scans of container images.” Hufsmith ¶2. Hufsmith is therefore directed to methods for “determining combined threat score[s] for container images,” not snapshot-based assessments of VMs. *See* Hufsmith, Title, ¶2, ¶¶4-6. Hufsmith does not discuss snapshots at all.

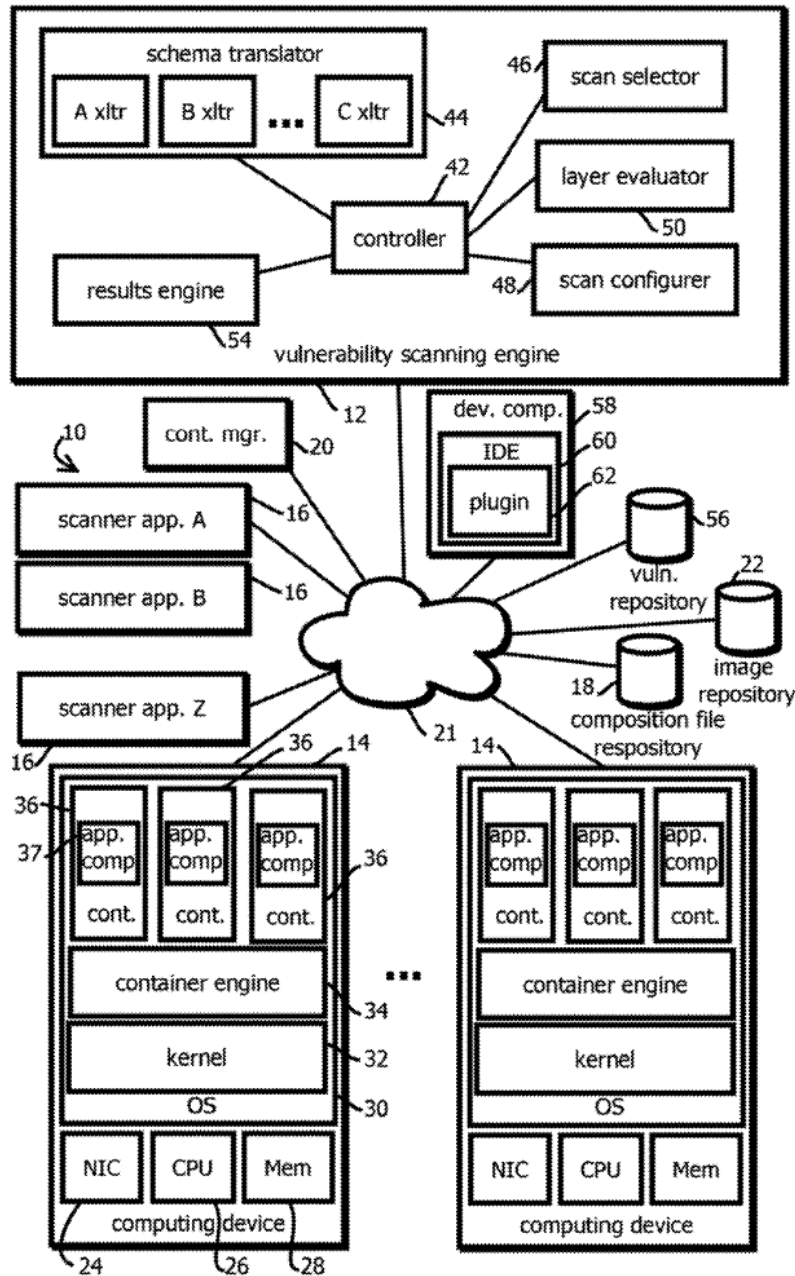
60. Hufsmith discusses problems in the field of scanning container images and teaches solutions involving its techniques for analyzing scores for container images to determine a combined threat score for the images. For example, Hufsmith discloses that components of distributed applications, called “services,” may be composed of “one or more executing bodies of code.” Ex.1078 (Hufsmith), ¶5. And, Hufsmith explains that such “bodies of code and other resources” can be difficult to secure because, for example, services arise from “diverse sets of bodies of code,” “can undergo relatively frequent version changes,” and can be “downloaded from third parties ... that may be un-trusted or accorded less trust than code built in-house.” Ex.1078 (Hufsmith), ¶5. Additionally, Hufsmith provides that in such “potentially highly complex” environments, even where potential vulnerabilities are detected, “a degree to which they pose a threat” in an execution environment with distributed code and other resources can be “difficult to quantify.” Ex.1078 (Hufsmith), ¶6. And, Hufsmith

explains that such difficulties are heightened for applications built with containers such as DockerTM containers, Open Container Initiative (OCI) containers, or with KubernetesTM because these containers “generally virtualize at the operating system level, in contrast to virtual machines” which also emulate underlying hardware. Ex.1078 (Hufsmith), ¶26. For example, containers are generally formed from multiple layers and offer OS-level virtualization with various benefits not seen in virtual machines, “including lower computational load, faster spin up, and sharing of resources,” as well as the ability to reuse “intermediate images of read-only bodies of code and other resources” “across multiple container images.” Ex.1078 (Hufsmith), ¶¶26-27. However, such features can complicate security of the distributed application due to “false positives” in the container context along with an inability to obtain holistic views into container exposure (i.e., because of the container’s layered nature). Ex.1078 (Hufsmith), ¶¶28-29. Also, Hufsmith explains the prior art lacks “multi-sourced vulnerability assessments” and “combined threat scores for container images.” Ex.1078 (Hufsmith), ¶¶28-29.

61. Accordingly, Hufsmith teaches techniques to combine metrics of heterogenous vulnerability scans of container images. Ex.1078 (Hufsmith), ¶2. Hufsmith teaches that its techniques are useful in the context of “software development related to distributed applications” (Ex.1078 (Hufsmith), ¶2) and its techniques assist software developers during development of container images,

i.e., before container instances are instantiated from a container image. Ex.1078 (Hufsmith), ¶153; *see also* ¶63 (“In some embodiments, container images are instantiated as container instances in which code of the container image is executed and functionality of the container images provided, for instance, as one of the above-described services.”).

62. Hufsmith provides an architecture to implement a scanning engine to facilitate the vulnerability scans. Ex.1078 (Hufsmith), Fig. 1A.



63. Figure 1A depicts a vulnerability scanning engine (shown as no. 12) that implements a controller (shown as no. 42), a schema translator (shown as no. 44), a scan selector (shown as no. 46), a layer evaluator (shown as no. 50), a scan configurer (shown as no. 48), and a results engine (shown as no. 54) that coordinate

to arbitrate which layers and which portions of layers will be scanned by one of various scanning applications (shown as no. 16). Ex.1078 (Hufsmith), ¶68. The vulnerability scanning engine decides which scanner application will perform the scan among a set of heterogeneous scanner applications that are each configured to apply a different type of scan to a different type of body of code and other resources such as images and files. Ex.1078 (Hufsmith), ¶68.

64. Figure 1B demonstrates the results engine (shown as no. 54) communicatively coupled (via a network shown as no. 21) with the container manager (shown as no. 20) and the container score database (shown as no. 102).

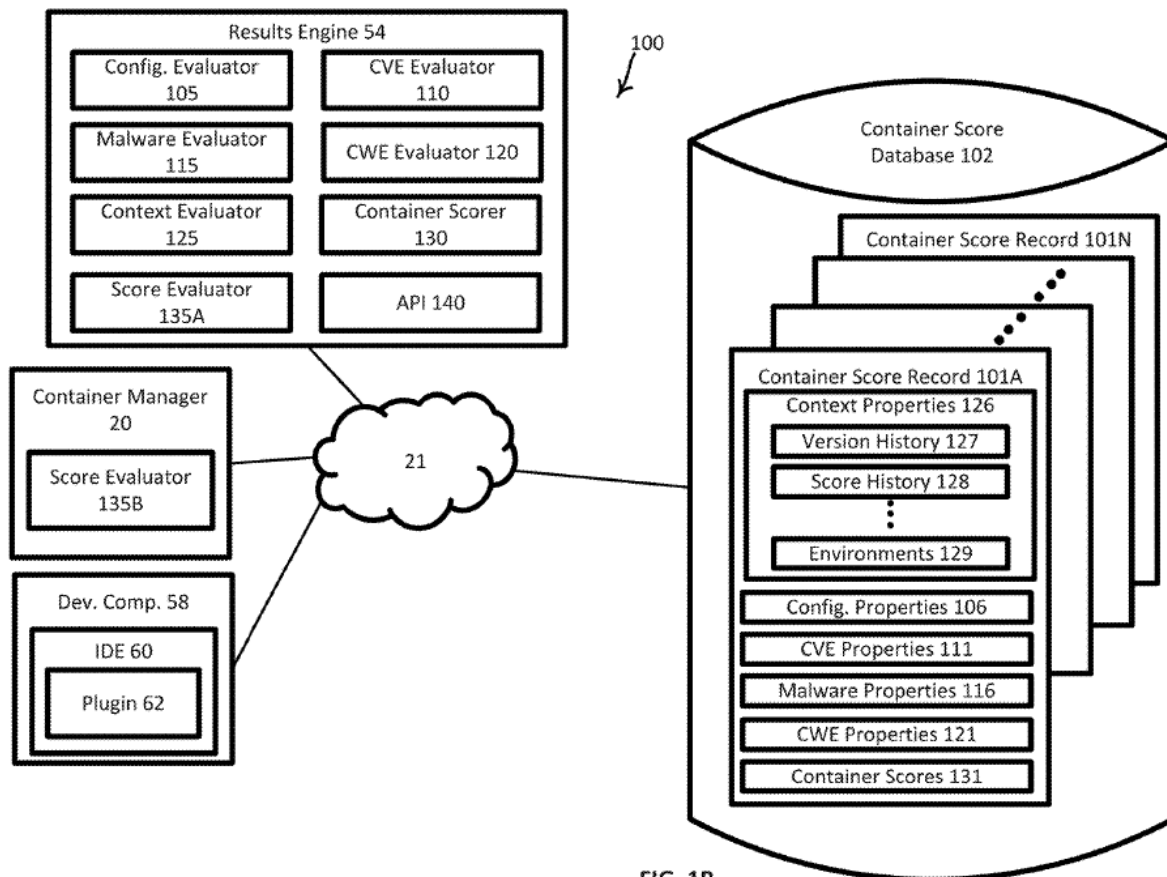


FIG. 1B

65. Hufsmith provides that the results of the vulnerability scanning engine will be translated into a unified schema of scanner results or properties, which may be stored in a database “in association with an identifier of the container image.” Ex.1078 (Hufsmith), ¶101. The database may be a container score database which comprises a number of container score records (shown as nos. 101A to 101N) that are referenced by, for example, a cryptographic hash value of the container image. Ex.1078 (Hufsmith), ¶101. In these scenarios, where the container score records correspond to container images that constitute a distributed application, a given container score record may reference one or more other container score records. Ex.1078 (Hufsmith), ¶104. These container score records “assist a developer in determining which one of a plurality of container images to utilize based on a risk associated with each container image.” Ex.1078 (Hufsmith), ¶103.

66. Figure 4, below, illustrates a process for “determin[ing] one or more score records for a distributed application having one or more container images. Ex.1078 (Hufsmith), Fig. 4, ¶¶18, 199.

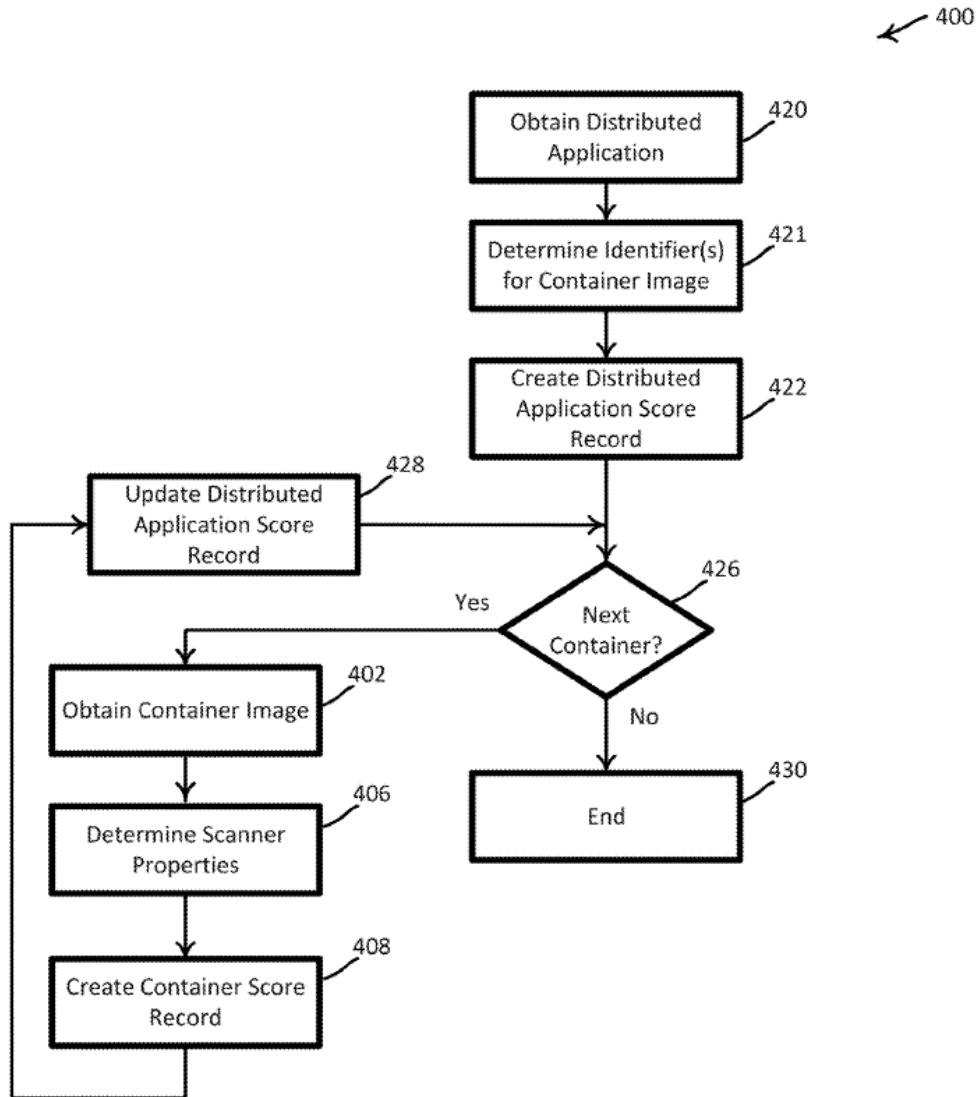


FIG. 4

67. First, Hufsmith provides that the system obtains a “distributed application” (depicted by block 420). Ex.1078 (Hufsmith), ¶201. The second step (depicted by block 421) provides that the system will identify container images

associated with the distributed application “(which may be a multi-container distributed application).” Ex.1078 (Hufsmith), ¶202.

68. Then, at the third step (depicted by block 422), the system creates a distributed application score record for the distributed application. Ex.1078 (Hufsmith), ¶¶201-202. And, when the system determines that another container image to process exists (depicted by block 426), the system will execute steps 402-408. Ex.1078 (Hufsmith), ¶¶201-202. The system therefore obtains a container image (depicted by block 402). Ex.1078 (Hufsmith), ¶200. Then, scanner properties are determined for the container image (depicted by block 406). Ex.1078 (Hufsmith), ¶200. The system then proceeds to Figure 2 (below). Ex.1078 (Hufsmith), ¶200.

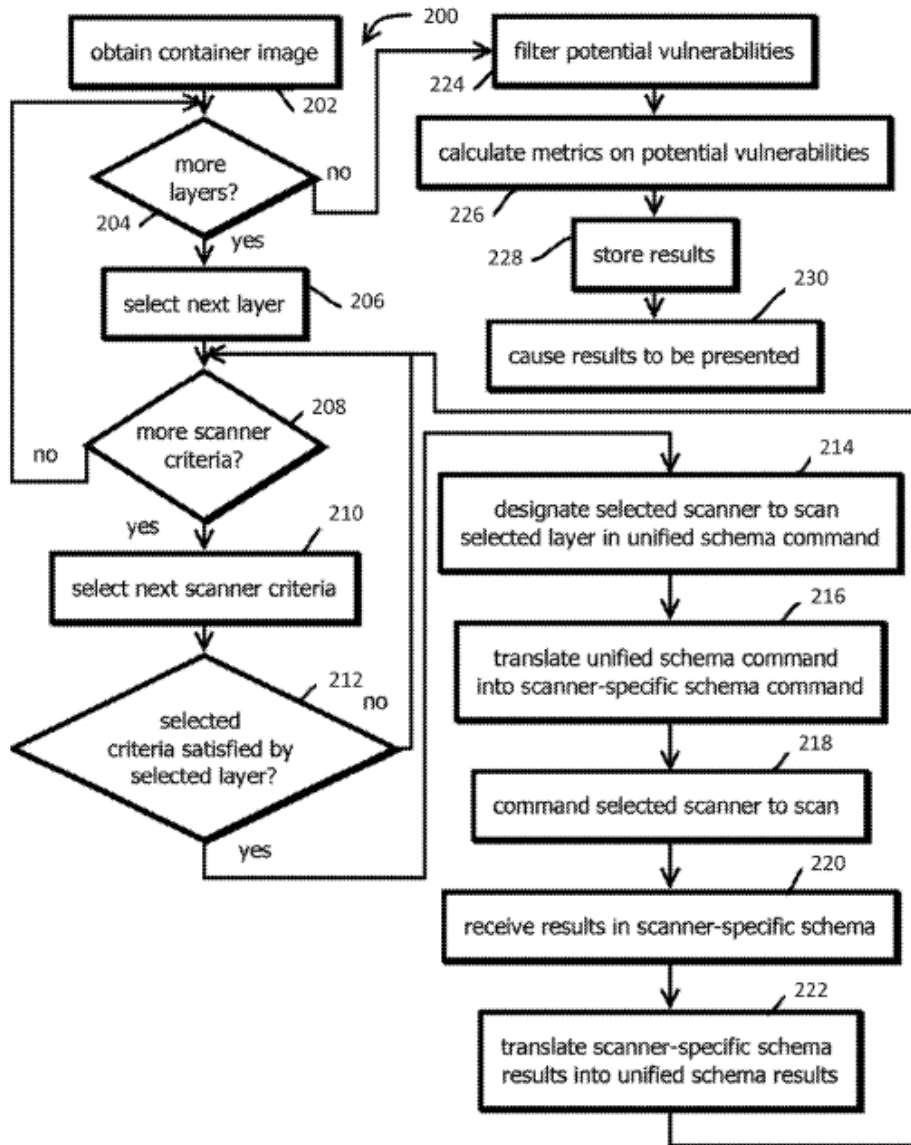


FIG. 2

69. At Figure 2, the system first obtains a container image (depicted by block 202). Ex.1078 (Hufsmith), ¶152. Then, Hufsmith provides that the system determines whether there are additional layers in the container image to be processed (depicted by block 204). Ex.1078 (Hufsmith), ¶152. For example, the

system will start with the base or top layers of a multi-layer container, and, if there are additional layers to process, then the system will continue to select a layer (depicted by block 206). Ex.1078 (Hufsmith), ¶152. Then, the system determines whether there is additional scanner criteria that should be applied to a given layer (depicted by block 208). Ex.1078 (Hufsmith), ¶152. If the system determines that there are additional criteria to apply, then an option for “next scanner criteria” may be selected (depicted by block 210). Ex.1078 (Hufsmith), ¶152. The system determines whether a given layer is satisfied by selected criteria (depicted by block 212), and if so, the scanner that corresponds to the chosen criteria will be selected to scan the given layer (depicted by block 214). Ex.1078 (Hufsmith), ¶152. Next, the system executes a command to the scanner to scan, and the results of the scanning are received and translated (depicted by blocks 216-222). Ex.1078 (Hufsmith), ¶152. Then, once there are no further layers to scan, the system will filter potential vulnerabilities and calculate metrics (depicted by blocks 224, 226). Ex.1078 (Hufsmith), ¶152. Subsequently, the system may store results and present results (depicted by blocks 228, 230). Ex.1078 (Hufsmith), ¶152. Then, the system resumes at Figure 4, where a container score record for the container image may be created and a distributed application score may be updated. Ex.1078 (Hufsmith), ¶¶200, 202. The system will continue to process container images

iteratively until it finds no additional container images to process. Ex.1078 (Hufsmith), ¶202 (describing blocks 426, 430 of Figure 4).

70. Figure 5 illustrates a process for “determin[ing] a combined threat score for a distributed application having one or more container images or a container image.” Ex.1078 (Hufsmith), ¶19, 204, Fig. 5.

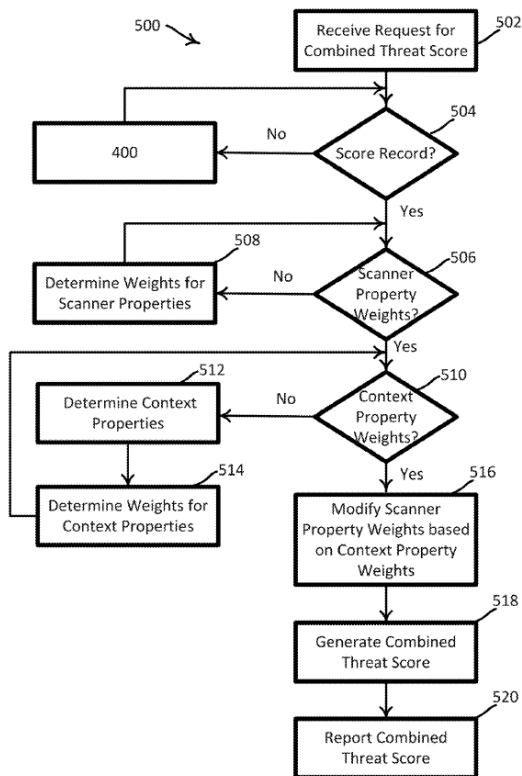


FIG. 5

71. Hufsmith’s system of Figure 5 begins by receiving a request for a combined threat score (depicted by block 502). Ex.1078 (Hufsmith), ¶205. Then, the system determines whether a “score record for a distributed application or

container image” exists, “such as a score record created by [the] process of FIG. 4.” Ex.1078 (Hufsmith), ¶205. If no score record exists, Hufsmith provides that the “process 400 of FIG. 4 may be called to create” one. Ex.1078 (Hufsmith), ¶206. Then, after the system has a score record, the record “may be examined for scanner property weights” (depicted by block 506), which may be determined and stored to the score record if none already exist (depicted by block 508). Ex.1078 (Hufsmith), ¶206. Also, a score record “may be examined for context property weights” (depicted by block 510). Ex.1078 (Hufsmith), ¶207. And, if no context property weights exist, then the system includes an additional step of “requesting context property information, such as environment information,” from which “weights are determined for the context properties at block 514.” Ex.1078 (Hufsmith), ¶207. Then, weights associated with different scanner properties are modified based on weights of corresponding context properties (depicted by block 516). Ex.1078 (Hufsmith), ¶208. And, after “each of the context property weights are applied, a combined threat score may be generated based on the modified scanner property weights” (depicted by block 518) and reported (depicted by block 520). Ex.1078 (Hufsmith), ¶¶208-209.

72. Accordingly, Hufsmith “determin[es] threat scores for container images or distributed applications that consider the results of a multitude of

different scanners and other factors such as context information” for “container images.” Ex.1078 (Hufsmith), Abstract.

C. Overview of Hutchins

73. Hutchins is titled “Offloading Operations to a Replicate Virtual Machine.” Ex.1070 (Hutchins) Cover. Hutchins discloses a “method for detecting malicious code within a first virtual machine comprising creating a snapshot of the first virtual machine and transferring the snapshot to a second machine.” Ex.1070 (Hutchins) Abstract. “A scan operation is run on the snapshot using resources of the second machine.” *Id.* “In response to detecting malicious code during the scan operation, action is taken at the first virtual machine to address the detection of the malicious code.” *Id.*

74. Figure 6 of Hutchins, reproduced below, is a flowchart showing a disclosed “process for creating and running a replicated virtual machine.” Ex.1070 (Hutchins) ¶39. In step 602, a “synchronous snapshot of the virtual machine device state is taken, excluding the main memory.” *Id.* A replica of the virtual machine is made, which includes copying the device state and configuration file corresponding to the original VM to a new, second snapshot in step 605. *Id.* ¶40. The main memory is then cloned in step 607, and then both virtual machines can run from the cloned state. *Id.* ¶¶40-41. A script can then be run on the replicated

VM to perform, e.g., data mining, backup, virus scanning, etc. *Id.* ¶44. The replicated VM is then “usually discarded.” *Id.* ¶45.

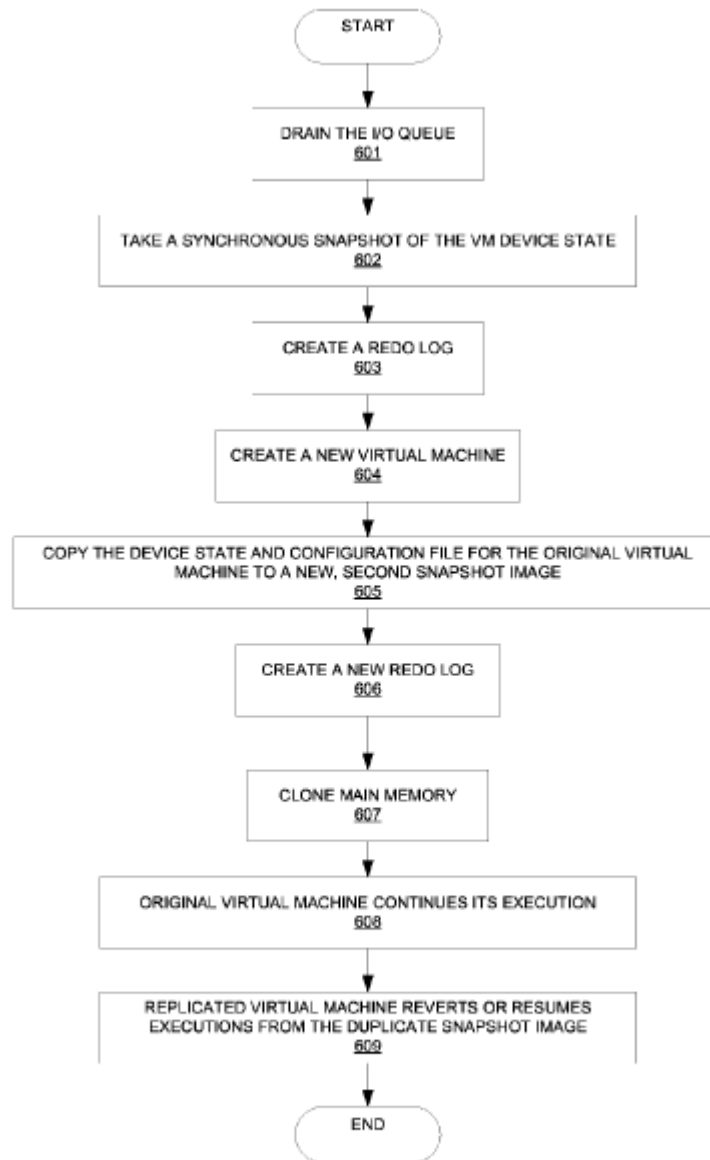


FIG. 6

75. Hutchins further discloses that if a virus is detected in the replicated VM, the original VM can be quarantined. *Id.* ¶53.

VIII. The Claims Of The '032 Patent Would Not Have Been Obvious

76. Petitioner and Dr. Stavrou challenge claims 1-25 of the '032 patent based on two grounds of purported obviousness, each of which depends on the combination of U.S. Pat. No. 11,216,563 ("Veselov," Ex.1007) and U.S. Patent App. Pub. No. 2020/0097662 ("Hufsmith," Ex.1078).

77. In my opinion, none of the claims of the '032 patent would have been obvious over Petitioner's and Dr. Stavrou's prior art, for reasons I explain below.

A. Ground 1: Claims 1-11 and 13-25 Would Not Have Been Obvious Over Veselov and Hufsmith

78. In Ground 1 of the Petition, Petitioner and Dr. Stavrou challenge claims 1-11 and 13-25 of the '032 patent based on the combination of Veselov and Hufsmith. *See* Ex.1002, Stavrou Decl. §XII.

79. In my opinion, Petitioner and Dr. Stavrou fail to show that any of claims 1-11 and 13-25 of the '032 patent would have been obvious over the combination of Veselov and Hufsmith.

1. A Person of Ordinary Skill in the Art Would Not Have Been Motivated to Combine Veselov and Hufsmith and Would Not Have Had a Reasonable Expectation of Success in Doing So

80. Dr. Stavrou asserts that a person of ordinary skill in the art would have been motivated to make the combination of Veselov and Hufsmith on which both of Petitioner's grounds depend. *See* Ex.1002 (Stavrou Decl.), ¶102. I disagree.

81. To a person of ordinary skill in the art, Veselov and Hufsmith have unrelated goals and contradictory teachings, and a person of ordinary skill in the art with Veselov in hand would not have been motivated to combine the teachings of Hufsmith as Petitioner and Dr. Stavrou assert for their challenges to the '032 patent claims. In view of these references' distinctions, which I discuss in more detail below, it appears to me that Petitioner and Dr. Stavrou are working backwards from the '032 patent's claims, looking for disclosures in disparate references, without meaningfully considering whether the references they cite would, in fact, be combined in the first place. As noted above, I understand that obviousness arguments based on this type of hindsight are improper.

82. First, Dr. Stavrou asserts that "Hufsmith's matching and prioritization techniques were a natural fit and directly applicable to Veselov's teachings regarding snapshot-based security assessments." Ex.1002 (Stavrou Decl.), ¶103. I disagree. Veselov discloses analysis of a "logical volume image" including, e.g., with snapshots. As I discussed above, Hufsmith does not discuss snapshots. *See* Section VII.B. Nor does Hufsmith discuss any type of analysis specific to a particular "volume." Hufsmith relates to a different solution than any problem in Veselov. Specifically, Hufsmith is directed to "techniques [for] combin[ing] metrics of heterogeneous vulnerability scans of container images," by, e.g., scanning the source code of container images, to provide combined threat scores

for container images and multi-container distributed applications. Hufsmith ¶¶2, 184. Dr. Stavrou characterizes Hufsmith as providing a “more detailed discussion of common techniques for detecting and prioritizing vulnerabilities/alerts” (Ex.1002 (Stavrou Decl.) ¶103), but the alleged prioritization of alerts that Dr. Stavrou points to are unrelated to anything in Veselov. Hufsmith applies weights to metrics, and annotations to source code, neither of which “natural[ly] fit” with Veselov. Veselov does not disclose metrics, and does not have access to such image source code for annotation.

83. Dr. Stavrou attempts to minimize the differences between Hufsmith’s “container images” and “virtual machine images” and Veselov’s snapshots. Ex.1002 (Stavrou Decl.) ¶101. Although Veselov refers to its snapshots as “***volume*** images,” these snapshots are *not* the same as the “container images” and “virtual machine images” used in Hufsmith.² Dr. Stavrou clarified at his

² Additionally, the disclosures of Veselov relating to assessing a duplicate/assessment VM are inapplicable to the claims because, as I discussed above, analysis of a duplicate VM instantiated from a snapshot is **not** “analyzing the snapshot” as recited in limitations 1.3, 18.3, and 22.3. *See, e.g.*, Veselov 10:37-16:14 (discussing Figs. 3A/3B embodiments), 16:15-55 (describing “example method 400 of performing a security assessment of a target virtual machine (VM)

deposition, “snapshots and images are different.” Ex.2077 (Stavrou Tr.) 272:7-13.

I agree. A VM’s snapshot is, as the ’032 patent states, “a copy of the machine’s virtual disk (or disk file) at a given point in time.” Ex.1001 (’032 patent), 4:60-61.

A VM image, in contrast, is a data file that “include[s] an installed and configured guest operating system and applications” and is used for running a virtual machine.

Ex.2018 [Abraham Silberschatz et al., *Operating System Concepts* (9th ed. 2013)], p.716; *see also, e.g.*, Ex.2005, [US Pat. App. Pub. No. 2008/0320583], ¶¶160-63

(distinguishing a “disk image file,” which is “a virtual disk file that stores the contents of the virtual machine’s hard disk drive,” and a “snapshot file,” which is

“a file that stores the running state of the VM at a particular moment in time, usually in terms of deltas with respect to a baseline state.”); Ex.2020 [Huawei,

Snapshots vs. Images] (explaining that images “[c]an’t be created for running instances” whereas snapshots “record[] the state and data of the virtual machine at

the moment the snapshot was taken”); *Compare* Ex.2006 [Red Hat Virtualization 4.3 – *Virtual Machine Snapshots*] (“Snapshot are a storage function that allows an

administrator to create a restore point of a virtual machine’s operating system,

in ... FIGS. 3A-B”), 20:57-22:6, Figs. 3A-3B, 4; *see* Ex.1002 (Stavrou Decl.) ¶139

(“Veselov discloses two general ways to perform the analysis” and relying on launching a duplicate VM instance for analysis).

applications, and data at a certain point in time.”) and Ex.2007 [Birk, *Technical Challenges*], 4 (“Snapshots, also referred to as forensic image, of virtual machines provide a powerful tool with which a virtual machine can be cloned by one click including also the running system's memory. This leads to the main benefit that systems hosting crucial business processes do not have to be shutdown for performing a forensic analysis.”) with Ex.2025 [Red Hat Open Stack], p. 5 (“A virtual machine image is a file that contains a virtual disk which has a bootable operating system installed on it.”). A container image, as referred to in Hufsmith, is even farther afield from Veselov’s “volume images” or snapshots, because of the additional differences between containers and VMs as discussed below. *See, e.g.,* Ex.1078 (Hufsmith), ¶27; Ex.2075, Wiz Academy, *Containers vs. VMs: What’s the difference?* (April 8, 2024) (“In a nutshell, containers and virtual machines (VMs) are two inherently different approaches to packaging and deploying applications/services in isolated environments.”).

84. Additionally, a person of ordinary skill in the art would have known that snapshots and images are used to address different classes of vulnerabilities. For example, a point-in-time snapshot of a computer system disk is known to allow capturing and cleaning of files infected with malware at the time when the snapshot was taken. Ex.2023 [U.S. Pat. No. 8,220,053 B1 (“Sun”)], 5:49-58. In contrast, because VM images are used to instantiate VMs, scanning, detecting, and

removing of malware from offline VM images would prevent the spread of malwares. Ex.2024 [Wei, *Managing Security of Virtual Machine Images*], Sec. 2.3 (“Anecdotal evidence shows that, if dormant VM images are not managed (e.g., scanned for worms), a virtual environment may never converge to a steady state, because worm-carrying VM images can sporadically run, infect other machines, and disappear before they can be detected”).

85. Dr. Stavrou also recognizes that snapshots and images are not the same thing. *See, e.g.*, Ex.1002 (Stavrou Decl.) ¶37 (“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.”) (emphasis added) (quoting Ex.1018 at 2-6).

86. A person of ordinary skill in the art thus would have understood that Hufsmith and Veselov do **not** “perform their assessment on analogous data,” as Dr. Stavrou asserts. Ex.1002 (Stavrou Decl.), ¶101. Instead, a person of ordinary skill in the art would have recognized that Hufsmith’s teachings were not compatible with those of Veselov, and would not have sought to combine the teachings of the references as Dr. Stavrou asserts. Additionally, Dr. Stavrou relies on a series of tertiary references to assert his rationale for combining Veselov and Hufsmith. *See* Stavrou (Ex.1002) ¶101 (citing Ex.1014 (Klemperer) 24-25; Ex.1025 (NIST), 32; Ex.1034 (Tang), 5643; Ex.1044 (Giakouminakis) 6:9-18,

11:51-67; Ex.1045 (Canzanese) ¶7; Ex.1046 (Crabtree) 11:52-12:14; Ex.1048 (Price), ¶43, Ex.1063 (Hibbert) ¶¶13, 187-189, 207-11; Ex.1076; Ex.1076 (Milner), 2:24-67; Ex.1077 (Khalid), 2:45-51, 8:1-30). But none of these disclosures explain why persons of ordinary skill in the art would make the purported combination of Veselov and Hufsmith. Klemperer, NIST, Tang, Giakouminakis, Canzanese, Crabtree, Hibbert, and Khalid do not even mention containers or container images and therefore provide no support for why a person of ordinary skill in the art would consider Hufsmith when analyzing the techniques of Veselov. Price and Milner disclose offline images and container files, respectively, but are lacking in any mention of snapshots and fail to support a motivation to combine with Veselov's technology. Further, Hibbert is directed to agent-based collection of vulnerability data and is not relevant to Veselov's purported non-intrusive techniques. And the NIST Guide broadly discloses prioritizing incidents based on business factors and does not demonstrate that the features of Hufsmith – or, much less, the '032 patent's challenged claims – were well-known or would be insightful to a person of ordinary skill in the art considering Veselov's snapshot scanning technology. At base, Dr. Stavrou's other cited references similarly do not motivate a person of ordinary skill in the art to combine Veselov's and Hufsmith's disclosures and do not demonstrate any rationale for the purported combination of Hufsmith's analysis of container images

and Veselov's active, snapshot-based scanning of virtual machines. Further, Dr. Stavrou's reliance on Basavapatna and Hibbert (Ex.1002 (Stavrou Decl.) ¶101) do not motivate the combination to render the claims obvious. Basavapatna merely discloses that "the user should be alerted" when an "aggregate risk metric for any asset rises above a specific threshold." Ex.1008 (Basavapatna) ¶¶138-140. In other words, the alerts are *not* prioritized at all, but rather all alerts arise from a metric surpassing the same "specific threshold." Hibbert teaches the exact same approach, which differs from Dr. Stavrou's characterization and from the claimed limitation. Ex.1063 (Hibbert) ¶¶13, 189 ("generate an alert... based on the relevant risk threshold (e.g., determined that a risk value exceeds a relevant risk threshold).").

87. Hufsmith's security assessments of container images are different than Veselov's snapshot-based security assessments of virtual machines and a person of ordinary skill in the art would not have combined them. For example, as Wiz acknowledges on a page on Wiz's website, "[i]n a nutshell, containers and virtual machines (VMs) are two inherently different approaches to packaging and deploying applications/services in isolated environments." Ex.2075, Wiz Academy, *Containers vs. VMs: What's the difference?* (April 8, 2024).

88. Dr. Stavrou then asserts that a person of ordinary skill in the art would have been motivated to combine Veselov and Hufsmith "to obtain known benefits

of prioritized alerts” and that “rapid identification” of which asserts were at highest risk was well-known. Ex.1002 ¶104. But Dr. Stavrou does not explain how any concern of Hufsmith applies to Veselov. As discussed above, Hufsmith analyzes “container images in complex environments.” *See* Section VII.B. Such containers, according to Wiz’s webpage, were known to be “less secure” than virtual machines “because there is no network isolation” and “containers [are] more prone to vulnerabilities.” Ex.2075, Wiz Academy, *Containers vs. VMs: What’s the difference?* (April 8, 2024) at 6. Dr. Stavrou does not address how the same concerns apply to Veselov, which assesses particular volume images and provides assessment results to an “owner or administrator of the virtual machine.” Ex.1007 (Veselov) 4:5-8. Nor does Veselov provide any indication that alert fatigue is a concern in its system.

89. Hufsmith focuses on environments with multiple scanner applications that can each report their own alerts. Hufsmith ¶71, *see also id.* ¶¶68 (“heterogeneous set of scanner applications”), 73 (“each scanner application may have different criteria”), 79-85 (describing plurality of “scanner applications”), 85-86 (describing each scanner returning results). That is not the case in Veselov. There is no reason to implement the score-based system in Veselov without these multiple scanners, which would add significant complexity to Veselov. In fact, Veselov leads a person of ordinary skill in the art away from modifications that

add such complexity. Veselov teaches that its disclosed system “overcome[s] the drawbacks of existing security systems” including “complexity of installation and monitoring of the assessment system.” Ex.1007 (Veselov) 2:42-46. And Hufsmith’s disclosures relating to source code annotations in a development environment teach applications in a pre-deployment scenario where products are not being assessed in a live setting, unlike Veselov. Ex.1078 (Hufsmith) ¶¶153-156, 180-183.

90. To the extent that Hufsmith even discloses prioritized alerts in this scenario—which I don’t agree with (see Section VIII.A.2.b)—a person of ordinary skill in the art would have understood such prioritized alerts to be only applicable in the pre-deployment development context, i.e., during coding of a container, so that the container has as few major vulnerabilities as possible by the time of deployment. Once the container is deployed, it would be expected to be substantially free of vulnerabilities if one were to follow Hufsmith’s techniques. So, the concerns detailed in Hufsmith, where containers are actively being developed to be free of vulnerabilities, would not be the same in a live setting, where the container image is deployed and executing and source code is not available to change. Further, Veselov’s goal of keeping the VMs active during assessment is essentially moot in a pre-deployment scenario where a VM is not actively deployed. *See* Section VII.A. Nor would a person of ordinary skill in the

art have had a reasonable expectation of success in applying Hufsmith's teachings to Veselov, including because, as discussed above, Veselov's system does not have access to source code to provide such annotations.

91. For the foregoing reasons, it is my opinion that Petitioner and Dr. Stavrou have failed to show—because they cannot show—that a person of ordinary skill in the art would have made the Veselov-Hufsmith combination on which every ground in the Petition depends. A person of ordinary skill in the art would not have made and would not have reasonably expected to succeed in making, the Veselov-Hufsmith combination.

2. Even If Combined, Veselov and Hufsmith Do Not Disclose or Suggest Multiple Limitations in the Independent Claims

92. Even if a person of ordinary skill in the art would have made the Veselov-Hufsmith combination that Dr. Stavrou and Petitioner urge, it is my opinion that none of the claims of the '032 patent would have been obvious over that combination (or other combinations depending on this combination) at least because Veselov and Hufsmith do not disclose or suggest multiple limitations in the claims.

a. The Combination Does Not Disclose Or Render Obvious Determining Whether Installed

Applications Are Used (Limitations 1.5, 18.5, and 22.5)³

93. Limitations 1.5, 18.5, and 22.5 recite “[determining/determine] whether the matching installed applications are used by the protected virtual cloud asset.” Dr. Stavrou relies only on Hufsmith for disclosure of these limitations. Ex.1002 (Stavrou Decl.), ¶¶150-155.

94. Dr. Stavrou asserts that Hufsmith discloses these limitations “in several ways.” Ex.1002 (Stavrou Decl.) ¶150. I disagree. It is my opinion that Hufsmith does not teach or suggest these limitations.

95. First, Dr. Stavrou asserts that “Hufsmith teaches adjusting previously assigned threat levels based on whether the associated file/code is dormant or active” and asserts that this teaching discloses the claim limitation Ex.1002 (Stavrou Decl.) ¶151. He cites to Hufsmith’s “context evaluator” that “evaluate[s] context properties” such as “environments.” Ex.1078 (Hufsmith) ¶135; Ex.1002 (Stavrou Decl.) ¶151. Hufsmith explains that execution environment records can “indicate which portions of a container image are dormant and which are active in a given use case.” Ex.1078 (Hufsmith) ¶140. For instance, “a container image

³ In this declaration, I use the claim limitation labels that Dr. Stavrou and Petitioner used. A listing of the claims of the '032 patent, with those labels, is presented at the beginning of the Petition, in a “Listing of Challenged Claims.” Pet. at iii-viii.

may include a subroutine that... is never invoked.” *Id.* This can be found by analyzing a “call graph of [] the larger distributed application.” *Id.*

96. Dr. Stavrou asserts that “this dormancy analysis of ‘code’ (or ‘portions’ of the image)” would have been understood by a person of ordinary skill in the art to “include, or at least suggest, assessing the dormancy/activity of installed applications.” Ex.1002 (Stavrou Decl.) ¶152. To support this assertion, Dr. Stavrou appears to equate portions or subroutines of code of the “container image” or “distributed application” with the installed applications of the claim.

97. But determining whether a portion of code, or a subroutine of a distributed application, is active does not determine whether an installed application is used. Distributed applications, and components thereof (e.g., code of the application or a subroutine of an application), are different concepts. Hufsmith explains that “[d]istributed applications are computer applications implemented across multiple network hosts.” Ex.1078 (Hufsmith) ¶3. The various hosts “often each execute at least a part of the application’s code and cooperate to provide the functionality of the application.” *Id.* In particular, distributed applications have “components replicated on multiple computers” or “different components on different computers that execute different aspects of the application.” *Id.* The different components “can be characterized as services”

including “an application component (e.g., one or more executing bodies of code) that communicates via a network... with another application component.” *Id.* ¶4.

98. But determining whether some portion or subroutine of an application will be called in a call graph does not teach or render obvious determining whether an installed application is used at all. Rather, such information presumes the application *is* used and then analyzes only whether, when used, specific portions will be called or remain dormant. Additionally, the '032 patent distinguishes between an application and its components (Ex.1002 ('032 patent) 5:40-43 (“sub-modules of an application”), 5:49-54 (“applications and modules”)), and differentiates between determining use of an “application” and determining “modules executed in the protected virtual cloud asset.” *Compare, e.g., '032 Patent* cl. 1 (“determining whether the *matching installed applications* are used”), *with* cl. 10 (“deduce applications and *modules executed in the protected virtual cloud asset*”); *see also* 6:7-12 (“deduction of running applications and modules”). Stated differently, while a subroutine or code of an application may never be invoked, that does not mean that the application itself is not executed, contrary to Dr. Stavrou’s assertions.

99. For the same reasons, Dr. Stavrou’s second argument also does not show that this limitation would have been obvious to a person of ordinary skill in the art. Dr. Stavrou asserts that “Hufsmith teaches or suggests making the recited

use determination as part of the initial calculation of individual weights.” Ex.1002 (Stavrou Decl.) ¶153. Dr. Stavrou further asserts that “[d]iscussing the calculation of individual weights for malware—which includes installed applications that are infected with malware—the malware evaluator considers ‘whether the file or code is executed (i.e., whether the matching installed application is used by the protected virtual cloud asset).” Ex.1002 (Stavrou Decl.) ¶153. But again, this calculation involves evaluating whether a “file or code” is executed, and such “file or code” is simply a portion of a larger “application package in a container image” even if it has malware. Ex.1078 (Hufsmith) ¶¶121, 123. Thus, in both passages Dr. Stavrou cites for disclosure of this limitation, Hufsmith only determines whether a particular component of an application (file, subroutine, etc.) is executed, but does not teach or suggest determining whether an installed application is used.

100. In fact, it would make little sense for Hufsmith to disclose this limitation. As described above, Hufsmith teaches that its container images run portions of a distributed application. Ex.1078 (Hufsmith) ¶¶3-5, 26-27. There is no disclosure of any sort of installed application apart from the distributed application. This is why Hufsmith explains that calculating a “distributed application score” involves calculating a “container score” as a *subcomponent* of the distributed application score. Ex.1078 (Hufsmith) ¶202 (disclosing updating the distributed application score after the container score is updated). Further,

Hufsmith's embodiments relate to pre-deployment scanning for developers—an environment that exists before protected virtual cloud assets could use an installed application. Veselov discloses techniques that are meant to assist software developers during development of a container image. In the paragraphs Dr. Stavrou cites, Hufsmith describes ways to “annotate code specifying container images” for developers actively working in an “integrated development environment” (or “IDE”), e.g., to reduce the “cognitive load of developing container images.” See Ex.1078 (Hufsmith), ¶¶153-197 (describing Hufsmith's “IDE”). These actions occur before a container is deployed, and are not similar to the assessments contemplated by Veselov or the '032 patent. See, e.g., Ex.1078 (Hufsmith) ¶¶108 (“the combined threat scores may be considered *during application or container development and deployment*”), 158 (emphasizing developers “need to have deep insight into vulnerabilities that *will be introduced* by them into the container that is *going to be created and deployed* with their services”), cl. 1 (“execution environment *in which* the container *is deployable* in a use-case”). Thus, determining whether a file or code (a component of an application) is never called, in the context of a pre-deployment environment where applications are not installed or executing, does not teach determining whether an installed application is used by a protected virtual cloud asset as claimed. To persons of ordinary skill in the art, Hufsmith's IDE disclosures are vastly different

from the security assessment systems and methods disclosed by Veselov (or the '032 patent).

b. The Combination Does Not Render Obvious Reporting Detected Vulnerabilities As Alerts Prioritized By Use (Limitations 1.7, 18.7, 22.7)

101. Limitations 1.7, 18.7, and 22.7 recite, “reporting the determined [plurality of] potential cyber vulnerabilities, as prioritized alerts according to the use determinations.”

102. I discussed in the previous section that Dr. Stavrou’s proposed combination does not disclose or render obvious the determination that certain applications are used in limitations 1.5, 18.5, and 22.5. That determination corresponds to the “use determinations” of limitations 1.7, 18.7, and 22.7, and therefore these limitations are not disclosed for the same reasons.

103. Additionally, Veselov does not mention alerts. Veselov only discloses that its “scanning service 110 may be configured to perform security assessments and produce assessment results based at least in part on snapshot data.” Ex.1007 (Veselov), 5:9-13. Veselov also provides that the scanning service may “process the snapshot data” and “generate one or more assessment results.” Ex.1007 (Veselov), 8:19-21, 18:53-56 (similar).

104. Dr. Stavrou, however, incorrectly equates communicating “assessment results” with “reporting... vulnerabilities as alerts.” Ex.1002 (Stavrou

Decl.), ¶161. But a person of ordinary skill in the art would have understood that communicating results does not require or suggest reporting detected vulnerabilities “as alerts,” as claimed. *See, e.g.,* Ex.1001 (’032 patent), 7:53-54 (“[T]he detected cyber threats and/or vulnerabilities are reported, *for example, as alerts*”) (italics added), cls. 1, 18, 22 (“[report/reporting] the determined potential cyber vulnerabilities, *as prioritized alerts...*”). Veselov’s disclosure of simply “communicating” results is not enough to constitute an alert. For example, Veselov’s “assessment results” include “findings” resulting from “applying the rules of the assessment” that do not suggest the urgency implied by an alert. Ex.1007 (Veselov) 10:18-22. Rather, simply “communicating” results in Veselov suggests a standard report. Persons of ordinary skill in the art would have understood that alerts and reports are different and can involve different system components, capabilities, and purposes. *Compare* Ex.2010 [U.S. Pat. No. 9,467,465 (“Hibbert”)], 20:34-44 (“The **alert** module 516 may **generate an alert** based on the assessment. If one or more applications or files identified within the records of the assessment request are classified as vulnerable or highly suspicious, the alert module 516 may generate an alert. The alert may flag the vulnerability, potentially identify the exploit, indicate the degree of danger, or provide any other information. The alert may be provided in any manner. For example, the alert module 516 may provide one or more alerts via email, SMS text message, MMS,

web page, intranet alert, extranet alert, or any other way.”) *with id.*, 20:12-21 (“The **report** module 514 may **generate any kind of report** indicating the results of one or more assessments. Information contained within the report may include the digital device, one or more users of the digital device, applications and files identified by the records which may be safe, vulnerable, suspicious, or unknown, or any other kind of information. The report module 512 may provide the report to the digital device associates with the assessment, the security administrator system 220, a network administrator, or any digital device(s) or individual(s).”).⁴

⁴ See also, e.g., Ex.2011 [US Pat. No. 10,298,599 (“Zhang”)], 9:50-67 (providing that a system may “**report the results without** causing the browser to notify, request permission from, or otherwise **alert**, a user using the browser” because “[p]resenting an alert [] interrupts the user’s experience with the web page”); Ex.2012 [US Pat No. 9,594,912 (“Thioux”)], 10:41-48 (illustrating a specific “alert generation logic” as a subset of “reporting logic” to “generate an alert for the client device”); Ex.1023 (ITL Bulletin), at 4 (“Agencies should use CVE in their internal reports of vulnerability scans, notifications to system owners of observed vulnerabilities, and alerts identifying the vulnerabilities targeted by active exploits.”); Ex.1017 (US Pat. No. 11,106,784), 13:12-19 (“generate a report to be

105. Alerts serve to notify stakeholders of situations that require attention, and not all reported results are alerts. Dr. Stavrou agreed in his deposition that “an alert is different than a report.” Ex.2077 (Stavrou Tr.) 286:9-287:3. Dr. Stavrou’s opinions about Veselov for these limitations, however, contradict how a person of ordinary skill in the art would understand limitations 1.7, 18.7, and 22.7, as those limitations do not simply recite “report[ing] the determined cyber vulnerabilities.” Ex.1002 ¶161. To the contrary, they recite that the determined cyber vulnerabilities be reported “*as alerts*.” ’032 patent, cls. 1, 18, 22.

106. Dr. Stavrou acknowledges that “Veselov does not expressly describe reporting the vulnerabilities as ‘prioritized alerts according to the use determinations.’” Ex.1002 (Stavrou Decl.) ¶162. He asserts, however, that “Hufsmith teaches this feature.” *Id.* Specifically, Dr. Stavrou cites to his discussion of limitation 1.5 and asserts that “weights” are reported by “Hufsmith’s evaluators,” teaching “or at least suggest[ing]” that they are “prioritized based on the use determinations.” *Id.* But I disagree for the reasons discussed in the previous section.

shown in the report interface”), 13:26-38 (“transmitting an alert (e.g., to a system admin)”).

107. Dr. Stavrou further asserts that, in Hufsmith, “reports of the vulnerability assessments can be ‘comprehensive’ and convey an overall score for the resource as well as information about individual vulnerabilities.” Ex.1002 (Stavrou Decl.) ¶162. Dr. Stavrou also cites “resulting metrics” in Hufsmith that may be “presented with user selectable links through to descriptions of the potential vulnerabilities upon which those metrics are based.” Ex.1002 (Stavrou Decl.) ¶162 (quoting Ex.1078 (Hufsmith) ¶96). From these disclosures, Dr. Stavrou asserts that “[a] POSA therefore would have understood that Hufsmith’s comprehensive reporting includes, or at least suggests, reporting the weights/priorities associated with the identified vulnerabilities.” Ex.1002 (Stavrou Decl.) ¶162.

108. But providing an “overall score for the resource”—i.e., for the entire container image—does not teach or suggest reporting vulnerabilities found within a virtual cloud asset as prioritized alerts. Further, these weights do not indicate the severity of the risk associated with a given security issue. Hufsmith’s weights are applied to evaluate “context properties” “identified within a container score record associated with a container image.” Ex.1078 (Hufsmith), ¶135; *see also* Ex.1078 (Hufsmith), ¶¶135-143. The weights are used to calculate a weighted sum of vulnerabilities and therefore are used to calculate a type of “count of the number of potential vulnerabilities detected.” Ex.1078 (Hufsmith) ¶90. Further, the

weights are not reported as alerts, nor are they “prioritized...according to the use determinations” as claimed. In paragraph 120, cited by Dr. Stavrou, the weights relate to metrics and values, not to a detected vulnerability itself. *See* Ex.1078 (Hufsmith) ¶112 (“the configuration evaluator 105 determines a weight for each metric or value of a metric based on whether the configuration properties implicate that the container is more, less, or neutrally susceptible to that metric or specific value associated with that metric.”).

109. Dr. Stavrou cites to Ex.1075 (Mohanty) without explanation. Ex.1002 (Stavrou Decl.) ¶162 (quoting Ex.1075 (Mohanty) 9:3-14). The quoted passage says nothing about prioritizing alerts “according to the use determinations.” *See id.* And Dr. Stavrou does not explain why a person of ordinary skill in the art would have interpreted Hufsmith in view of Mohanty, which appears to be unrelated to Hufsmith. Hufsmith focuses on calculating a “Combined Threat Score for Container Images,” particularly for distributed applications (Ex.1078 (Hufsmith), Title, Abstract), and Mohanty focuses on prioritization of vulnerabilities of “an asset (e.g., a virtual machine or a virtual application)” (Ex.1075 (Mohanty) 4:4-5) and is entirely silent about container images for distributed applications. Nor does Dr. Stavrou assert that a person of ordinary skill in the art would have made any modifications to Veselov or Hufsmith based on Mohanty.

110. Similarly, Dr. Stavrou cites to Ex.1063 (Hibbert) for support without explanation. Ex.1002 (Stavrou Decl.) ¶162 (quoting Ex.1063 (Hibbert) ¶187). The quoted passage says nothing about prioritizing alerts “according to the use determinations.” *See id.* And again, Dr. Stavrou does not explain why a person of ordinary skill in the art would have interpreted Hufsmith in view of Hibbert, which appears to be unrelated to Hufsmith. Hufsmith focuses on calculating a “Combined Threat Score for Container Images,” particularly for distributed applications (Ex.1078 (Hufsmith), Title, Abstract), and Hibbert focuses on agent-based risk assessments of applications on a device (Ex.1063 (Hibbert) Abstract) and is entirely silent about container images for distributed applications or even virtual machines. Nor does Dr. Stavrou assert that a person of ordinary skill in the art would have made any modifications to Veselov or Hufsmith based on Hibbert.

111. Dr. Stavrou then cites to Hufsmith’s “discussion of presenting annotations in a development environment,” i.e., displaying information about a vulnerability in order to show a developer particular lines of source code in a pre-deployment container image undergoing development that should be addressed. Ex.1002 (Stavrou Decl.) ¶163 (citing Hufsmith ¶¶154, 156, 180-183). However, a person of ordinary skill in the art would have understood this context to be inapplicable to Veselov, which only analyzes deployed VMs or snapshots where such source code would be unavailable for such annotating. Dr. Stavrou does not

explain how or why persons of ordinary skill in the art could have or would have implemented this feature in Veselov's systems. Although he asserts that this disclosure "also appl[ies] to the other embodiments," Dr. Stavrou explicitly relies on "annotations" of the source code that persons of ordinary skill in the art would understand *not* to apply to situations, such as in Veselov, where *source code is not available to be annotated*. Ex.1002 (Stavrou Decl.) ¶¶163-164. And without the source code available, Hufsmith's stated objectives of annotating the source code—e.g., to increase awareness "earlier in the workflow of container development" and identify vulnerabilities "in a more real time manner" (Hufsmith ¶160)—do not apply. A person of ordinary skill in the art would not have had a reasonable expectation of success in combining these cited portions of Hufsmith with Veselov as a result, and because the targets of Veselov's security assessments are expressly designed to be "read-only." Veselov 18:1-3 ("read only to avoid inadvertently modifying the volume corresponding to a volume image."), 18:26-29 ("read-only virtual hard disk"), Fig. 6 ("read-only file system"); *see also id.* 18:38-39 ("volume image can be read"), 17:5-8 (scanning service can "read the copied files"), 17:23-29 ("read files").. Even so, Dr. Stavrou does not explain how such annotations of source code (see Hufsmith's Figure 3B, below) would be included in "reporting the determined [plurality of] potential cyber vulnerabilities, *as prioritized alerts* according to the use determinations." *Id.* Simply making it

“available to the user even when a partial report is presented” does not constitute an alert as discussed above. Rather, because the pre-deployment context in which Hufsmith’s alleged “alerts” are displayed, where such annotations can trigger a change to source code, is entirely different than the read-only execution environment of Veselov, a person of ordinary skill in the art would not have expected success or have combined Veselov and Hufsmith as Dr. Stavrou alleges.

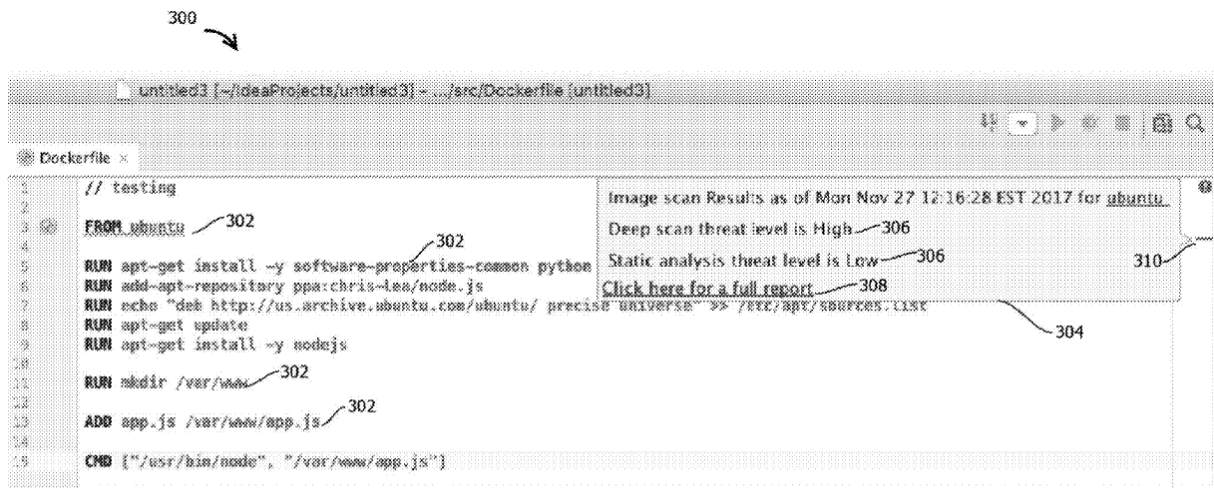


FIG. 3B

Hufsmith, Figure 3B

c. Dependent Claim 2

112. Claim 2 depends from claim 1 and recites “wherein determining whether the matching installed applications are used by the protected virtual cloud asset includes determining whether at least one of the matching installed

applications is not in use by the protected virtual cloud asset, and wherein prioritizing reduces priority of potential cyber vulnerabilities for a matching installed application not in use.” Dr. Stavrou asserts that Hufsmith teaches this limitation. Ex.1002 (Stavrou Decl.) ¶166.

113. Specifically, Dr. Stavrou relies on his analysis of limitation 1.5, stating that the claim 2 limitation is taught “for the same reasons.” Ex.1002 (Stavrou Decl.) ¶167. I disagree. As I discussed above for limitation 1.5, Hufsmith’s disclosures relate to application components, such as subroutines, specific code, or a file of a larger application, and as such the analysis thereof presumes execution and use of the application itself. Thus, Hufsmith does not determine whether any “installed application” is in use or not in use, either in relation to limitation 1.5 or claim 2. Of course, without making such determination, Hufsmith also does not disclose reducing priority of potential cyber vulnerabilities for such an application not in use.

d. Dependent Claims 3-5

114. Dependent claims 3-5 all require determining whether “at least one of the matching installed applications is not in use.” Hufsmith does not teach this limitation for the reasons discussed above for claim 2 and limitation 1.5.

115. Specifically, claim 3 depends from claim 1 and recites “wherein determining whether the matching installed applications are used by the protected

virtual cloud asset includes checking configuration files of the matching installed applications to determine whether at least one of the matching installed applications is not in use by the protected virtual cloud asset, and wherein prioritizing reduces priority of potential cyber vulnerabilities for a matching installed application not in use.” Dr. Stavrou relies on his claim 2 analysis, which I disagree with as discussed above. Ex.1002 (Stavrou Decl.) ¶171. Thus, the combination does not render obvious claim 3.

116. Claim 4 depends from claim 1 and recites “wherein determining whether the matching installed applications are used by the protected virtual cloud asset *includes verifying access times to files by an operating system* of the protected virtual cloud asset to determine whether at least one of the matching installed applications is not in use by the protected virtual cloud asset, and wherein prioritizing reduces priority of potential cyber vulnerabilities for a matching installed application not in use.”

117. Dr. Stavrou asserts that “Hufsmith does not expressly state that [execution environment] records include access times to files by an operating system.” Ex.1002 (Stavrou Decl.) ¶175. I agree. However, he asserts that “such access times were a common part of such records, and application-use determinations such as those described in Hufsmith were routinely and predictably performed by verifying such access times.” Ex.1002 (Stavrou Decl.) ¶175. Dr.

Stavrou cites to Ex.1079 (Carroll) and Ex.1063 (Hibbert) for support. *Id.* But as discussed above for claim 2 and limitation 1.5, Hufsmith does not perform “application-use determinations,” nor are such determinations applicable for Hufsmith’s container images, especially in Hufsmith’s pre-deployment scenario, where files would not be accessed by applications at all (as applications are not executing). For example, Dr. Stavrou does not explain how disclosures in Carroll regarding “the Office suite” and “most recently used files” or “user activity” would be combined with Hufsmith or used in Hufsmith to teach determining “whether at least one of the matching installed applications is not in use” as required by this claim. Similarly, Dr. Stavrou does not explain how Hibbert’s disclosures regarding agent-based risk assessments of applications on a device apply to Hufsmith, which details a different context of container development, as detailed throughout this declaration. Thus, the combination does not render obvious claim 4.

118. Claim 5 depends from claim 2 and recites “wherein determining whether the matching installed applications are used by the protected virtual cloud asset includes *analyzing application logs or system logs* to determine matching installed applications not in use by the protected virtual cloud asset, and wherein prioritizing reduces priority of potential cyber vulnerabilities for at least one of the matching installed applications not in use.”

119. Claim 5 is not obvious for its dependence on claim 2. It is also not obvious for similar reasons as claim 4. Dr. Stavrou similarly asserts that “application logs and system logs were routinely kept in cloud systems” and that “such logs were routinely analyzed to determine application usage.” Ex.1002 (Stavrou Decl.) ¶177. Dr. Stavrou again cites to Ex.1079 (Carroll) and Ex.1063 (Hibbert) but does not explain how such disclosures would motivate a person of ordinary skill in the art to modify the Veselov-Hufsmith combination. Dr. Stavrou specifically ignores the fact that determining whether installed applications are used, including in the manner recited in claim 5, would have added significant complexity to Veselov contrary to its specific teachings. Veselov 2:42-63 (Veselov’s disclosure “overcome[s] drawbacks of existing security assessment systems” such as “complexity of... monitoring.”). Thus, the combination does not render obvious claim 5.

e. Dependent claims 16 and 20

120. Dependent claims 16 and 20 depend from claims 1 and 18, respectively, and recite “wherein determining the location of the snapshot of at least one virtual disk further includes querying a cloud management console of the cloud computing environment for the location of the snapshot and the *location of the virtual disk.*”

121. Dr. Stavrou asserts that “an environment API or web interface to communicate with the computing resource service provider hosting the VM” constitutes the “cloud management console of the cloud computing environment.” Ex.1002 (Stavrou Decl.) ¶228. I first note that “computing resource service provider” is not typically used in the art to refer to a cloud computing environment. Nevertheless, although the examples of Figures 1 and 3A-3B include a computing resource service provider and APIs, the examples of Figures 5A-5B do not include such a computing resource service provider or APIs. But with respect to the Figures 5A-5B environment, where there is no API or computing resource service provider, Veselov discloses that a snapshot of the VM can be “obtained.” Ex.1007 (Veselov) 18:18-21. And in Figures 3A, 3B, 4, 5A, 5B, Veselov teaches that the volume image may be *provided to* a hypervisor or service that launches an assessment machine, or may be read by the hypervisor to launch the assessment machine. Veselov, 12:37-51; 18:16-21. A person of ordinary skill in the art therefore would understand that APIs are not needed to obtain snapshots in Veselov, and therefore can be, and would be, obtained by means other than APIs even in Figures 1 and 3A-3B, and certainly within Figures 5A-5B, where no API exists.

122. Dr. Stavrou further asserts that determining the snapshot’s location “at a minimum would have suggested that such a query would also determine the

virtual disk's location.” Ex.1002 (Stavrou Decl.) ¶232. For example, Dr. Stavrou asserts that the snapshot and virtual disk can have the “same general location (e.g., being located in the same computing environment).” Ex.1002 (Stavrou Decl.) ¶232. Even if true, that would not suggest, however, actually determining the virtual disk's location. For example, Veselov explicitly shows in Figure 1 that snapshots are located separately from any VM in a “snapshot storage service” external to the computing resource service provider. And Veselov states that “computing systems may be located in a single geographic location or located in multiple, distinct geographic locations” (Ex. 1007 (Veselov), 1:12-15) so the same “general location” does not necessarily mean that a determination of a snapshot location obviously leads to the determination of a virtual disk's location. And, it is especially true that Dr. Stavrou has not shown that this limitation would have been obvious, where Dr. Stavrou fails to provide any reason why the Veselov-Hufsmith combination would *benefit* from such information.

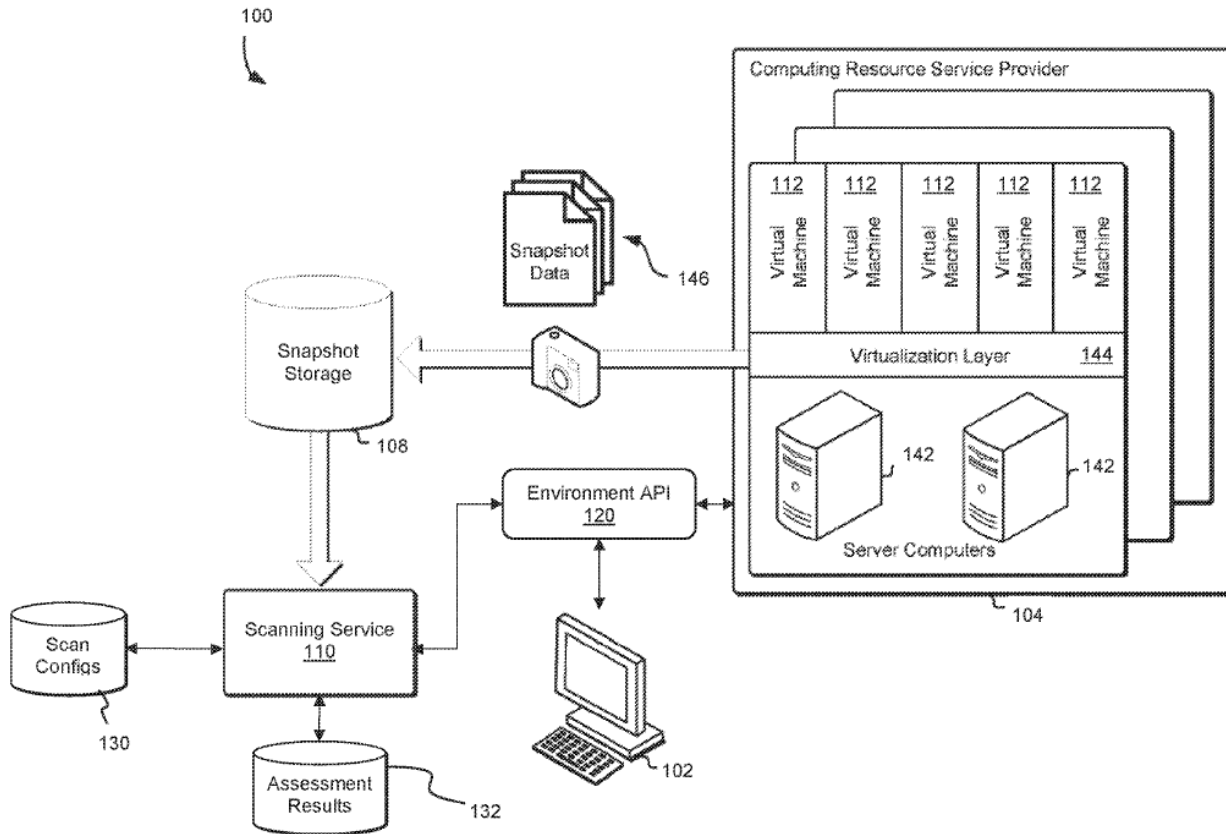


FIG. 1

Ex.1007 (Veslov) Fig. 1

123. At best, Veslov identifies a virtual machine but does not teach or suggest determining a location of a virtual disk. Identifying a VM does not determine a location of a virtual disk. A virtual disk's location cannot be deterministically solely determined based on the location of a virtual machine because virtualization allows for decoupling of the location of the virtual machine and a virtual disk. The fact that Veslov's systems and methods work with and from the identification of VMs (not identifying, let alone determining locations of, virtual disks) belies Dr. Stavrou's assertions about the virtual disk location

determination of claim 16. Dr. Stavrou offers no explanation for why a person of ordinary skill in the art would have changed those systems and methods to instead determine locations of *virtual disks* of a VM. Veselov's system has no need to provide a "UI/API" for the user *to locate a virtual disk*. Dr. Stavrou asserts that this was a "common, routine task for a POSA" but none of the references he cites supports such common, routine knowledge. Ex.1002 (Stavrou Decl.) ¶¶233-234.

124. As the '032 patent explains and shows, virtual disks in cloud computing platforms associated with VMs can be located apart from those VMs. *See* Section VI.A, *see also, e.g.*, Ex.1001 ('032 patent), Fig. 1B, 3:55-60 ("[S]torage 117 emulat[ing] virtual discs for the VMs" "may be part of the server" hosting VMs, or, as depicted in Figure 1B, separate from but "connected to" that server). And, as Dr. Stavrou admitted at his deposition, "Virtual disks are different from virtual machines"—"[t]hey are definitely not synonymous." Ex.2077, (Stavrou Tr.) 97:13-98:10. Thus, the identification of a VM does not necessarily or obviously determine the location of a virtual disk of that VM. Further, identifying a VM does not require or render obvious using an API to "query" such location at least because an identification means the user does not need to query for a location.

125. Dr. Stavrou cites Ex.1015 (VMware vSphere) but does not explain why those disclosures are applicable to Veselov, e.g., Dr. Stavrou makes no effort

to explain that VMware vSphere is used in Veselov or that its teachings would apply to Veselov. Ex.1002 (Stavrou Decl.) ¶231. Dr. Stavrou also quotes Exhibit 1031 as emphasizing the importance of knowing the location of “assets” but omits that this reference is referring to “networked devices and software,” not virtual disks in a cloud computing system. *See* Ex.1031 (NIST IT Asset Mgmt) at 1 (“track the location and configuration of networked devices and software across an enterprise”), 2 (“Our example solution has the following benefits: enables faster responses to security alerts by revealing the location, configuration, and owner of a device.”). This reference says nothing about querying a cloud management console of the cloud computing environment for the location of a virtual disk.

126. Dr. Stavrou also cites to his discussion of claim 13, which recites “wherein determining the location of the snapshot of at least one virtual disk further includes determining a virtual disk allocated to the protected virtual cloud asset.” But his analysis for that section does nothing to address the issues I discuss above.

127. Additionally, in his discussion of claim 13, Dr. Stavrou relies on a handful of references other than Veselov, including Ex.1051 (Workstation Pro), Ex.1053 (Prahlad), and Ex.1020 (Sancheti), purportedly to show that “snapshot generation typically included determining virtual disks allocated to the VM that will be included in the snapshot.” Ex.1002 (Stavrou Decl.), ¶¶221-222. But Dr. Stavrou does not identify anything in those references that would explain why a

person of ordinary skill in the art would have been motivated to change Veselov's systems and methods to identify a location of a virtual disk, as claimed, as opposed to locating a snapshot or identifying a VM generally. Veselov's systems and methods lack such determinations, and I have been informed and understand that, in an obviousness analysis, merely collecting disparate disclosures from a handful of references, to purportedly show that a person of ordinary skill in the art *could have* made the claimed inventions because such disclosures allegedly show that each piece of a claim was known or routine or commonplace is not enough. Rather, I understand that obviousness analyses look to what a person of ordinary skill in the art *would have* been motivated to make or do. In my opinion, nothing in the other references Dr. Stavrou cites show that a person of ordinary skill in the art would have been motivated to change Veselov's systems and methods, at all, let alone to modify those systems and methods as claimed in the '032 patent's claims.⁵

⁵ Dr. Stavrou's tertiary references do not demonstrate that the claimed features are routine or commonplace. Dr. Stavrou cites three pages of Ex.1015 (VMware vSphere) but again does not explain why those disclosures are applicable to Veselov. Similarly, the disclosure Dr. Stavrou cites from the Workstation Pro reference (Ex.1051) simply discloses that data from a disk is included by default in a snapshot of a virtual machine, and makes no mention of a step of determining

a location of a virtual disk. Ex.1051 (Workstation Pro), 47, 125. The VirtualCenter reference (Ex.1052), relied on in Ex.1002, Stavrou Decl. ¶222, shows that there may be an identification for a virtual machine location, but does not demonstrate determining a location of a virtual disk. Ex.1052 (VirtualCenter), 445-46. The disclosures Dr. Stavrou cites from Prahlad (Ex.1053) does not demonstrate determining a location of a virtual disk, nor is it relied upon in Dr. Stavrou's grounds. Ex.1053 (Prahlad), ¶¶77-79, 81-82, 87-88, 92. Nor does Ex.1020 (Sancheti) demonstrate routineness of determining locations of virtual disks—in fact, its disclosure (which is not part of Dr. Stavrou's proposed grounds), supports that merely identifying a VM does not determine a location of a virtual disk. *See* Ex.1002 ¶222 (quoting Sancheti 21:9-41). And Sancheti does not disclose determining a location of virtual disks for the purpose of security assessments. Finally, Dr. Stavrou's citation to Ex.1074 also does not explain how that reference determines a location of virtual disks, or even if it did, how that determination would be incorporated with Veselov to teach this limitation. Rather, Dr. Stavrou's citation to page 12 of that reference suggest it only describes “querying the location of the VM” and not the virtual disk, as required.

f. Summary of Ground 1 Opinions

128. For at least the foregoing reasons, it is my opinion that Petitioner and Dr. Stavrou have not shown that any of the independent claims of the '032 patent—claims 1, 18, and 22—as well as dependent claims 2-5, 13, 16, and 20, would have been obvious. It follows that Petitioner and Dr. Stavrou have not shown that any of the claims depending on those independent claims, including but not limited to the dependent claims challenged in Petitioner's ground 1 (claims 2-11, 13-17, 19-21, and 23-25), would have been obvious. Thus, Petitioner and Dr. Stavrou have not shown that any of claims 1-11, and 13-25 of the '032 patent would have been obvious.

B. Ground 2: Claim 12 Would Not Have Been Obvious Over Veselov, Hufsmith, and Hutchins

129. Dr. Stavrou sets forth Ground 2 to address only the additional limitations of claim 12. Ex.1002 (Stavrou Decl.), ¶¶245-253. But Dr. Stavrou challenges claims 1 and 11, on which claim 12 depends, based on only Veselov and Hufsmith. *See* Ex.1002 (Stavrou Decl.), ¶97. Accordingly, for at least the same reasons Dr. Stavrou fails to demonstrate a reasonable likelihood of prevailing against claims 1 and 11, it is also my opinion that Dr. Stavrou fails to demonstrate a reasonable likelihood of prevailing against claim 12.

130. It is my opinion that claim 12 is also not obvious for the following additional reasons.

1. A Person of Ordinary Skill in the Art Would Not Have Been Motivated to Combine Veselov, Hufsmith, and Hutchins and Would Not Have Had a Reasonable Expectation of Success in Doing So

131. Claim 12 depends from claim 11 and recites “wherein mitigating a potential cyber vulnerability includes one of: blocking traffic from untrusted networks to the protected cloud asset, halting operation of the protected virtual cloud asset, or quarantining the protected cloud asset.”

132. Dr. Stavrou proposes to combine Veselov, Hufsmith, and Hutchins to render claim 12 obvious. Ex.1002 ¶¶245-250. I disagree.

133. A person of ordinary skill in the art would not have been motivated to combine Hutchins with Veselov and Hufsmith. As an initial matter, I discussed above in relation to Ground 1 that a person of ordinary skill in the art would not have been motivated to combine Veselov and Hufsmith in the proposed manner, and that discussion applies here to Ground 2 because Ground 2 assumes the combination of Veselov and Hufsmith. *See* Section VIII.A.1.

134. Additionally, even assuming that Veselov and Hufsmith are combined, a person of ordinary skill in the art would not have combined them with Hutchins to render the claims obvious. Hutchins uses a VM snapshot to replicate a VM to temporarily offload computationally-intensive tasks while both the original VM and replicated VM are active. *E.g.*, Hutchins ¶53. This approach is fundamentally different than Hufsmith’s analysis, which focuses on offline, pre-

deployment, container images. *Compare* Hutchins, Abstract *with* Hufsmith, Abstract; *see also* Section VII.C. I explained above with respect to Ground 1 the fundamental differences, known to a person of ordinary skill in the art, between analyzing snapshots and container images.

135. Hutchins's approach is also different than, and would not have been combined with, Veselov's approach. Veselov discloses scanning operations that are focused on a "scannable volume" from a snapshot captured at a specific point in time. *E.g.* Ex.1007 (Veselov) 10:1-2. In contrast, as discussed above, Hutchins is directed towards offloading resource-intensive (i.e., virus scanning) applications to replicated VMs. Ex.1070 (Hutchins) ¶53; *see also* Section VII.C. Application of Hutchins to Veselov would require additional resources beyond those provided for in Veselov. At best, Hutchins's approach has similarities to Veselov's Figs. 3A/3B examples, which, as discussed above, does not analyze snapshots themselves, and does not discuss or suggest combining Hufsmith's analysis of container images with Veselov's snapshot-based assessments. Dr. Stavrou fails to motivate the application of Hutchins's system with the Figs. 5A/5B examples of Veselov, which operate outside of a computing resource service provider, without Veselov's APIs, and without instantiating duplicate VMs.

2. Even If Combined, Veselov, Hufsmith, and Hutchins Do Not Render Claim 12 Obvious

136. Because claim 12 depends on claims 1 and 11, it is not obvious over Veselov, Hufsmith, and Hutchins for the same reasons that claims 1 and 11 are not obvious in view of Veselov and Hufsmith. *See* Section VIII.A.

IX. Objective Indicia of Non-obviousness

137. I have considered objective evidence that supports that a person of ordinary skill in the art would not have considered the claims of the '032 patent to be obvious, as discussed below. It is my opinion that, even assuming that Petitioner and Dr. Stavrou made a threshold showing of obviousness (they have not for the reasons explained above), there is objective evidence of non-obviousness that shows that the claimed combination of features would not have been obvious to a person of ordinary skill in the art.

A. Orca's SideScanning™ Uses The '032 Patent Inventions

138. I have reviewed Section III of Exhibit 2074, which is the Declaration of Avi Shua. I understand that Mr. Shua is a co-founder of Orca and the named inventor of the '032 patent. In that section of his declaration, Mr. Shua maps Orca's SideScanning™ to the claims of the '032 patent. I have further reviewed the Exhibits cited in that Section of Exhibit 2074. Mr. Shua's mapping is consistent with my review of those Exhibits. Thus, I adopt and incorporate by reference Section III of Exhibit 2074. Based on my review of those Exhibits and

Mr. Shua's mapping, it is my opinion that Orca's SideScanning™ practices the claims of the '032 patent.

B. Long-Felt, Unresolved Need

139. Between 2010-2014, there was a major shift of commercial computing services to cloud computing platforms. Ex.2076, 5-6 (describing the “shift to cloud” beginning in 2010-2014). Amazon introduced the first commercial cloud in 2006, followed by Google in 2008, and Microsoft in 2010. Ex.2134, *The History of Cloud Computing*, Cyber Magazine (Nov. 28, 2021) at 2. By 2014, cloud adoption had “take[n] off” with huge year-over-year increases in adoption. Ex.2134 at 2 (“Cloud adoption takes off” around 2012). As the '032 patent describes, “[o]rganizations [] increasingly adapted their applications to be run from multiple cloud computing platforms.” '032 patent, 1:23-24. Virtualization technologies such as virtual machines, software containers, and serverless functions helped enable this shift to commercial cloud computing. See '032 patent, 1:39-42.

140. As applications moved to the cloud, security lagged behind. “In the early day of cloud, [security] vendors rushed to deploy their existing solutions into this new environment.” Ex.2076 at 4. But even as vendors realized that cloud computing posed new security challenges, new solutions had a number of drawbacks. See Ex.2136, Cloud Security Report, 2 (2018) (“As more workloads

are shifting to the cloud, cybersecurity professionals remain concerned about security of data, systems, and services in the cloud.”); Ex.1029, *Overview of Attacks on Cloud Computing* (April 2012) at 1 (“[S]ecurity is the biggest problem of” cloud computing systems). As the ’032 patent describes, the cloud security industry developed a number of methods for protecting cloud computing infrastructures, including: (1) “traffic inspection”, (2) using “network scanner[s] deployed out of path,” (3) “traffic monitor[ing],” and (4) using “agents installed in each server,” but these methods suffered from multiple disadvantages. ’032 patent, 1:47-2:17.

141. Generally, these problems relate to (1) limited visibility and (2) limited coverage. The problem of limited visibility includes problems with inspecting encrypted traffic, scanning servers that either don’t respond or risk exposing sensitive data, and inability to detect misconfiguration based on only observing, e.g., network traffic. ’032 patent, 1:52-2:9. The problem of limited coverage, which was especially true with agent-based systems, includes difficulties associated with installing agents on every VM. Thus, while an agent may have better visibility into a VM’s security posture, the difficulty in installing agents throughout a platform left big gaps in coverage. *See* Ex.2076, 11 (“Traditional agents struggle with complex deployment and maintenance.”); ’032 patent, 2:16-17 (“[I]ninstalling agents in a large data center may take months.”); Ex.2103,

Security Agents Don't Belong In Your Cloud (Aug. 18, 2021) at 3 (“Customers tell me that agent-based approaches are only able to cover up to 50% of the cloud real estate even after several months of deployment.”).

142. Thus, even by 2019, “[a]lthough cloud security has evolved over the years and is becoming more mature, the endless journey to obtain the right level of visibility and control over the cloud workloads [was] still a challenge.” Ex.2137, Diogenes, *The Quest for Visibility and Control in the Cloud*, ISSA Journal, 1 (March 2019); *see also* Ex.2136, Cloud Security Report, 2 (“Organizations continue to adopt cloud computing at a rapid pace,” but “traditional security tools are often not suited for the challenges of dynamic, virtual and distributed cloud environments.”), 3 (“Eighty-four percent [of respondents] say traditional security solutions either don’t work at all in cloud environments or have only limited functionality.”).

143. Accordingly, there was a long-felt, unmet need for a cloud security solution that could provide both visibility and coverage to adequately address “undetected vulnerabilities.” ’032 patent, 1:49-55, 1:60-62, 2:4-9, 2:18-19.

144. Orca’s claimed inventions solved this need. As discussed in the following sections, Orca’s claimed approach, embodied in its SideScanning™ product, was a “genuinely genius innovation” such that cloud security posture management “can be defined by pre and post-Orca.” Ex.2076 at 7. I have also

reviewed articles by Wiz and Palo Alto Networks that describe agentless scanning, and its benefits, in a manner that closely matches the claimed invention. For example, I have reviewed Exhibit 2138, which appears to be a Wiz article from September 28, 2021, titled “Agents Are Not Enough.” Ex.2138 at 1. In this article, Wiz describes: “[A]gentless deep scanning approaches [] achieve a level of comprehensiveness in terms of coverage and features that is not possible with agent-based approaches due to the tradeoff between performance and impact to the environment.” Additionally, I have reviewed Exhibit 2135, which appears to be a Palo Alto Networks article from November 16, 2021, titled “Palo Alto Networks Shifts Left with Prisma Cloud 3.0.” Ex.2135 at 1. In this article, Palo Alto Networks describes: “agentless scanning... works off cloud provider APIs and disk snapshots” to “extend[] coverage to both active *and dormant* virtual machines”. Ex.2135 at 3.

145. “Orca launched the first version of **agentless side scanning**, which allowed deep visibility into cloud workloads” “without the need to install agents.” Ex.2076 at 11.

Orca Security made a *big leap* in cloud security with the introduction of its patented agentless SideScanning technique in 2019. This innovation introduced workload visibility without needing security teams to install an agent on endpoints or installing software on workloads for assessments. They operate out-of-band by accessing

cloud workloads at the runtime storage level, gathering crucial data through the cloud provider's APIs, while scanning disk snapshots.⁶

Ex.2076 at 31.

146.

REDACTED

⁶ See '032 patent, cls. 1, 18, 22 (limitations 1.1-1.3, 18.1-18.3, 22.1-22.3).

REDACTED

C. Orca's claimed inventions overcame industry skepticism to become widely accepted--and widely praised—in the industry.

147. As discussed below, I have reviewed objective evidence that there was industry skepticism of Orca's claimed inventions, but that they were quickly accepted and widely praised by the industry. This evidence further supports that skilled artisans at the time of the invention would not have viewed Orca's claims as obvious.

148. A 2024 report on “the Future of Cloud Security” states “[t]here is no doubt, the pivotal moment was the introduction of agentless scanning by companies like Orca and Wiz”—of course, “Orca launched the first version of agentless side scanning” and, as shown below, Wiz later copied Orca. Ex.2076 at 8, 11. It was a “turning point – but anyone who tells you they knew it at the time is lying.” *Id.* at 8.

149. After Orca “introduced the concept of agentless side scanning,” it was “difficult” for “older providers to come to terms with [its] success.” Ex.2076 at 7. That 2024 report states that these providers were “skeptics” because “from a security perspective, agentless scanning was nothing compared to what attacks

⁷ See '032 patent, cls. 1, 18, 22 (limitations 1.7, 18.7, 22.7).

they were able to detect and protect.” *Id.* “They missed however that security teams were not yet in the weeds of these architectures,” so before Orca, they “didn’t know what they didn’t know.” *Id.* 7. And in 2021, Palo Alto Networks released an article about how Orca’s solutions “fall short.” Ex.2106, *How Agentless Approaches Sideline Cloud Workload Protection Strategies*, Palo Alto Networks (Feb. 23, 2021) at 1 (“Agentless Solutions Fall Short”). By November, Palo Alto Networks announced their own agentless scanning solution. Ex.2135, *Palo Alto Networks Shifts Left with Prisma Cloud 3.0* at 2 (announcing “Agentless Security for Flexible Cloud Workload Protection” that “Provide[s] visibility into an organization’s cloud workload and application risks”).

150.

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

151. Eventually, “everyone needed [agentless scanning] in order to stay competitive.” Ex.2076, 7; Ex.2103, *Security Agents Don’t Belong In Your Cloud* (Aug. 18, 2021) (“[A] new ‘agent-less’ approach to cloud security has recently become possible due to cloud-native APIs now available from CSPs. This

approach provides security in a fundamentally different way.”). As I mentioned above, Palo Alto Networks quickly reversed course on its public criticisms of the approach, and Wiz entered the market a year after Orca with the same technology—copied from Orca.

D. Extensive Industry Praise Supports Non-Obviousness of the Claims

152. I have reviewed evidence of industry praise of Orca’s inventions that further shows that skilled artisans did not view (and a person of ordinary skill in the art would not have viewed) the inventions as obvious. This evidence includes numerous industry awards, including:

- CRN’s “20 Coolest Tech Startups of 2020.” Ex.2119, *The 20 Coolest Tech Startups of 2020*, CRN (Nov. 19, 2020) at 9-10. CRN touted “Orca’s SideScanning technology” which “automatically assesses the security state of every discovered asset” to provide “complete visibility.” *Id.*
- Forbes’s “Top 20 Cybersecurity Startups to Watch in 2021.” Ex.2120, *The Top 20 Cybersecurity Startups to Watch in 2021 Based on Crunchbase*, Forbes (Nov. 20, 2020) at 6-7. Forbes found Orca “noteworthy for its innovative approach to providing instant-on, workload-deep security” for cloud computing systems, emphasizing Orca’s “patent-pending SideScanningTM technology” that “reads cloud configuration and workloads’ runtime block storage out-of-band.” *Id.*

- SC Media's Best Cloud Workload Protection Solution.
Ex.2123, *SC Award Winners 2023 Orca Security – Best Cloud Workload Protection Solution*, SC Media (Aug. 21, 2023)
("After years of dealing with agents, then seeing how easy it is to install and use Orca, I knew that its agentless approach was both a major innovation and a game changer")

153.

REDACTED

REDACTED

REDACTED

REDACTED

155. I have also reviewed evidence of public praise of Orca's inventions, including:

- Exhibit 2076, which I have been informed is a 2024 report published on Substack by Software Analyst Cyber Research (founded by Francis Odum) and James Berthoty that heaped praise on Orca's inventions specific to the claimed combination of features of the '032 patent, such as using cloud computing platform APIs to determine locations of virtual disks and virtual disk snapshots and facilitate snapshot analyses. Select quotes include:

Orca: Introduced the concept of **agentless side scanning**, which allowed for deep visibility into cloud environments, and workloads, without the need to install agents on each workload. CSPM can be defined by pre and post-Orca: the ability to view workload data without an agent was a **genuinely genius** innovation - utilizing the nature of cloud APIs being accessible to vendors.

Id. at 7

Orca Security made a big leap in cloud security with the introduction of its patented **agentless SideScanning** technique in 2019. This **innovation** introduced workload visibility without needing security teams to install an agent on endpoints or installing software on workloads for assessments.

Id. at 31

They operate out-of-band by accessing cloud workloads at the runtime storage level, gathering crucial data through the cloud provider's APIs, while scanning disk snapshots. Their technology evaluates block storage bytes and reconstructs file systems, scans operating systems, and applications in a read-only mode, ensuring that virtual assets remain unaffected. This **unique** method allows companies to deploy Orca across their entire cloud environment without worrying about performance impacts or operational disruptions. For security teams, this means getting visibility into engineering workloads, without needing to bother them at all along the way.

***Id.* at 31**

Orca's SideScanning technology has proven **highly effective** for vulnerability assessments and scanning with a fast and unified database for attack paths especially for non-container assets. While Orca's advantages in agentless remains a key strength, it also has shown some limitations as cloud security teams start looking for more runtime data and visibility into their workloads. Critically, agentless scanning has always had difficulties being applied to ephemeral workloads, where what's on disk is changing minute by minute.

***Id.* at 31**

- **Quick deployment and rapid-time to value:** Agentless solutions are easier to deploy since they do not require installing software on each workload. They **leverage cloud APIs to snapshot underlying instance hard disks**, copy them into the security vendors account, where they're hooked up on the vendor's side and scanned. This creates an **elegant "like magic" approach** - where you get full visibility into the workload, without any instrumentation.

***Id.* at 11-12**

- Exhibit 2118, which is titled "Vendor Landscape and TCO Analysis for Achieving Cloud-Wide, Workload-Deep Security and Compliance for Public Cloud Environments." Ex.2118 at 1. This document describes "Orca uses a patent-

pending technique called SideScanning™, ... which unobtrusively scans the run-time block storage layer—out of band—to extract valuable data about the environment and the assets communicating in it.” Ex.2118 at 5. The report states that a Senior Cloud Security Engineer from a company called Sisense described Orca’s solution as “unique in that it locates vulnerabilities with precision and delivers tangible, actionable results—without having to sift through all the noise.” *Id.* at 4.

156. In summary, Orca’s extensive praise for the claimed combination of features in the ’032 patent is objective evidence that persons of ordinary skill in the art would not have viewed Orca’s claims as obvious at the time of invention.

E. Evidence Shows Wiz Copied Orca’s SideScanning™

157. I have reviewed evidence that Wiz copied Orca’s SideScanning™. This evidence, which I discuss below, further demonstrates that a person of ordinary skill in the art, at the time of the invention, would not have found the ’032 patent’s claims obvious.

158.

REDACTED

REDACTED

159. REDACTED

160. REDACTED

161. In the following section I discuss additional evidence that Wiz copied Orca's product.

1. Evidence shows Wiz had access to Orca's invention

162. I have been informed that Wiz had access to Orca's invention at least because Orca was founded before Wiz and released its SideScanning™ before

Wiz. Specifically, I have been informed that Orca and Wiz were both startup companies, that Orca was formed before Wiz, and that Orca presented its SideScanning™ technology to Wiz's founder when he (Wiz's founder) was still at Microsoft in May 2019.

163.

REDACTED

164.

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

182. As the above-discussed evidence shows, Wiz accessed and copied Orca's claimed inventions.

2. Evidence shows Wiz's Workload Scanning is substantially similar to Orca's SideScanning™

183. Wiz's Cloud Security Platform uses what Wiz calls "workload scanning," which is substantially similar to Orca's SideScanning™.

184. REDACTED

185. REDACTED

REDACTED

REDACTED

REDACTED

186. I have also reviewed some of Wiz's public documents, which show a substantial similarity with Orca's SideScanning™, including:

- Exhibit 2026, which I am informed is a webpage on Wiz's website. This webpage touts the benefits of agentless scanning and states that agentless analysis is "based on two fundamental principles" including "Privileged access to customer's cloud environment *via APIs*" and "*Snapshot scanning*." Ex.2026 at 3. "By connecting to cloud APIs, they can discover new resources as they're created, without requiring manual installation of an agent process. This maximizes security coverage from day one, improving the visibility of security issues." Ex.2025 at 8.

What is agentless security?

Agentless security provides visibility into the threats in your environments, without requiring the installation of software-driven agents.

Under the agent-based model, each of your hosts must run a monitoring process that collects data from the host's environment and sends it to your security service. Agentless security removes this requirement by having the service collect data itself, using cloud provider APIs and metadata.

Agentless analysis approaches are based on two fundamental principles:

- Privileged access to customer's cloud environment via APIs
- Snapshot scanning

Privileged access assigned to the security vendor enables discovery of all resources and services used in the cloud environment. This information is then processed to determine the list of workloads to be scanned.

Ex.2026 at 3

Advantages of agentless security

Agentless security solves most of the problems associated with agents. Instead of running an agent in each of your environments, agentless services sit *outside* your resources. They collate security information by monitoring data provided by cloud APIs and infrastructure services. This model presents several compelling advantages for security teams and administrators.

Simple, automatic coverage

Agentless platforms automatically monitor the resources in your cloud provider accounts. By connecting to cloud APIs, they can discover new resources as they're created, without requiring manual installation of an agent process. This maximizes security coverage from day one, improving the visibility of security issues.

Ex.2026 at 8

- Exhibit 2115, which I have been informed is a webpage on Wiz's website, and is titled "Securing clouds, securely." Ex.2115 at 1. This page describes "agentless scanning approaches" as "cloud native and API-based." Ex.2115 at 1. This page also describes the Wiz approach to agentless scanning, which it calls "workload scanner." Ex.2115 at 3. For example, in a "Full SaaS Wiz-hosted workload scanner," the scanner has "limited permissions to run the scan – including creating, sharing, and deleting snapshots." Ex.2115 at 3. I note that the "limited permissions" here refer at least in part to a policy file I discussed in the previous section.

Full SaaS Wiz-hosted workload scanner — In this model, the Wiz workload scanner is hosted in a Wiz cloud environment, with limited permissions to run the scan — including **creating, sharing, and deleting snapshots**. During the scan, the scanner has read-only access to the scanned disk, so some environments are a better fit for the hosted model (highly regulated, sensitive data, etc). This model offers several benefits including ease of deployment, scalability, and lack of ongoing management.

Ex.2115 at 3

This is one of the reasons that agentless scanning approaches to securing the cloud — approaches that are cloud native and **API-based** — have taken over the cloud security market, replacing legacy agent-based practices. At Wiz, we see this firsthand: from the pain of customers who want certainty and context, to the relief they feel when they gain that much-needed visibility. As **the CISO of Fox said**, “Wiz came into the picture to allow us to feel secure and confident in how fast we’re moving, even as our cybersecurity challenges keep changing.”

Ex.2115 at 1

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

195. In the discussion that follows, I discuss additional exhibits I have reviewed that further support the substantial similarity between Wiz and Orca's products as it relates to specific limitations of the claims of the '032 patent.^{REDACTED}

Notably, the below is not an infringement analysis, but rather a discussion of how Wiz's product is substantially similar to Orca's product with respect to the below limitations.

- a. **Wiz's Workload Scanner is substantially similar to Orca's SideScanning in relation to limitations 1.1-1.2, 18.1-18.2, 22.1-22.2.**

196. Limitations 1.1-1.2 recite "*determining, using an API or service provided by the cloud computing environment, a location of a snapshot of at least one virtual disk of a protected virtual cloud asset, wherein the protected virtual cloud asset is instantiated in the cloud computing environment*" and "*access, based*

on the determined location and using an API or service provided by the cloud computing environment, the snapshot of the virtual disk.” Wiz’s Workload Scanner is substantially similar to Orca’s SideScanning™ in relation to limitations 1.1, 18.1, and 22.1 of claims 1, 18, and 22 of the ’032 Patent.

197. For example, similar to Orca’s SideScanning™, Wiz’s Workload Scanner is based on assigning privileges to access and scan an environment using cloud APIs. For example, in Exhibit 2115, which I discussed above, Wiz characterizes “Agentless scanning approaches” as being “based on assigning privileges to security vendors to access and scan an environment using cloud APIs.” Ex.2115 at 1. Wiz then “outline[s]” its “approach to secure agentless scanning” by describing the “Wiz workload scanner.” Ex.2115 at 2. I therefore understand Wiz’s description of “Agentless scanning approaches” in this document to encompass its own Workload Scanner.

Agentless scanning approaches are based on assigning privileges to security vendors to access and scan an environment using cloud APIs. When selecting an agentless scanning technology, it’s important to assess the privileges assigned to the security vendor, to ensure **least privileges** and secure access to your data. In addition, it’s important to understand exactly which data is being shared with the vendor and how that data is shared across cloud environments.

Ex.2115 at 1.

198. Wiz’s Workload Scanner thus includes an onboarding process where a client environment provides Wiz’s Workload Scanner with “privileges” that

allow access to an environment and to scan the environment using cloud APIs. *Id.*

This establishes an interface between a client organization's cloud computing environment implemented by one or more cloud computing platforms, such as AWS, Azure, or Google Cloud Platform (GCP), and Wiz's Workload Scanner.

REDACTED

200. Wiz's Workload Scanner uses the interface between a client's environment and Wiz's security components to use an API to determine a location of a snapshot of a virtual disk of a protected virtual cloud asset.

The Wiz workload scanner

Traditional cloud security posture management tools are focused on the configuration layer only. When we started Wiz, we understood that in order to analyze an end-to-end risk, we would need to analyze *multiple* layers, including network, identity, data, and the workloads themselves. This is why we built a workload scanner that is designed for the cloud and uses cloud APIs, instead of relying on older agent-based approaches.

Ex.2115 at 2

REDACTED

REDACTED

REDACTED

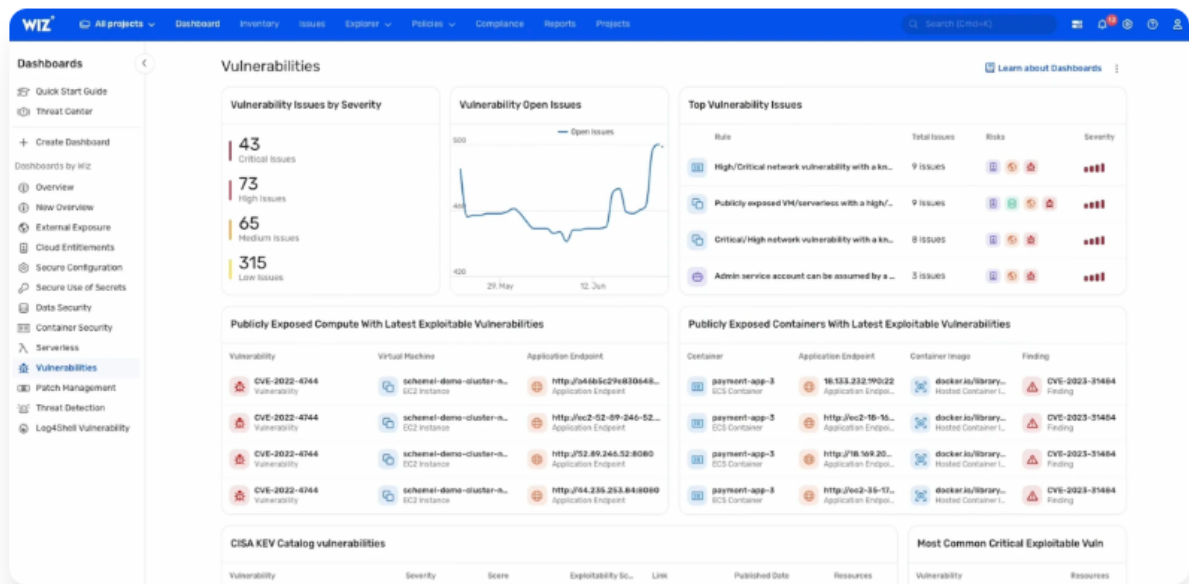
- b. Wiz's Workload Scanner is substantially similar to Orca's SideScanning in relation to limitations 1.3-1.5, 18.3-18.5, 22.3-22.5.**

202. Limitations 1.3-1.5 recite “*analyze the snapshot of the at least one virtual disk by matching installed applications with applications on a known list of vulnerable applications*” and “*determine, based on the matching, an existence of a plurality of potential cyber vulnerabilities*” and “*determine whether the matching installed applications are used by the protected virtual cloud asset.*” Wiz's Workload Scanner is substantially similar to Orca's SideScanningTM in relation to

limitations 1.3-1.5, 18.3-18.5, and 22.3-22.5 of claims 1, 18, and 22 of the ’032 Patent (reproduced in full above).

203. For example, Wiz’s workload scanner “analyze[s] an end-to-end risk” by “analyz[ing] multiple layers, including network, identity, data, and the workloads themselves.” Ex.2115 at 2. Additionally, Wiz performs “vulnerability analysis” based on the snapshot.

Stage 6: Vulnerability analysis



Example vulnerability dashboard that prioritizes issues by contextual severity

After scanning, the tool will generate a comprehensive list of identified vulnerabilities, order them based on severity, filter away false positives, and provide options for remediation.

Ex.2165 at 8

204. Additionally, Wiz detects and prioritizes a catalog of “known exploited vulnerabilities.” See Ex.2147.

What is the CISA KEV Catalog?

CISA, the Cybersecurity & Infrastructure Security Agency, released a **binding operational directive** to reduce the significant risk of known exploited vulnerabilities. In recognition that vulnerabilities that have been previously exploited are a frequent vector for malicious cyber actors, CISA prioritizes these vulnerabilities as the biggest threats that must be aggressively remediated. Since November 2021, CISA has managed a catalog of known exploited vulnerabilities that carry significant risk, along with remediation requirements. Federal agencies are compelled to patch these vulnerabilities within the timeframe defined by CISA.

Currently, there are more than 352 vulnerabilities in the **catalog**, covering vulnerabilities in operating systems (e.g. Windows, Linux), proprietary applications (e.g. SolarWinds products, Zoho ManageEngine ServiceDesk), and open source projects (e.g. Apache HTTP Server, Log4j).

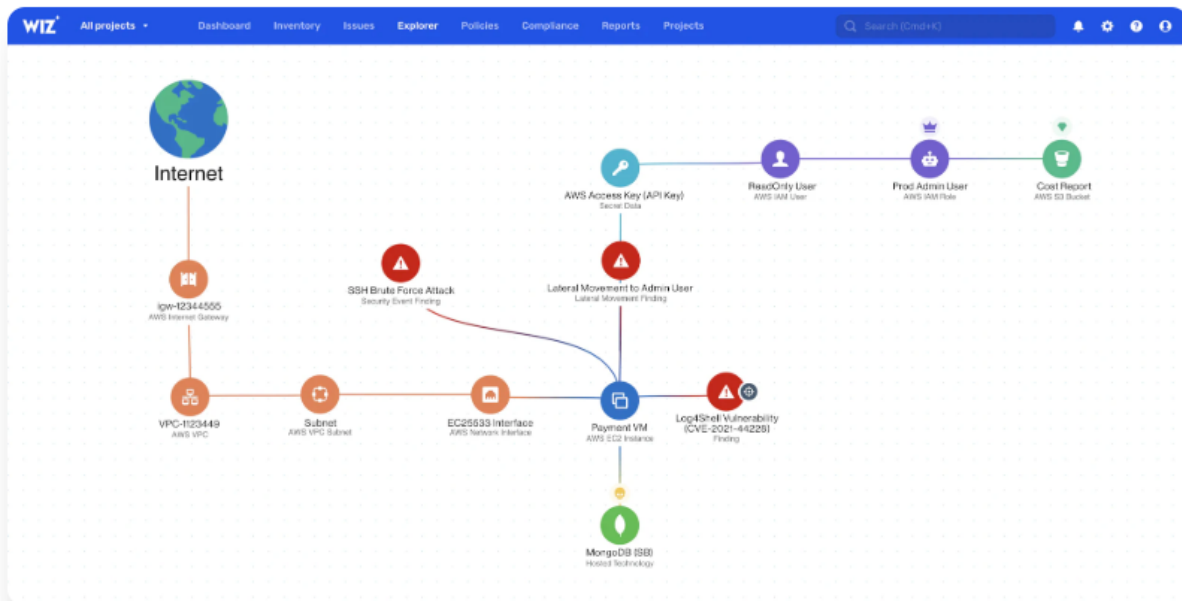
The **latest vulnerability**, a Microsoft Win32k privilege escalation vulnerability (CVE-2022-21882), was added on February 4th, 2022 and required remediation action within 14 days. Even though the CISA directive is binding for US federal agencies specifically, it is quickly becoming a best security practice for public and private sectors as well.

Ex.2147 at 1-2**c. Wiz's Workload Scanner is substantially similar to Orca's SideScanning in relation to limitations 1.6-1.7, 18.6-18.7, 22.6-22.7**

205. Limitations 1.6-1.7 recite *“prioritize the potential cyber vulnerabilities based on the use determinations”* and *“report the determined plurality of potential cyber vulnerabilities, as prioritized alerts according to the use determinations.”* Wiz's Workload Scanner uses the features recited in limitations 1.6-1.7, 18.6-18.7, and 22.6-22.7 of claims 1, 18, and 22 of the '032 Patent (reproduced in full above).

206. For example, Wiz's Workload Scanner performs "Risk-based prioritization" of vulnerabilities that purports to "mitigate the amount of alert fatigue you experience."

3. Risk-based prioritization



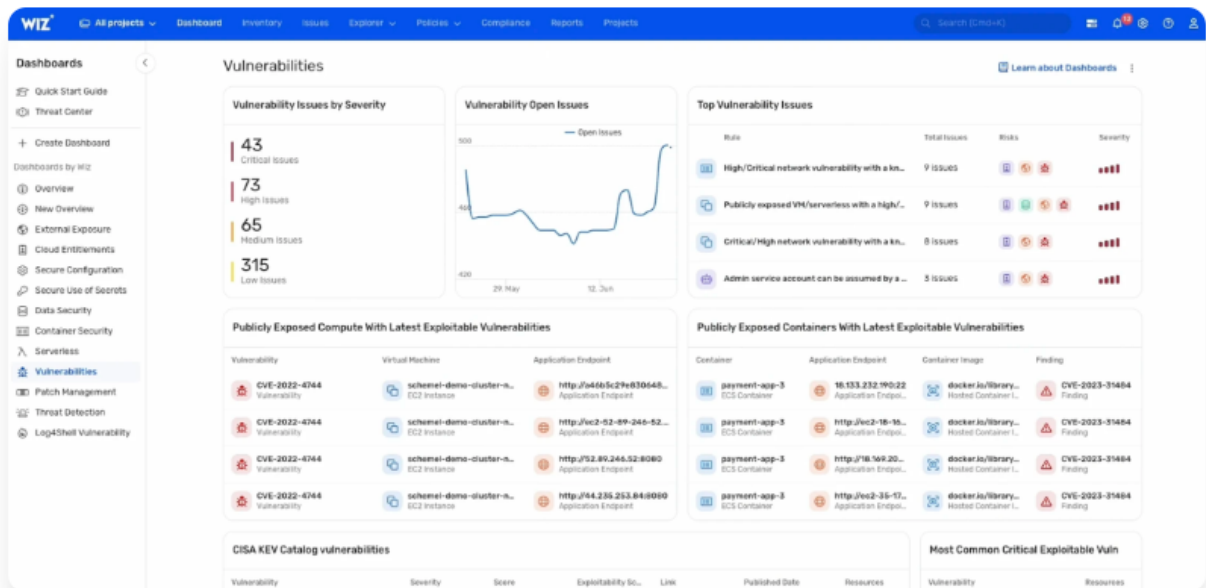
Example visualization of the risk-based context that a vulnerability scanner should provide

Choose a tool that provides risk-based prioritization of vulnerabilities, considering factors such as severity, exploitability, and asset criticality alongside elements such as external exposure, **cloud entitlements**, secrets, misconfigurations, and malware presence. A tool with this functionality can correlate your vulnerabilities that have numerous risk factors to mitigate the amount of alert fatigue you experience.

Ex.2165 at 14-15

207. Wiz's Workload Scanner reports issues as alerts.

Stage 6: Vulnerability analysis



Example vulnerability dashboard that prioritizes issues by contextual severity

After scanning, the tool will generate a comprehensive list of identified vulnerabilities, order them based on severity, filter away false positives, and provide options for remediation.

Ex.2165 at 8

F. Commercial Success

208. I have reviewed evidence that Orca's SideScanning™ was commercially successful, including investment funding of more than \$600 million and REDACTED Ex.2140, *Orca 2020 Recap*, 2; Ex.2112, *Upstart Orca Security Raises \$210M In Google-Backed Funding*, CRN, 1-2; Ex.2139, *Orca Security Raises \$340M To Scale Platform R&D, Channel*, CRN, 2; Ex.2312. I have also reviewed evidence that Orca's innovations created a new market for agentless Cloud Security Posture Management. *See* Ex.2076, 7

(“CSPM can be defined by pre and post-Orca”); *see also* REDACTED

209. I have also reviewed evidence that Wiz’s Cloud Security Platform, which uses what it calls “workload scanning” and which practices the claims (as shown above) has been commercially successful as well, including claims that it became the fastest-growing software company in history, and articles describing valuations of the company in the tens of billions of dollars. Ex.2116, *\$100M ARR in 18 months: Wiz becomes the fastest growing software company ever*, 2 (Aug. 10, 2022); Ex.2141, *Cloud Security Startup Wiz, Now Valued at \$10B, Raises \$300M*, 2; Ex.2142, *Wiz walks away from \$23 billion deal with Google, will pursue IPO* (July 22, 2024).

210. It is my opinion that the commercial success of both Orca’s and Wiz’s products has nexus to the claims of the ’032 patent. For example, both systems practice the claims of the ’032 patent as discussed above. REDACTED

X. CONCLUSION

All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true. Further, I am aware that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001. I declare under penalty of perjury that the foregoing is true and correct.

Executed on 4/7/2025, in Boston, Massachusetts.



Dr. David Kaeli