

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CROWDSTRIKE, INC.,
Petitioner,

v.

SKYSONG INNOVATIONS, LLC,
Patent Owner.

PTAB Case No. IPR2025-01397
U. S. Patent No. 10,313,385 B2

PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	RELATED PROCEEDINGS	1
III.	SKYSONG BACKGROUND	1
IV.	THE ASSERTED PRIOR ART	2
A.	<i>Vallone</i> (EX1004).....	2
B.	<i>Oberheide</i> (EX1005)	3
C.	<i>Patsos</i> (EX1006)	5
D.	<i>Roytman</i> (EX1007).....	6
E.	<i>Juels</i> (EX1008).....	7
F.	<i>Wolff</i> (EX1009)	8
V.	THE '385 PATENT.....	9
A.	Technical Background.....	9
B.	Prosecution History	11
C.	Priority Date	12
VI.	LEVEL OF ORDINARY SKILL IN THE ART	13
VII.	PATENT OWNER’S CLAIM CONSTRUCTIONS	13
A.	“constraint set”	14
VIII.	PETITIONER HAS FAILED TO DEMONSTRATE A REASONABLE LIKELIHOOD OF PREVAILING AS TO ANY CHALLENGED CLAIM.	16
A.	GROUND 1: Obviousness in Light of <i>Vallone</i>	16

1.	Claim 1	17
2.	Claim 8	28
3.	Claim 14	31
4.	Claim 15	32
B.	GROUND 2: Obviousness in Light of <i>Oberheide</i> , <i>Patsos</i> , and <i>Roytman</i>	33
1.	Claim 1	34
2.	Claim 8	41
3.	Claim 14	44
4.	Claim 15	44
5.	Motivation To Combine	46
C.	GROUND 3: Obviousness in Light of <i>Oberheide</i> , <i>Patsos</i> , <i>Roytman</i> , and <i>Juels</i>	47
1.	Claim 2	47
2.	Claim 3	48
3.	Claim 4	49
4.	Claim 5	49
5.	Claim 6	51
6.	Claim 9	51
7.	Claim 10	52
8.	Claim 11	52
9.	Claim 12	53

10. Claim 16.....	53
11. Claim 17.....	53
12. Motivation to Combine	54
D. GROUND 4: Obviousness in Light of <i>Oberheide</i> , <i>Patsos</i> , <i>Roytman</i> , and <i>Wolff</i>	54
1. Claim 7.....	55
2. Claim 13.....	55
3. Motivation to Combine	55
IX. SECONDARY CONSIDERATIONS OF NON-OBVIOUSNESS	56
A. Long-Felt Need and Government Support	56
B. Praise By Others	57
C. Copying By Competitors	57
X. CONCLUSION.....	57

TABLE OF AUTHORITIES

CASES

<i>Graham v. John Deere Co. of Kan. City</i> , 383 U.S. 1 (1966).....	64
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (<i>en banc</i>)	13

TABLE OF ABBREVIATIONS

Patent Owner	Skysong Innovations, LLC (“Skysong”)
Petitioner	CrowdStrike, Inc. (“CrowdStrike”)
PTAB	Patent Trial and Appeals Board
’385 patent	U.S. Patent No. 10,313,385 (EX1001)
<i>Vallone</i>	U.S. Patent Application Publication No. 2016/0134653 to Vallone et al. (EX1004)
<i>Oberheide</i>	U.S. Patent Application Publication No. 2014/0245450 to Oberheide et al. (EX1005)
<i>Patsos</i>	Dimitrios Patsos et al., <i>Expanding Topological Vulnerability Analysis To Intrusion Detection Through The Incident Response Intelligence System</i> , Information Management & Computer Security, Vol. 18, No. 4, 291-309 (2010) (EX1006)
<i>Roytman</i>	U.S. Patent Application Publication No. 2015/0237065 to Roytman et al. (EX1007)
<i>Juels</i>	U.S. Patent No. 9,471,777 to Juels et al. (EX1008)
<i>Wolff</i>	U.S. Patent Publication No. 2013/0198848 to Wolff. (EX1009)

LIST OF EXHIBITS

Exhibit No.	Descriptions
2001	Skysong's First Amended Complaint against CrowdStrike
2002	Declaration of Dr. Paulo Shakarian regarding discretionary denial with Appendix
2003	Civil Docket For Case # 7:25-cv-00040-DC-DTG
2004	<i>Skysong v. CrowdStrike</i> , Dkt. No. 55, Scheduling Order
2005	Table of CrowdStrike IPRs and Parallel District Litigation Schedules
2006	Civil Docket For Case # 2:25-cv-00098-JRG
2007	Revision to Rules of Practice Before the Patent Trial and Appeal Board
2008	Executive Order No. 14,028
2009	Declaration of Dr. Nenad Medvidovic

LIST OF CHALLENGED CLAIMS

Limitation	Claim Language
Claim 1	
1	A system for protecting a computer system from attack, the system comprising:
1.a	a network connection;
1.b	a processing device; and
1.c	a non-transitory computer-readable medium connected to the processing device configured to store instructions that, when executed by the processing device, performs the operations of:
1.d	accessing data through the network connection, the data comprising dark net information associated with exploits of a computer system, comprising:
1.e	obtaining a set of exploits from the dark net information, the set of exploits configured to penetrate the computer system;
1.f	applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities;
1.g	creating a constraint set of the computer system from the set of vulnerabilities, the constraint set comprising a minimum set of dependencies to operate the computer system;
1.h	applying the set of exploits to the constraint set of the computer system;
1.i	analyzing the application of the set of exploits on the computer system to detect a particular vulnerability of the computer system; and
1.j	altering a configuration of the computer system in response to the analysis of the application of the set of exploits to reduce potential damage of a cyberattack.
Claim 2	
2	The system of claim 1, wherein analyzing the application of the set of exploits on the computer system comprises quantifying a risk level posed to the computer system related to a cost to obtain the set of exploits.
Claim 3	
3	The system of claim 2, wherein the processing device further performs the operation of: aligning internet data with the constraint set of vulnerabilities based on information obtained about the set of exploits.

Claim 4	
4	The system of claim 3, wherein obtaining a set of exploits through the network connection comprises: applying one or more algorithms to obtain the set of exploits.
Claim 5	
5	The system of claim 4, wherein the one or more algorithms determine the cost to obtain the set of exploits to create a budget and obtains the set of exploits constrained by the determined cost to obtain the set of exploits.
Claim 6	
6	The system of claim 5, wherein obtaining the set of exploits is based on maximizing the quantifying of the risk level posed to the computer system related to the cost to obtain the set of exploits.
Claim 7	
7	The system of claim 1, wherein altering the configuration of the computer system comprises removing a software component of the computer system that is utilized by the set of exploits to attack the computer system.
Claim 8	
8	A method for improving a computing device, the method comprising:
8.a	accessing data comprising dark net information associated with a computer system;
8.b	obtaining a set of exploits from the dark net information, the set of exploits configured to bypass a security feature of the computer system;
8.c	applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities;
8.d	creating a constraint set of vulnerabilities of the computer system from the set of vulnerabilities comprising a minimum set of dependencies to operate the computer system, wherein application of the set of exploits on the computer system comprises determining the effect of the set of exploits on the constraint set of vulnerabilities of the computer system;
8.e	analyzing an application associated with the set of exploits on the computer system to detect a particular vulnerability of the constraint set of vulnerabilities of the computer system; and

8.f	altering a configuration of the computer system in response to the analysis of the application of the set of exploits to reduce potential damage of a cyberattack.
Claim 9	
9	The method of claim 8, wherein analyzing the application of the set of exploits on the computer system comprises quantifying a risk level posed to the computer system related to a cost to obtain the set of exploits.
Claim 10	
10	The method of claim 9, wherein obtaining a set of exploits comprises: applying one or more algorithms to obtain the set of exploits based on the constraint set of vulnerabilities of the computer system from one or more darknet marketplaces on the Internet.
Claim 11	
11	The method of claim 10, wherein the one or more algorithms determine the cost to obtain the set of exploits to create a budget and obtains the set of exploits constrained by the determined cost to obtain the set of exploits.
Claim 12	
12	The method of claim 11, wherein obtaining the set of exploits is based on maximizing the quantifying of the risk level posed to the computer system related to the cost to obtain the set of exploits.
Claim 13	
13	The method of claim 8, wherein altering the configuration of the computer system comprises removing a software component of the computer system that is utilized by the set of exploits to attack the computer system.
Claim 14	
14	The method of claim 8, wherein altering the configuration of the computer system comprises installing a patch to a software component of the computer system that is exposed by the set of exploits to attack the computer system.
Claim 15	
15	One or more non-transitory tangible computer-readable storage media storing computer-executable instructions for performing a computer process on a machine, the computer process comprising:

15.a	obtaining a set of exploits from dark net information, the set of exploits configured to bypass a security feature of a computer system;
15.b	applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities;
15.c	creating a constraint set of vulnerabilities of the computer system from the set of vulnerabilities comprising a minimum set of dependencies to operate the computer system;
15.d	applying the set of exploits to the constraint set of vulnerabilities of the computer system;
15.e	analyzing the application of the set of exploits on the computer system to detect a particular vulnerability of the computer system; and
15.f	altering a configuration of the computer system in response to the analysis of the application of the set of exploits to reduce potential damage of a cyberattack.
Claim 16	
16	The one or more non-transitory computer-readable storage media of claim 15, wherein analyzing the application of the set of exploits on the computer system comprises quantifying a risk level posed to the computer system related to a cost to obtain the set of exploits.
Claim 17	
17	The one or more non-transitory computer-readable storage media of claim 15, wherein obtaining a set of exploits comprises applying one or more algorithms to obtain the set of exploits from one or more darknet marketplaces on the Internet.

I. INTRODUCTION

Patent Owner Skysong respectfully submits this Preliminary Response to the Petition for *Inter Partes* Review dated Aug 15, 2025 (“Petition”) of U.S. Patent No. 10,313,385 (EX1001, “’385 patent”). This response is timely filed. *See* Paper 4.

II. RELATED PROCEEDINGS

The ’385 patent is the subject of the following related litigation:

- *Skysong Innovations, LLC v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.*, Case No. 7:25-cv-00040, filed January 31, 2025, in the United States District Court for the Western District of Texas; and
- *Skysong Innovations, LLC v. Fortinet, Inc.*, Case No. 2:25-cv-00098, filed January 31, 2025, in the United States District Court for the Eastern District of Texas.

III. SKYSONG BACKGROUND

Skysong is a nonprofit organization, and the exclusive intellectual property management and technology transfer organization for Arizona State University (“ASU”). Skysong works closely with ASU and its technology industry partners to translate ASU’s innovations into commercial applications, fostering economic development and public benefit. *See* EX2001 (Skysong’s First Amended Complaint), ¶¶3–4. Skysong significantly contributes to the mission statement of

the University and has become a core component of Arizona’s innovation ecosystem and a driver of high-tech industry in the state.

The ’385 patent stems from research conducted by Dr. Paulo Shakarian and his team. *See* EX2002, ¶¶5–6, 11. It discloses game-theory based methods to model an attacker based on exploit market data actively mined from the darknet to develop strategies for the defender. *See* EX1001, abstract.

IV. THE ASSERTED PRIOR ART

Dr. Nenad Medvidovic, Department Chair of the Computer Science Department of the University of Southern California and leading artificial intelligence and cybersecurity authority, has been retained by Skysong as an independent expert to assist in the technical analysis of the ’385 patent and the asserted prior art. He has provided a declaration (EX2009, “*Declaration Of Nenad Medvidovic, Ph. D., In Support Of Patent Owner’s Preliminary Response*”) setting forth his opinions, including a synopsis of the relevant aspects of the asserted prior art. His synopsis is relevant and reproduced here for reference:

A. Vallone (EX1004)¹

Vallone refers to U.S. Patent Application Publication No. 2016/0134653 to Vallone et al. It teaches a “device . . . for assessing a target network’s vulnerability to a real cyberthreat [using] synthetic tests configured to model the behavior of the

¹ EX2009 at ¶¶83–85 (*Vallone*).

cyberthreat.” EX1004 at Abstract. *Vallone* accesses cyberthreat intelligence including from the dark net, to determine cyberthreat characteristics. This includes, for example “generating a file, creating a network communication, accessing a user account/directory/database/registry, [or] a sequence of actions performed by the threat.” EX1004 at [0016]. *Vallone* does not retrieve or execute the actual exploit on its network. Instead, *Vallone* performs a “synthetic penetration test” using “agents” to simulate “the threat’s characteristics in[] the target environment,” and then “monitors target network 300 activity during [the] synthetic test” and provides “feedback data” to “increase the efficiency at which [the] target network 300 can heal any vulnerabilities to one or more cyberthreats.” EX1004 at abstract, [0016], [0019], [0039], [0040], and Fig. 6.

Vallone only performs a simulation such that “the ultimate objective of the cyberthreat . . . is not committed, while still providing an administrator with knowledge . . . regarding the depth of vulnerability of the target network 300 to an actual attack.” EX1004 at [0038].

B. *Oberheide* (EX1005)²

Oberheide refers to U.S. Patent Application Publication No. 2014/0245450 to Oberheide et al. *Oberheide* teaches a “system and method that includes identifying a vulnerability in a computing device; accessing a vulnerability exploitation mapped

² EX2009 at ¶¶86–89 (*Oberheide*).

to the identified vulnerability; at the computing device, executing the vulnerability exploitation and . . . updating the system with a vulnerability resolution.” EX1005 at Abstract, [0013], [0019], and Fig. 6.

Oberheide teaches use (i.e., exploitation) of an identified vulnerability to effectively “hack” the system to “enter[] an operating mode of escalated privileges through the identified vulnerability.” EX1005 at Fig. 2. For this purpose, *Oberheide* discloses an “exploit engine” to “execute an exploitation process” which “preferably uses at least one of the vulnerabilities identified by the VAC to grant the vulnerability tool heightened permissions within the operating system.” *Oberheide* at [0017]. *Oberheide* further teaches that it “may systematically attempt various exploits and test if the exploit succeeded, which would circumvent the process of analyzing and predicting vulnerabilities.” EX1005 at [0017].

Once in that escalated privileged mode, *Oberheide* teaches patching known vulnerabilities, including the specific exploited vulnerability. EX1005 at [0017]; [0019] (“*Oberheide* teaches exploiting a vulnerability in order to resolve or mitigate the vulnerability”); [0024] (“Step S120, which includes entering an operating mode of escalated privileges through the identified vulnerability, functions to benevolently leverage the vulnerability to enable necessary capabilities to fix or lessen the threat.”); [0025] (“The exploit preferably grants the vulnerability tool the necessary privileges to patch the vulnerability. The privileges are preferably escalated

privileges such as those for a superuser, root account, system administrator, kernel level account, higher privileged services/drivers, privileges of another user/application/service, and/or other suitable modes of increased permissions within the device. Often these are system privileges that are reserved for carriers”).

C. *Patsos* (EX1006)³

Patsos refers to Dimitrios Patsos et al., *Expanding Topological Vulnerability Analysis To Intrusion Detection Through The Incident Response Intelligence System*, Information Management & Computer Security, Vol. 18, No. 4, 291-309 (2010). *Patsos* aims to automate the incident response (IR) process through collection, normalization, and correlation of security data, i.e. vulnerability, exploit, and intrusion detection information. EX1006 at 291. To that end, *Patsos* “models the context of discovered vulnerabilities, calculates their significance, finds and analyzes potential exploit code and defines the necessary intrusion detection signatures that combat possible attacks, using standardized techniques.” *Id.*

Specifically, *Patsos* introduces the “incident response intelligence system (IRIS) that works towards the elimination of the gap between automatic vulnerability identification; exploit verification; and intrusion detection.” EX1006 at 293. “The system collects security information related to vulnerabilities, exploits and IDP [Intrusion Detection and Prevention] signatures from a large number of sources and

³ EX2009 at ¶¶90–92 (*Patsos*).

stores them in a number of local databases.” *Id.* at 295–296 (parenthetical omitted). “This vulnerability information is then normalized (i.e. redundant data are de-duplicated) and correlated with relative exploit code and IDP signatures.” *Id.* “When IRIS parses a report from a VA [Vulnerability Analysis] tool, it can construct the attack and response paths for the identified vulnerabilities . . . , score every identified vulnerability . . . , and provide the IDP signature(s) required to combat this particular attack.” *Id.*

D. *Roytman* (EX1007)⁴

Roytman refers to U.S. Patent Application Publication No. 2015/0237065 to Roytman et al. *Roytman* teaches techniques for “ranking a set of vulnerabilities of a computing asset and set of remediations for a computing asset” EX1007 at abstract. Specifically, *Roytman* receives vulnerabilities “at a vulnerability intelligence platform,” and further “breach data indicating set of breaches that occurred.” *Id.*

Roytman uses this information to identify a “subset of the set of vulnerabilities that are most vulnerable to a breach” based on the breach data; a risk for the computing asset based on its vulnerabilities;” or “multiple remediations associated with a risk score and multiple vulnerabilities,” which are ordered based on the remediations that would reduce the risk score the most. *Id.*

⁴ EX2009 at ¶¶93–95 (*Roytman*).

E. *Juels* (EX1008)⁵

Juels refers to U.S. Patent No. 9,471,777 to Juels et al. *Juels* discloses “identifying a plurality of defensive security actions to be taken to address a persistent security threat to a [computer system],” and “to determine a schedule for performance of the defensive security actions based at least in part on a selected distribution derived from a game-theoretic model.” EX1008 at abstract. “The system subject to the persistent security threat is configured to perform the defensive security actions in accordance with the schedule in order to deter the persistent security threat.” *Id.* *Juels* disclosing “optimiz[ing] defender benefit in the context of the game-theoretic model.” *Id.*

Juels’ model is based on a hypothetical “game”—called FlipIt, which is a “stealthy takeover game,” in which attacker and defender entities can each take control of a specified resource at any time by taking an action having a particular cost, but cannot determine current game state without taking the action.” EX1008 at 5:32–36. Specific details of this attacker-defender game are disclosed in EX1008 starting at 9:36.

In one example, *Jules* discloses that “the attacker might choose between two types of move: (1) Use of a published exploit or (2) Use of a zero-day exploit, while the defender chooses either to: (1) Patch a machine or (2) Reinstall its software.”

⁵ EX2009 at ¶¶96–100 (*Juels*).

EX1008 at 13:25–29. “For both players, (1) is the less expensive, but (2) the more effective.” *Id.* at 13:28–29. “Action (2) results in takeover for either player, while action (1) will only work if the opponent's most recent move was action (1).” *Id.* at 13:30–32. “For instance, patching a machine won’t recover control from a zero-day exploit, but software reinstallation will.” *Id.* at 13:32–34.

F. *Wolff* (EX1009)⁶

Wolff refers to Exhibit 1009 U.S. Patent Publication No. 2013/0198848 to Wolff. *Wolff* was specifically considered by the examiner. EX1002 at 89–90. *Wolff* discloses a computer security vulnerability remediation system (CSVRS), including a CSVRS client and a remediation server. “A RSP [Remediation Service Provider] receives information about the exploit and vulnerability from a team of remediation experts.” EX1002 at 1. “The RSP may prepare a remedial exploit, which carries a self-healing pay load.” *Id.* “The remedial exploit may be delivered either through the vulnerability itself, or through credentials granted by the CSVRS client to the RSP.” *Id.* “The self-healing pay-load takes appropriate action, such as closing ports or disabling scripts, to prevent the vulnerability from being further exploited.” *Id.* “The self-healing payload may take actions on client 120, such as disabling certain types of scripting, modifying firewall rules, disabling services, modifying registry settings, downloading programs, executing programs, downloading software

⁶ EX2009 at ¶¶101–102 (*Wolff*).

updates, installing patches, removing software, modifying privileges, adjusting file attributes, modifying configuration files, downloading antivirus or malware updates, placing selected applications in a sandbox, and upgrading system level components.” EX1002 at 2.

V. THE '385 PATENT

A. Technical Background

Dr. Medvidovic provides a helpful explanation of the technical background of the '385 patent in paragraphs 67–74 of his declaration, reproduced in full here:

67.The '385 patent teaches various system and methods for a data-driven security game framework that models an attacker based on exploit market data actively mined from the darknet to develop strategies for the defender. '385 patent at abstract.

68.Generally, the '385 patent notes that “given a system configuration (or a distribution of system configurations within an organization), the present disclosure models an attacker who, given a budget, will purchase exploits to maximize his level of access to the target system.” '385 patent at 2:63–66. “Likewise, a defender will look to adjust system configurations in an effort to minimize the effectiveness of an attacker while ensuring that necessary software dependencies are satisfied.” '385 patent at 2:67–3:3.

69.Based on this intuitive insight, the '385 patent addresses what exploits an attacker will likely use and what software in the organization poses the biggest threat. '385 patent at 1:45–49. To achieve these objectives, the '385 patent discloses a “game theoretic approach to host-based defense where the activities of the attacker are informed from an ‘un-conventional’ source (information not directly related to the defender’s system)—specifically, information from darknet markets. '385 patent at 2:26–31 and 3:6–7.

70.To accomplish this objective, the '385 patent discloses “ a security game where the attacker **102** is a malicious hacker with access to

a darknet exploit markets **104** and the defender **106** is tasked with host-based defense.” *Id.* at 3:16–19. In this framework, “the notation V is used $[\]$ to represent the entire set of vulnerabilities within a given computer system.” *Id.* at 3:20–22.

71. The ’385 patent further teaches that “in a real-world organization, system administrators are not able to patch all vulnerabilities for a variety of reasons. Software dependencies, use of legacy systems, and non-availability of patches are some examples.” *Id.* at 3:25–29.
72. To model this, a ‘constraint set’ (denoted C) is defined as a subset of V .” *Id.* at 3:29–30. This constraint set is the set of vulnerabilities that cannot be avoided, because avoiding these vulnerabilities would limit the functionality of the computer network below acceptable levels. *Id.* at 3:30–32 (“The vulnerabilities in a constraint set represent the vulnerabilities required for some system functionality.”).
73. The ’385 patent then defines “ C ” as the “set of all possible constraint sets.” *Id.* at 3:36–37. “Each constraint set in C_i , represents a set of vulnerabilities that together will provide complete functionality of application i .” *id.* at 3:39–41. If “at least one constraint set in C_i is satisfied, application i will properly operate on the system.” *Id.* at 3:43–45. Therefore, since “ C is the set of all application constraint sets for a given system configuration”, it “represents all of the applications to be run on the system.” *Id.* at 3:45–47. “In this framework, for a given system, a system administrator selects which vulnerabilities may be present in order to allow each application i to function.” *Id.* at 3:47–50. The ’385 patent teaches how the system administrator can then choose the proper constraint set that minimizes the vulnerabilities that are most likely to be attacked, by comparing the possible vulnerability constraint sets to the vulnerabilities associated with the set of exploits determined to be most likely used in an attack.
74. In summary, the ’385 patent teaches a highly innovative method of allowing a computer administrator to configure a computer system to be protected in such a way that the set of vulnerabilities exposed by the computer system has the least possible overlap with the set of vulnerabilities that an attacker is most likely to exploit.

EX2009 at ¶¶67–74.

B. Prosecution History

The provisional application that resulted in the '385 patent was filed on Nov. 30, 2015, and the corresponding non-provisional application on Nov. 28, 2016. EX1001. On October 15, 2018, an office action issued rejecting all claims in part under 35 U.S.C. § 101, and in part as obvious under 35 U.S.C. § 103 based on U.S. Patent Application Publication No. 2004/0088565 to Norman et al., in view of *Wolff* and U.S. Patent Application Publication No. 2014/0137257 to Martinez et al. EX1002 at 85.

On February 15, 2019, patent owner amended the claims, adding claim limitations including “accessing data through the network connection, the data comprising dark net information associated with exploits of a computer system;” and “applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities.” EX1002 at 118.

On March 26, 2019, the notice of allowance issued, and the examiner added an amendment for “creating a constraint set of vulnerabilities of the computer system from the set of vulnerabilities comprising a minimum set of dependencies to operate the computer system, wherein application of the set of exploits on the computer system comprises determining the effect of the set of exploits on the constraint set of vulnerabilities of the computer system.” EX1002 at 138. The examiner specifically noted that the prior art the prior art did not teach, and there was no

motivation to combine any of the prior art to disclose or suggest the amended claims.
Id. at 139.

C. Priority Date

The '385 patent derives priority from provisional application No. 62/261,200, filed on Nov. 30, 2015. The provisional application fully discloses the subject matter of the claims in the '385 patent. EX2009 at ¶76. Petitioner alleges the following priority dates for the asserted prior art:

- a. *Vallone* (EX1004): November 11, 2014 (Pet. at 5)
- b. *Oberheide* (EX1005): February 24, 2014 (Pet. at 7)
- c. *Patsos* (EX1006): October 2010 (Pet. at 9)
- d. *Roytman* (EX1007): March 9, 2015 (Pet. at 10)
- e. *Juels* (EX1008): February 24, 2012 (Pet. at 12)
- f. *Wolff* (EX1009): October 17, 2011 (Pet. at 13)

Since these dates predate the provisional patent application as well, Patent Owner's analysis relies on the specification of the '385 patent itself. However, Patent Owner is still investigating the veracity of Petitioner's alleged dates. To the extent these dates are inaccurate and in fact post-date the provisional application, Patent Owner reserves the right to identify the respective portions of the provisional application supporting the claims of the '385 patent.

VI. LEVEL OF ORDINARY SKILL IN THE ART

In consideration of the type of problems encountered in the art, the prior solutions to those problems found in the prior art references, the rapidity with which innovations are made, the sophistication of the technology, the level of education of active workers in the field, and in reliance on Dr. Medvidovic's expert opinion, Patent Owner defines the relevant skill in the art as follows: A person of ordinary skill in the art ("POSITA") at the time of the filing of the '385 patent would have at least a Bachelor's degree in computer science, computer engineering, or cybersecurity, and 2–4 years of experience working in the field of software development, distributed systems, and cybersecurity. EX2009 at ¶35. A graduate degree in computer science, computer engineering, artificial intelligence, cybersecurity, or software engineering may be substituted wholly or in part for work experience. *Id.* Dr. Medvidovic qualifies as a POSITA. *Id.* at ¶36.

VII. PATENT OWNER'S CLAIM CONSTRUCTIONS

Patent claims are construed using the framework set forth in *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–15 (Fed. Cir. 2005) (*en banc*). CrowdStrike relies on the plain and ordinary meaning of all claim terms. Pet. at 5. Patent Owner agrees and identifies what that plain and ordinary meaning for some key claim elements to avoid any ambiguities here, and within the relevant claims.

A. “constraint set”

The plain and ordinary meaning of the term “constraint set” in the context of the ’385 patent is “a set of vulnerabilities that cannot be avoided for a given computer system configuration.” This is called a “constraint” set of vulnerabilities because avoiding these vulnerabilities would limit the functionality of the computer network below acceptable levels. EX2009 at ¶72.

Generally, the ’385 patent models an attacker who, given a budget, will purchase exploits to maximize his level of access to the target system.” EX2009 at ¶68 (citing ’385 patent at 2:63–66). “Likewise, a defender will look to adjust system configurations in an effort to minimize the effectiveness of an attacker while ensuring that necessary software dependencies are satisfied.” *Id.* (citing ’385 patent at 2:67–3:3). The ’385 patent further teaches that “in a real-world organization, system administrators are not able to patch all vulnerabilities for a variety of reasons. Software dependencies, use of legacy systems, and non-availability of patches are some examples.” EX2009 at ¶71 (citing ’385 patent at 3:25–29).

In this framework, “the notation V is used [] to represent the entire set of vulnerabilities within a given computer system.” EX2009 at ¶70 (citing ’385 patent at 3:20–22). Then a “‘constraint set’ (denoted C_i) is defined as a subset of V .” EX2009 at ¶72 (citing ’385 patent at 3:29–30). This constraint set is a set of vulnerabilities that cannot be avoided for a given computer system configuration,

because avoiding these vulnerabilities would limit the functionality of the computer network below acceptable levels. *Id.* (citing '385 patent at 3:30–32 (“The vulnerabilities in a constraint set represent the vulnerabilities required for some system functionality.”)).

The '385 patent then defines “C” as the “set of all possible constraint sets.” *Id.* at 3:36–37. “Each constraint set [C_i] in C, represents a set of vulnerabilities that together will provide complete functionality of application i.” *id.* at 3:39–41. If “at least one constraint set [C_i] is satisfied, application i will properly operate on the system.” *Id.* at 3:43–45. Therefore, since “C is the set of all application constraint sets for a given system configuration”, it “represents all of the applications to be run on the system.” *Id.* at 3:45–47. “In this framework, for a given system, a system administrator selects which vulnerabilities may be present in order to allow each [desired] application i to function.” *Id.* at 3:47–50.

The '385 patent thus teaches how the system administrator can choose the proper constraint set C_i that minimizes the vulnerabilities that are most likely to be attacked, by comparing all possible vulnerability constraint sets to the vulnerabilities associated with the set of exploits determined to be most likely used in an attack, in other words, it allows the computer administrator to configure a computer system to be configured in such a way that the set of vulnerabilities exposed by the computer system has the least possible overlap with the set of vulnerabilities that an attacker

is most likely to exploit, while still achieving the minimal desired/required functionality.

Thus, a POSITA would understand plain and ordinary meaning of the term “constraint set” in the context of the ’385 patent to be “a set of vulnerabilities that cannot be avoided for a given computer system configuration.” EX2009 at ¶72.

VIII. PETITIONER HAS FAILED TO DEMONSTRATE A REASONABLE LIKELIHOOD OF PREVAILING AS TO ANY CHALLENGED CLAIM

A. GROUND 1: Obviousness in Light of *Vallone*

Petitioner asserts that claims 1, 8, 14, and 15 are obvious in light of *Vallone*. The ’385 patent differs significantly from *Vallone*. It discloses a method of altering a network configuration to vary which vulnerabilities are exposed in order to minimize the overlap with the specific vulnerabilities utilized by the most likely set of exploits. EX2009 at ¶74.

To accomplish this objective, the ’385 patent teaches the following steps:

- access dark net information (e.g.: 1.d)
- obtain a set of most likely exploits from the dark net information (e.g.: 1.e)
- determines the vulnerabilities associated with that set of exploits (e.g.: 1.f)
- determine a “constraint set,” which is a minimal set of vulnerabilities that must be accepted to operate the computer system (e.g.: 1.g)
- comparing the vulnerabilities associated with the set of exploits, with the vulnerabilities in the constraint set (e.g.: 1.h)

- determining the extent of overlap between the vulnerabilities associated with the set of exploits, and the vulnerabilities in the constraint set (e.g.: 1.i)
- altering the computer system configuration to utilize a new constraint set which has less overlap between the vulnerabilities associated with the set of exploits, and the vulnerabilities in the new constraint set (e.g.: 1.j).

EX2009 at ¶¶72–74. Although the exemplary references are to claim 1, each independent claim follows this paradigm.

1. Claim 1

Claim 1. A system for protecting a computer system from attack, the system comprising:

a) Claim 1.e: obtaining a set of exploits from the dark net information, the set of exploits configured to penetrate the computer system

Petitioner construes “obtaining a set of exploits” as “crawling the net for exploit code.” Pet. at 19. *Vallone* does not teach or suggest this limitation. EX2009 at ¶107.

Petitioner argues that “*Vallone* teaches ‘obtain[ing]’ ‘a threat characteristic,’ which ‘is a threat’s signature, action, sequence of actions or other consequence of the real threat being executed.’” Pet. at 19 (citing *Vallone* at [0016]). As explained by Dr. Medvidovic, “this disclosure does not refer to ‘exploit code,’ and *Vallone*’s discussion of ‘threat characteristics’ shows that it is referring not to the exploit itself (i.e., the source code of the exploit), but to knowledge of the ‘characteristics’ of the

exploit, including for example “a threat’s signature, action, sequence of actions or other consequence.” EX2009 at ¶108. *Vallone* discloses other examples of “threat characteristics,” such as “events (collectively referred to herein as ‘characteristics’), such as generating a file, creating a network communication, accessing a user account/directory/database/registry, a sequence of actions performed by the threat etc.” *Vallone* at [0016].

Vallone explicitly discloses that “threat characteristics” are the behavior of the threat—but not the threat itself. Dr. Medvidovic provides an example: “*Vallone* discloses a sensor environment 212 that contains ‘monitoring technology that is configured to monitor at least host 208 activity for the occurrence of characteristics of the cyberthreat.’” EX2009 at ¶109 (citing *Vallone* at [0036]). *Vallone* does not disclose executing the threat directly. EX2009 at ¶109. Indeed, the petition confirms that *Vallone* only seeks to achieve “an objective of the cyberthreat” (Pet. at 19), “which could be merely an approximation of the cyberthreat.” EX2009 at ¶109 (citing *Vallone* at [0017] (reference to “approximately achieve”)).

This is confirmed by *Vallone*’s teaching of a feedback mechanism designed to learn more about the characteristics of the threat it is analyzing. Specifically, *Vallone* discloses that “feedback allows synthetic testing for cyberthreat vulnerabilities to be specified at higher levels of aggregation when details are not known, and allows for the progressive incorporation of higher-level detailed

knowledge to allow improved resolution of modeling synthetic tests as more information is gained on the cyberthreat's objectives and characteristics, target network vulnerabilities, and other pertinent details.” *Vallone* at [0036] (emphasis added). Per Dr. Medvidovic, “this would serve no purpose if *Vallone* presumed possession of the source code of the exploit, which could simply be analyzed directly to determine the threat characteristic.” EX2009 at ¶110.

Vallone also teaches that these characteristics are “mapped to instructions via logic . . . that when executed by an agent would achieve . . . an objective of the cyberthreat.” *Vallone* at [0017] (“synthetic characteristics” are “intended to exhibit, via an agent, the real threat’s characteristics”). “As explicitly disclosed, *Vallone* does not obtain these instructions by ‘crawling the net for exploit code.’” EX2009 at ¶111. “Instead, *Vallone* creates these instructions in order to emulate the characteristics of the threat it is analyzing.” *Id.*

This is explicitly confirmed by *Vallone*:

The technology, in various embodiments, defines a sequence of one or more synthetic characteristics and/or events (‘instructions’) intended to exhibit, via an agent, the real threat’s characteristics. Characteristics are mapped to instructions via logic (e.g., logical steps in a programming language, script, program, text file, or database) that when executed by an agent would achieve or approximately achieve an objective of the cyberthreat.”

Vallone at [0017]. “*Vallone* thus discloses using its own instructions to exhibit threat characteristic that achieve or approximate the actual characteristic of the threat.” EX2009 at ¶111.

Dr. Medvidovic identifies many parts of *Vallone* that confirm that it only simulates the exploit’s characteristic of whether the computer network will recognize the threat. EX2009 at ¶112 (citing *Vallone* at [0038] (“Agents 350 are configurable by the technology to execute instructions on a host 320 for performing characteristics of a cyberthreat as a means for determining whether the target network 300 would be vulnerable to the actual cyberthreat.”); *see also*, *Vallone* at [0017] (“The agent can simulate characteristics of the cyberthreat to exercise a target network’s sensor environment’s capabilities at detecting cyberthreats within an organization’s network(s).” (emphasis added)); *Vallone* at [0019] (referring to “synthetically generated characteristics” and that “additional instructions can be initiated to carry out additional characteristics”); *Vallone* at [0038] (referring to agents that can be “configured with instructions can synthesize the characteristics of the cyberthreat.)).

Petitioner further argues that “*Vallone* receives threat intelligence from, e.g., the dark net, ‘defin[ing] a series of instructions (e.g., **a script or program**) that, when carried out by one or more agents, will exhibit the malware’s threat characteristics.’” Pet. at 20. Dr Medvidovic confirms that Petitioner

mischaracterizes this excerpt, which was taken out of context. “*Vallone* [actually] refers to a script or program created internally to simulate the characteristics of the threat.”) EX2009 at ¶113.

In an example scenario, threat intelligence is received that there is a class of malware that has the following characteristics: it creates a binary called c:/temp/malware.exe, creates a registry entry called HKLM/malware/installed, and periodically attempts to communicate to a malware.com website via TCP port 443. *The technology, in one or more embodiments, defines a series of instructions (e.g., a script or program) that, when carried out by one or more agents, will exhibit the malware’s threat characteristics.*

EX2009 at ¶113 (citing *Vallone* at [0019]) (emphasis added). *Vallone* explicitly “defines a series of instructions . . . to exhibit the malware’s threat characteristics. *Id.* Therefore, *Vallone* does not “receive” this “series of instructions” from the dark net as claimed by Petitioner. EX2009 at ¶114.

Petitioner further argues that the exploits obtained by *Vallone* are “configured to penetrate the computer system.” Pet at 20 (citing EX1003 at ¶¶111–112). But *Vallone* does not disclose obtaining either executable or source code of any exploit. Therefore, “*Vallone* cannot configure the exploit itself to ‘penetrate the computer system.’” EX2009 at ¶114. Indeed, Petitioner admits that “*Vallone*’s threat

simulation technology includes a synthetic test generator that ‘determines, for at least a portion of a cyberthreat’s objective, one or more steps of an instruction for emulating that . . . objective.’” Pet. at 20 (citing *Vallone* at [0040] (emphasis added)).

“*Vallone*’s threat simulation technology is a benign testing tool that cannot be considered to ‘penetrate the computer system’ as claimed.” EX2009 at ¶114. To the contrary, “*Vallone* explicitly discloses that its simulation stops short of actually penetrating the network being protected.” EX2009 at ¶114 (citing *Vallone* at [0038] (“The technology has the advantage of being able to synthesize the cyberthreat (e.g., harvest.pl) as executable instructions that can be limited so that the ultimate objective of the cyberthreat (e.g., posting stolen credit card numbers for sale on the Internet) is not committed.”))

For these reasons, *Vallone* does not teach or suggest this limitation.

b) Claim 1.f: applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities

Petitioner argues that *Vallone* teaches this claim element “by determining the ‘sequence of actions’ an exploit takes to penetrate the computer system.” Pet. at 20. However, Dr. Medvidovic explains that this action “pertains only to simulating some of the exploit characteristics.” EX2009 at ¶116. “*Vallone* does not disclose creating an ‘exploit function’ that ‘takes a set of exploits as input and returns a set of vulnerabilities.’” *Id.* (citing ’385 patent at 3:59–64 and claim 1). Rather:

At most, *Vallone* discloses emulating characteristics of exploits. Even if that information could be used to disclose the vulnerability targeted by that particular exploit, this would have to be done one at a time for each exploit *Vallone* analyzes. This does not teach or suggest an “exploit function” which takes an entire set of exploits and return the corresponding set of vulnerabilities.

EX2009 at ¶116.

Further, *Vallone* does not disclose a reason for using an exploit function to determining a set of vulnerabilities. As Dr. Medvidovic explains, “a POSITA implementing *Vallone* would not have been motivated to implement such an exploit function, because *Vallone* does not teach or suggest a use for such a function.”

EX2009 at ¶117.

c) Claim 1.g: creating a constraint set of the computer system from the set of vulnerabilities, the constraint set comprising a minimum set of dependencies to operate the computer system

Petitioner construes “creating a constraint set” where “the constraint set comprises a minimum set of dependencies to operate the computer system” as “an environment with enough resources to test the malicious code while preventing wider damage.” Pet at 23. *Vallone* does not teach or suggest this limitation under that construction. EX2009 at ¶118.

Petitioner cites to *Vallone*'s "synthesiz[ing] the cyberthreat (e.g., harvest.pl) as executable instructions that can be limited so that the ultimate objective of the cyberthreat . . . is not committed[.]" Pet. at 23. "But this disclosure does not relate to creating a 'constraint set' of a computer system with a minimum set of dependencies." EX2009 at ¶119. This portion of *Vallone*'s refers to "synthesizing characteristics of the threat used to attack the computer system, not the system itself." *Id.*

Petitioner further argues that "*Vallone*'s threat simulation technology's server 510 instantiates agents on host computers and instructs the agents to 'executes steps of the instructions as if the agent 350 were the actual cyberthreat (e.g., a malware agent will mimic the characteristics/actions of real malware).'" Pet. 24 (citing *Vallone* at [0038]). However, this disclosure refers to a simulated version of the exploit that operates in the actual computer system to be protected. It therefore does not teach or suggest this claim element.

Vallone tests its simulated threat in the actual environment to be tested. EX2009 at ¶120. Releasing the actual exploit in that environment is not an option because it would likely cause actual damage. *Id.* *Vallone*'s simulation is not restrained to an environment with only "enough resources to test the malicious code while preventing wider damage." *Id.*

Finally, Dr. Medvidovic provides a pertinent summary of the differences:

In summary, *Vallone* discloses, in a way, the opposite approach of the '385 patent. Instead of operating the actual exploit in a limited and safe environment, it operates a limited and safe simulation of the exploit in the actual environment. Contrary to Petitioner's statement, *Vallone* in no way "forms a controlled and isolated environment with the agents on the host computer." Pet. at 24. Instead, every aspect of *Vallone* teaches use of its agents on the live computer system to be protected. EX2009 at ¶121. Instead of executing a real exploit in a safe environment, *Vallone* discloses executing a safe simulated exploit in the real environment. Thus, *Vallone* does not teach or suggest this claim element.

d) Claim 1.h: applying the set of exploits to the constraint set of the computer system;

Petitioner cites that "once the instructions are compiled for simulating the exploit, the synthetic test is executed" as disclosing this claim element. Pet. at 25. Dr. Medvidovic identifies two reasons why this is not correct.

"First, *Vallone* discloses simulating and analyzing a single exploit at a time. Petitioner provides no support that *Vallone* creates or applies a 'set of exploits.'" EX2009 at ¶123. *Vallone* only runs "synthetic tests" that are exactly one exploit. *Id.* (citing *Vallone* at [0020] ("if analysis indicates that the target is vulnerable to a cyberattack (i.e., based on the synthetic test using the cyber-attack's characteristics)")).

Vallone also does not disclose a “constraint set of the computer system.” Petitioner construes this term as “an environment with enough resources to test the malicious code while preventing wider damage.” Pet at 23. However, as previously explained, *Vallone* does not test the malicious code directly. Instead, *Vallone* only runs a simulated synthetic test that “synthesize[s] the cyberthreat (e.g., harvest.pl) as executable instructions that can be limited so that the ultimate objective of the cyberthreat . . . is not committed, while still providing an administrator with knowledge . . . regarding the depth of vulnerability of the target network 300 to an actual attack.” EX2009 at ¶123 (citing *Vallone* at [0038]). *Vallone* also does not run this simulation in a constrained set, i.e., in a limited test environment. *Id.* “Instead, *Vallone* runs its simulation on the actual network to be protected.” *Id.* Petitioner confirms this as well, and notes that “*Vallone’s* threat simulation technology . . . [operates] by installing and instructing an agent to execute the instructions at the host computer.” Pet. at 25 (emphasis added).

e) Claim 1.i: analyzing the application of the set of exploits on the computer system to detect a particular vulnerability of the computer system; and

Vallone does not disclose this claim element because it does not disclose or suggest an “application of the set of exploits on the computer system.” See claim element 1.h. *Vallone* only analyzes the application of a simulation of a single exploit. It does not group exploits into a set, much less “apply a set of exploits.”

As Dr. Medvidovic explains, “the purpose of this claim element is to take advantage of the ‘set of all possible constraint sets.’” EX2009 at ¶126 (citing ’385 patent at 3:36–37). “By applying the set of exploits on the computer system, and then ‘determining the effect of the set of exploits on the constraint set of vulnerabilities of the computer system,’ the invention enables the system administrator to reconfigure the target computer system to utilize a different constraint set, preferably one that exhibits a lesser effect from application of the set of exploits as claimed in element 1.j.” EX2009 at ¶126. (citing ’385 patent at 3:48–50 (given “the set of all application constraint sets for a given system configuration,” “a system administrator selects which vulnerabilities may be present in order to allow each application i to function.”) and ’385 patent at 4:53–64 (discussion of the set of exploits and how they map to the vulnerabilities))). *Vallone* does not disclose such an analysis.

f) Claim 1.j: altering a configuration of the computer system in response to the analysis of the application of the set of exploits to reduce potential damage of a cyberattack.

First, *Vallone* does not disclose an “application of the set of exploits” because it only analyzes the application of a simulation of a single exploit. See claim element 1.h.

As further explained by Dr. Medvidovic, “‘altering a configuration of the computer system’ refers to enabling the system administrator to select a different

constraint set of vulnerabilities to minimize the impact of the set of exploits.” EX2009 at ¶127 (citing ’385 patent at 3:48–50 (“a system administrator selects which vulnerabilities may be present in order to allow each application i to function”)). “This is different from the kinds of updates *Vallone* teaches (e.g., updates to network monitoring, firmware, etc.).” *Id.* (citing *Vallone* at [0020]).

2. Claim 8

Claim 8. A method for improving a computing device, the method comprising:

a) Claim 8.b: obtaining a set of exploits from the dark net information, the set of exploits configured to bypass a security feature of the computer system

Petitioner references its analysis of claim element 1.e for this claim element. Patent Owner therefore incorporates its analysis of that claim element here. For the reasons stated there, *Vallone* does not disclose “obtaining a set of exploits.”

b) Claim 8.c: applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities

Petitioner references its analysis of claim element 1.f for this claim element. Patent Owner therefore incorporates its analysis of that claim element here. For the reasons stated there, *Vallone* does not disclose this claim element.

c) Claim 8.d: creating a constraint set of vulnerabilities of the computer system from the set of vulnerabilities comprising a minimum set of dependencies to operate the computer system, wherein application of the set of exploits on the computer system comprises determining the effect of the set of exploits on the constraint set of vulnerabilities of the computer system

Petitioner references its analysis of claim elements 1.g and 1.h for this claim element. Patent Owner therefore incorporates its analysis of that claim element here. For the reasons stated there, *Vallone* does not disclose this claim element.

Notably, this claim element differs from claim elements 1.g and 1.f in its reference to “determining the effect of the set of exploits on the constraint set of vulnerabilities.” This is not discussed by Petitioner, and is not disclosed by *Vallone*. *Vallone* does not disclose applying a “set of exploits” to determine its effect on the “constraint set of vulnerabilities.”

As explained by Dr. Medvidovic, “the purpose of this claim element is to take advantage of the ‘set of all possible constraint sets.’” EX2009 at ¶131 (citing ’385 patent at 3:36–37). “By applying the set of exploits on the computer system, and then ‘determining the effect of the set of exploits on the constraint set of vulnerabilities of the computer system,’ the invention enables the system administrator to reconfigure the target computer system to utilize a different constraint set, preferably one that exhibits a lesser effect from application of the set of exploits.” *Id.* (citing ’385 patent at 3:48–50 (given “the set of all application constraint sets for a given system configuration,” “a system administrator selects

which vulnerabilities may be present in order to allow each application i to function.”) and 4:53–64 (discussion of the set of exploits and how they map to the vulnerabilities)).

d) Claim 8.e: analyzing an application associated with the set of exploits on the computer system to detect a particular vulnerability of the constraint set of vulnerabilities of the computer system

Petitioner references its analysis of claim element 1.i for this claim element. Patent Owner therefore incorporates its analysis of that claim element here. For the reasons stated there, *Vallone* does not disclose this claim element.

Petitioner contends that *Vallone*’s analysis of the simulation of a single exploit as disclosing this claim element. Pet. at 30. But *Vallone* does not disclose “analyzing an application associated with the set of exploits on the computer system.” EX2009 at ¶133. Dr. Medvidovic explains:

The claimed application of a set of exploits in this claim element is not aimed at evaluating the effect of a single exploit. Instead, it is aimed evaluating a given constraint set of vulnerabilities of the computer system. The purpose of this evaluation is to enable the system administrator to reconfigure the target computer system to utilize a different constraint set, preferably one that exhibits a lesser effect from application of the set of exploits, as specifically claimed in claim element 8.f.

EX2009 at ¶133 (citing '385 patent at 3:48–50 (given “the set of all application constraint sets for a given system configuration,” “a system administrator selects which vulnerabilities may be present in order to allow each application i to function.”) and '385 patent at 4:53–64 (discussion of the set of exploits and how they map to the vulnerabilities))).

e) Claim 8.f: altering a configuration of the computer system in response to the analysis of the application of the set of exploits to reduce potential damage of a cyberattack

Petitioner references its analysis of claim element 1.j for this claim element. Patent Owner therefore incorporates its analysis of that claim element here. For the reasons stated there, *Vallone* does not disclose this claim element. Further *Vallone* does not disclose “altering a configuration of the computer system,” i.e., enabling the system administrator to select a different constraint set of vulnerabilities to minimize the impact of the set of exploits. *See* claim element 1.i.

3. Claim 14

Claim 14: The method of claim 8, wherein altering the configuration of the computer system comprises installing a patch to a software component of the computer system that is exposed by the set of exploits to attack the computer system.

Claim 14 depends on Claim 8 and is not disclosed by *Vallone* for the same reasons as stated in regard to Claim 8.

4. Claim 15

Claim 15. One or more non-transitory tangible computer-readable storage media storing computer-executable instructions for performing a computer process on a machine, the computer process comprising:

a) Claim 15.a: obtaining a set of exploits from dark net information, the set of exploits configured to bypass a security feature of a computer system

Petitioner references its analysis of claim elements 1.e and 8.b for this claim element. Pet. at 32. Patent Owner therefore incorporates its analysis of those claim elements here. For the reasons stated there, *Vallone* does not disclose “obtaining a set of exploits.”

b) Claim 15.b: applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities

Petitioner references its analysis of claim elements 1.f and 8.c for this claim element. Pet. at 32. Patent Owner therefore incorporates its analysis of those claim elements here. For the reasons stated there, *Vallone* does not disclose this claim element.

c) Claim 15.c: creating a constraint set of vulnerabilities of the computer system from the set of vulnerabilities comprising a minimum set of dependencies to operate the computer system;

Petitioner references its analysis of claim elements 1.g and 8.d for this claim element. Pet. at 33. Patent Owner therefore incorporates its analysis of those claim elements here. For the reasons stated there, *Vallone* does not disclose this claim element.

d) Claim 15.d: applying the set of exploits to the constraint set of vulnerabilities of the computer system;

Petitioner references its analysis of claim element 1.h for this claim element. Pet. at 33. Patent Owner therefore incorporates its analysis of that claim element here. For the reasons stated there, *Vallone* does not disclose this claim element.

e) Claim 15.e: analyzing the application of the set of exploits on the computer system to detect a particular vulnerability of the computer system; and

Petitioner references its analysis of claim elements 1.i and 8.e for this claim element. Pet. at 33. Patent Owner therefore incorporates its analysis of those claim elements here. For the reasons stated there, *Vallone* does not disclose this claim element.

f) Claim 15.f: altering a configuration of the computer system in response to the analysis of the application of the set of exploits to reduce potential damage of a cyberattack.

Petitioner references its analysis of claim elements 1.j and 8.f for this claim element. Pet. at 33. Patent Owner therefore incorporates its analysis of those claim elements here. For the reasons stated there, *Vallone* does not disclose this claim element.

B. GROUND 2: Obviousness in Light of *Oberheide*, *Patsos*, and *Roytman*

Petitioner asserts that claims 1, 8, 14, and 15 are obvious in light of the combination of *Oberheide*, *Patsos*, and *Roytman*.

1. Claim 1

Claim 1. A system for protecting a computer system from attack, the system comprising:

a) Claim 1.g: creating a constraint set of the computer system from the set of vulnerabilities, the constraint set comprising a minimum set of dependencies to operate the computer system

In the context of ground 2, Petitioner defines a “constraint set of vulnerabilities” as constraining a “vulnerability analysis to specific vulnerabilities based on: (1) vulnerabilities associated with a specific component of the system; or (2) system capabilities after exploitation of vulnerability.” Pet. at 41. While this is based on *Oberheide*’s disclosure, it is not supported by it. *Id.* “*Oberheide* does not use any variation of the word stem ‘constrain.’” *Id.*

Notably, Petitioner’s construction here differs from Petitioner’s construction of the same term in ground 1, where Petitioner construes “creating a constraint set” where “the constraint set comprises a minimum set of dependencies to operate the computer system” as “an environment with enough resources to test the malicious code while preventing wider damage.” Pet at 23.

Using its construction for ground 2, Petitioner notes that “while *Oberheide* does not expressly discuss how it constrains its analysis to these specific vulnerabilities, *Roytman* teaches a ranking technique to identify a subset of vulnerabilities most vulnerable to exploitation.” Pet. at 41. Petitioner continues that

“*Oberheide* constrains its vulnerability analysis to vulnerabilities associated with a particular computer component or vulnerabilities that significantly alter the computer’s capabilities after exploitation.” Pet. at 42. Petitioner concludes that “*Oberheide* therefore teaches or suggests creating a constraint set comprising “a minimum set of dependencies to operate the computer system” because its “vulnerability analysis may be constrained to vulnerabilities that shut down the computer system through successful exploitation and/or to vulnerabilities that render facets of the computer system inoperable.”

Dr. Medvidovic disagrees for multiple reasons. “First, neither *Oberheide* nor *Patsos* disclose a ‘minimum set of dependencies to operate the computer system.’” EX2009 at ¶145. In support, Dr. Medvidovic notes that “the ’385 patent specifically recognizes that ‘in a real-world organization, system administrators are not able to patch all vulnerabilities for a variety of reasons. Software dependencies, use of legacy systems, and non-availability of patches are some examples.’ *Id.* (citing ’385 patent at 3:25–29).

To model these dependencies, the ’385 patent first introduces the notation “V” “to represent the entire set of vulnerabilities within a given computer system.” ’385 patent at 3:20–22. Then, a “‘constraint set’ (denoted C) is defined as a subset of V.” *Id.* at 3:29–30. “The vulnerabilities in a constraint set represent the vulnerabilities required

for some system functionality.” *Id.* at 3:30–32. Thus, the claimed “constrained set” comprising the “minimum set of dependencies to operate the computer system” is the set of vulnerabilities that cannot be avoided, because avoiding these vulnerabilities would limit the functionality of the computer network below acceptable levels. This is useful to prioritize the exploits of those particular vulnerabilities (which you cannot avoid) you will need to guard against.

EX2009 at ¶146.

Oberheide teaches the exact opposite of this limitation. The claimed constraint set consisting vulnerabilities that are required to maintain functionality (i.e., vulnerabilities the administrator must leave unpatched in order to keep necessary features operating). *Id.* In stark contrast, *Oberheide* constrains its analysis to vulnerabilities that, if exploited, significantly impair or break system functionality. *Id.* at ¶144 (citing Pet. at 42). A reference that directs a POSITA to focus on vulnerabilities that disable the system cannot teach or suggest identifying the claimed “constrained set” that are necessary to keep the system operating at an acceptable level. *Id.* at ¶147.

To meet this “constrained set” limitation, Petitioner cites to *Oberheide*’s disclosure of constraining a “vulnerability analysis to specific vulnerabilities based on: (1) vulnerabilities associated with a specific component of the system; or (2)

system capabilities after exploitation of vulnerability.” Pet. at 41. As explained above, this is the opposite of the claimed invention and therefore does not disclose or suggest this claim element. “Neither *Oberheide* nor *Patsos* determine the minimum set of dependencies to operate the computer system it seeks to protect, nor creating a set of vulnerabilities based thereon as claimed.” EX2009 at ¶147.

Finally, Petitioner’s citation to *Roytman* as disclosing ranking vulnerabilities (Pet. at 42) is irrelevant to this claim element. “The claimed constraint set is not based on a ranking of vulnerabilities, but on the minimal set of dependencies to operate the computer system.” EX2009 at ¶148. “*Roytman* does not determine this minimal set of dependencies.” *Id.*

b) Claim 1.h: applying the set of exploits to the constraint set of the computer system;

Petitioner argues that this claim element is met because “*Oberheide*’s system applies exploits to the identified subset of vulnerabilities to determine which exploit will provide the necessary privileges for patching.” Pet. at 46. This is wrong for multiple reasons.

First, *Oberheide* does not disclose a “constraint set of the computer system” as explained above. See claim element 1.g.

Second, *Oberheide* does not disclose a “set of exploits.” “While *Oberheide* discusses various exploits, it does not disclose any metrics for grouping these into a set.” EX2009 at ¶150. “The ’385 patent, on the other hand, discloses determining

a set of exploits as a subset of all available exploits based on the attacker's budget. *Id.* (citing '385 patent at 4:7–8 (“An attacker 102 will use a set of exploits to attempt to gain access to a system, and will do so within a budget.”)).

Third, the combination of *Oberheide*, *Patsos*, and *Roytman* does not “apply” a set of exploits to a constraint set of vulnerabilities. Petitioner defines “constraint set of vulnerabilities” as constraining a “vulnerability analysis to specific vulnerabilities based on: (1) vulnerabilities associated with a specific component of the system; or (2) system capabilities after exploitation of vulnerability.” Pet. at 41. But “the combination of *Oberheide*, *Patsos*, and *Roytman* does not ‘apply’ a set of exploits to Petitioner’s ‘constraint set of vulnerabilities.’” EX2009 at ¶151. “In the '385 patent, this operation is defined as comparing the vulnerabilities associated with the set of exploits to the vulnerabilities in the constraint set of vulnerabilities.” *Id.* (citing '385 patent at 4:53–55 (“if the vulnerabilities generated by an attack strategy’s exploits and the vulnerabilities in a defense strategy are disjoint sets, the payoff function must return 0”)). Petitioner’s combination does not disclosed this functionality.

Further, Dr. Medvidovic explains that “based on the information gathered from ‘applying the set of exploits to the constraint set of the computer system,’ the '385 patent teaches that the system administrator can then choose the proper constraint set that minimizes the vulnerabilities that are most likely to be attacked,

by comparing the possible vulnerability constraint sets to the vulnerabilities associated with the set of exploits determined to be most likely used in an attack.” EX2009 at ¶152. This is captured in the last two claim elements of analyzing the information (1.i), and altering the computer system accordingly to utilize a more optimal constraint set of vulnerabilities (1.j). *Id.*

c) Claim 1.i: analyzing the application of the set of exploits on the computer system to detect a particular vulnerability of the computer system; and

As explained above, the combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose an “application of the set of exploits on the computer system.” See claim element 1.h.

Further, “the purpose of this claim element is to take advantage of the ‘set of all possible constraint sets.’” EX2009 at ¶153 (citing ’385 patent at 3:36–37). In other words, the invention enables the system administrator to reconfigure the target computer system to utilize a different constraint set, preferably one that exhibits a lesser effect from application of the set of exploits as claimed in element 1.j. *Id.* (citing ’385 patent at 3:48–50 (given “the set of all application constraint sets for a given system configuration,” “a system administrator selects which vulnerabilities may be present in order to allow each application i to function.”) and ’385 patent at 4:53–64 (discussion of the set of exploits and how they map to the vulnerabilities)). This is accomplished by applying the set of exploits on the computer system, and

then “determining the effect of the set of exploits on the constraint set of vulnerabilities of the computer system.” The prior art combination does not disclose this functionality.

Petitioner argues that “*Oberheide* exploits the vulnerabilities in its constraint set to gain access to privileges necessary to patch those vulnerabilities” discloses this claim element. Pet. at 48. Specifically, Petitioner states that the combination system would “identify the highest ranked vulnerability to patch.” *Id.* That functionality does not disclose this claim element. Instead, based on the ’385 patent, “the purpose of the claimed analysis is to determine the overlap between the vulnerabilities associated with the set of exploits to the constraint set of vulnerabilities.” EX2009 at ¶154. “This information then allows the system administrator to choose the proper constraint set that minimizes the overlap of vulnerabilities.” *Id.* (emphasis added).

d) Claim 1.j: altering a configuration of the computer system in response to the analysis of the application of the set of exploits to reduce potential damage of a cyberattack.

As explained above, the combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose an “application of the set of exploits.” See claim element 1.h. As further explained above, “altering a configuration of the computer system” refers to enabling the system administrator to select a different constraint set of vulnerabilities to minimize the impact of the set of exploits. See claim element 1.i.

Petitioner identifies the “‘update component’ that applies ‘a patch or system update to resolve or reduce an existing vulnerability’” as disclosing this claim element because that operation “alters a configuration of the computer system.” Pet. at 49. This is incorrect. As explained regarding claim element 1.i, the ’385 patent identifies “altering a configuration of the computer system” as the system administrator selecting a different constraint set of vulnerabilities to minimize the impact of the most likely set of exploits.

2. Claim 8

Claim 8. A method for improving a computing device, the method comprising:

a) Claim 8.b: obtaining a set of exploits from the dark net information, the set of exploits configured to bypass a security feature of the computer system

Petitioner references its analysis of claim element 1.e for this claim element.

Patent Owner therefore incorporates its analysis of that claim element here.

b) Claim 8.c: applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities

Petitioner references its analysis of claim element 1.f for this claim element.

Patent Owner therefore incorporates its analysis of that claim element here. For the reasons stated there, the combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose this claim element.

c) Claim 8.d: creating a constraint set of vulnerabilities of the computer system from the set of vulnerabilities comprising a minimum set of dependencies to operate the computer system, wherein application of the set of exploits on the computer system comprises determining the effect of the set of exploits on the constraint set of vulnerabilities of the computer system

Petitioner references its analysis of claim elements 1.g and 1.h for this claim element. Patent Owner therefore incorporates its analysis of those claim elements here. For the reasons stated there, the combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose this claim element.

Notably, this claim element differs from claim elements 1.g and 1.f by the limitation of “determining the effect of the set of exploits on the constraint set of vulnerabilities.” This is not discussed by Petitioner, and is not disclosed by the combination of *Oberheide*, *Patsos*, and *Roytman*. “The combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose applying a ‘set of exploits’ to determine its effect on the ‘constraint set of vulnerabilities.’” EX2009 at ¶160.

As Dr. Medvidovic explains, “the purpose of this claim element is to take advantage of the ‘set of all possible constraint sets.’” EX2009 at ¶161 (citing ’385 patent at 3:36–37). “By applying the set of exploits on the computer system, and then ‘determining the effect of the set of exploits on the constraint set of vulnerabilities of the computer system,’ the invention enables the system administrator to reconfigure the target computer system to utilize a different constraint set, preferably one that exhibits a lesser effect from application of the set

of exploits.” *Id.* (citing ’385 patent at 3:48–50 (given “the set of all application constraint sets for a given system configuration,” “a system administrator selects which vulnerabilities may be present in order to allow each application i to function.”) and ’385 patent at 4:53–64 (discussion of the set of exploits and how they map to the vulnerabilities))).

d) Claim 8.e: analyzing an application associated with the set of exploits on the computer system to detect a particular vulnerability of the constraint set of vulnerabilities of the computer system

Petitioner references its analysis of claim element 1.i for this claim element. Patent Owner therefore incorporates its analysis of that claim element here. For the reasons stated there, the combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose this claim element.

e) Claim 8.f: altering a configuration of the computer system in response to the analysis of the application of the set of exploits to reduce potential damage of a cyberattack

Petitioner references its analysis of claim element 1.j for this claim element. Patent Owner therefore incorporates its analysis of that claim element here. For the reasons stated there, the combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose this claim element. Further the combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose “altering a configuration of the computer system,” i.e., enabling the system administrator to select a different constraint set of vulnerabilities to minimize the impact of the set of exploits. *See* claim element 1.i.

3. Claim 14

Claim 14: The method of claim 8, wherein altering the configuration of the computer system comprises installing a patch to a software component of the computer system that is exposed by the set of exploits to attack the computer system.

Claim 14 depends on Claim 8 and is not disclosed by the combination of *Oberheide*, *Patsos*, and *Roytman* for the same reasons as stated in regard to Claim 8.

4. Claim 15

Claim 15. One or more non-transitory tangible computer-readable storage media storing computer-executable instructions for performing a computer process on a machine, the computer process comprising:

a) Claim 15.a: obtaining a set of exploits from dark net information, the set of exploits configured to bypass a security feature of a computer system

Petitioner references its analysis of claim elements 1.e and 8.b for this claim element. Pet. at 53. Patent Owner therefore incorporates its analysis of those claim elements here. For the reasons stated there, the combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose “obtaining a set of exploits.”

b) Claim 15.b: applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities

Petitioner references its analysis of claim elements 1.f and 8.c for this claim element. Pet. at 53. Patent Owner therefore incorporates its analysis of those claim

elements here. For the reasons stated there, the combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose this claim element.

c) Claim 15.c: creating a constraint set of vulnerabilities of the computer system from the set of vulnerabilities comprising a minimum set of dependencies to operate the computer system;

Petitioner references its analysis of claim elements 1.g and 8.d for this claim element. Pet. at 53. Patent Owner therefore incorporates its analysis of those claim elements here. For the reasons stated there, the combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose this claim element.

d) Claim 15.d: applying the set of exploits to the constraint set of vulnerabilities of the computer system;

Petitioner references its analysis of claim element 1.h for this claim element. Pet. at 53 Patent Owner therefore incorporates its analysis of that claim element here. For the reasons stated there, the combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose this claim element.

e) Claim 15.e: analyzing the application of the set of exploits on the computer system to detect a particular vulnerability of the computer system; and

Petitioner references its analysis of claim elements 1.i and 8.e for this claim element. Pet. at 54. Patent Owner therefore incorporates its analysis of those claim elements here. For the reasons stated there, the combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose this claim element.

f) Claim 15.f: altering a configuration of the computer system in response to the analysis of the application of the set of exploits to reduce potential damage of a cyberattack.

Petitioner references its analysis of claim elements 1.j and 8.f for this claim element. Pet. at 54. Patent Owner therefore incorporates its analysis of those claim elements here. For the reasons stated there, the combination of *Oberheide*, *Patsos*, and *Roytman* does not disclose this claim element.

5. Motivation To Combine

A POSITA would not be motivated to combine *Oberheide* with *Roytman*. EX2009 at ¶171. “*Oberheide* focuses on identifying and exploiting vulnerabilities that will give it elevated access privileges in order to alter the system configuration to apply its own patches.” *Id.* “A ranking of vulnerabilities is not necessary to achieve this goal.” *Id.* This is corroborated by *Oberheide*, which explicitly references vulnerability sets (i.e., unordered collections) instead of ranked lists as it allows system users or administrators to explore, visualize, and decide on remedying them based on their preference. *Id.* (citing *Oberheide* at [0020], [0022], and [0029]). Therefore, a “POSITA would not be motivated to incur the extra expense and complexity of implementing *Roytman* since it would provide no benefit to *Oberheide*’s system.” *Id.*

C. GROUND 3: Obviousness in Light of *Oberheide*, *Patsos*, *Roytman*, and *Juels*

Petitioner asserts that claims 2–6, 9–12, 16, and 17 are obvious in light of the combination of *Oberheide*, *Patsos*, *Roytman*, and *Juels*.

1. Claim 2

Claim 2. The system of claim 1, wherein analyzing the application of the set of exploits on the computer system comprises quantifying a risk level posed to the computer system related to a cost to obtain the set of exploits.

Claim 2 depends on claim 1 and is not disclosed for the same reasons as stated in regard to claim 1.

Petitioner notes that the combination of *Oberheide*, *Patsos*, and *Roytman* “does not expressly quantify risk level related to a cost to obtain exploits.” Pet. at 54–55. Petitioner then cites *Juels* for the disclosure of “a cost metric for exploits.” Pet. at 55.

Juels does not disclose this claim element because its consideration of costs is limited to a single “move” in its game. EX2009 at ¶174 (citing *Juels* at 13:25–29). Petitioner cites no support that *Juels* would motivate a POSITA to expand this analysis to “a set of exploits.” *Id.* “To the contrary, *Juels* discloses that players ‘choose among a set of actions with varying costs and effectiveness.’” EX2009 at ¶174. As Dr Medvidovic explains, “this disclosure would not encourage the consideration of the cost of a whole set of exploits, because *Juels* system teaches

making a single ‘move’ at a time.” EX2009 at ¶174 (citing *Juels* at 1:20–22 (“attackers orchestrating an APT typically introduce periods of delay among different stages of the attack”); and 12:3–5 (*Juels* discourages sets of ‘moves’ because “there is always a cost to pushing the button [i.e., making a “move]. In this example, pushing the button costs the equivalent of one second of ownership.”)).

2. Claim 3

Claim 3. The system of claim 2, wherein the processing device further performs the operation of: aligning Internet data with the constraint set of vulnerabilities based on information obtained about the set of exploits.

Claim 3 depends on Claim 2 and is not disclosed for the same reasons as stated in regard to Claim 2.

Petitioner argues that the combination of *Oberheide*, *Patsos*, and *Roytman* teaches creating a “constraint set” by constraining its vulnerability analysis to specific vulnerabilities, including by ranking vulnerabilities, as taught by *Roytman*, to identify a subset of vulnerabilities most vulnerable to exploitation.” Pet. at 56. This is incorrect. “The ’385 patent discloses the constraint set as the set of vulnerabilities that cannot be avoided, because avoiding these vulnerabilities would limit the functionality of the computer network below acceptable levels. EX2009 at ¶176 (citing ’385 patent at 3:30–32 (“The vulnerabilities in a constraint set represent the vulnerabilities required for some system functionality.”) and 3:45–50 (“C is the set of all application constraint sets for a given system configuration and represents

all of the applications to be run on the system. So, in this framework, for a given system, a system administrator selects which vulnerabilities may be present in order to allow each application i to function.”)) Petitioner identifies a different set and functionality, i.e., ranking of the vulnerabilities to identify “a subset of vulnerabilities most vulnerable to exploitation.” Pet. at 56.

3. Claim 4

Claim 4. The system of claim 3, wherein obtaining a set of exploits through the network connection comprises: applying one or more algorithms to obtain the set of exploits.

Claim 4 depends on Claim 3 and is not disclosed for the same reasons as stated in regard to Claim 3.

4. Claim 5

Claim 5. The system of claim 4, wherein the one or more algorithms determine the cost to obtain the set of exploits to create a budget and obtains the set of exploits constrained by the determined cost to obtain the set of exploits.

Claim 5 depends on Claim 4 and is not disclosed for the same reasons as stated in regard to Claim 4.

Petitioner argues that “*Oberheide* would have similarly benefitted from buying exploits based on collected cost information to update its repository and creating a budget to regulate this system.” Pet. at 58. This is not supported by *Oberheide*. “*Oberheide* aims to exploit a vulnerability to gain escalated privilege access to a system, including for example by ‘systematically attempt[ing] various

exploits and test[ing] if the exploit succeeded.” EX2009 at ¶179 (citing *Oberheide* at [0017]). Once a suitable exploit is identified, “*Oberheide* would have established the desired access to the system, and would have no need to consider a ‘budget’ for additional exploits.” EX2009 at ¶179. “Further, *Oberheide* requires finding a single vulnerability to exploit, and therefore does not consider a ‘set of exploits.’” *Id.*

Finally, Dr. Medvidovic explains that *Oberheide* explicitly allows sets of vulnerabilities to be detected and potentially patched in a variety of scenarios involving different temporal relationships between the vulnerabilities:

For example, a first vulnerability is detected at time t1; the first vulnerability is patched and a reference monitor is set up; a second vulnerability is detected at time t2, where t2 comes after t1; the established reference monitor is used to invoke heightened privileges without executing an exploit; and the second vulnerability is patched. The detection of the second vulnerability can happen for a vulnerability of a set of vulnerabilities detected with the first vulnerability, during a second local scan of new data objects of the device, scanning for latent vulnerability in data object records using new vulnerability information, or in response to any patching prompt.

Oberheide at [0029]. “Thus, *Oberheide* teaches addressing and patching vulnerabilities one at a time as they are detected, and does not disclose or suggest the cost for a set of exploits.” EX2009 at ¶180.

5. Claim 6

Claim 6. The system of claim 5, wherein obtaining the set of exploits is based on maximizing the quantifying of the risk level posed to the computer system related to the cost to obtain the set of exploits.

Claim 6 depends on claim 5 and is not disclosed for the same reasons as stated in regard to claim 5.

Petitioner relies on a combination of *Roytman*’s risk assessment with *Juels*’ cost of exploits for this claim element. Pet. at 60. This is not correct.

“*Juels* consideration of costs is limited to a single ‘move’ in its game.” EX2009 at ¶182 (citing *Juels* at 13:25–29) (*see also*, claim 2 above). “*Juels* provides nothing that would motivate that it expands this analysis to ‘a set of exploits.’” EX2009 at ¶182. “To the contrary, *Juels* discloses that players ‘choose among a set of actions with varying costs and effectiveness.’” *Id.* “This disclosure would not encourage the consideration of the cost of a whole set of exploits, because *Juels* system teaches making a single ‘move’ at a time.” *Id.* (citing *Juels* at 1:20–22 and 12:3–5).

6. Claim 9

Claim 9. The method of claim 8, wherein analyzing the application of the set of exploits on the computer system comprises

quantifying a risk level posed to the computer system related to a cost to obtain the set of exploits.

Claim 9 depends on claim 8 and is not disclosed for the same reasons as stated in regard to claim 8.

Petitioner references its analysis of claim 2 for this claim. Patent Owner therefore incorporates its response thereto here.

7. Claim 10

Claim 10. The method of claim 9, wherein obtaining a set of exploits comprises: applying one or more algorithms to obtain the set of exploits based on the constraint set of vulnerabilities of the computer system from one or more darknet marketplaces on the Internet.

Claim 10 depends on Claim 9 and is not disclosed for the same reasons as stated in regard to Claim 9.

Petitioner references its analysis of claim 4 for this claim. Patent Owner therefore incorporates its response thereto here.

8. Claim 11

Claim 11. The method of claim 10, wherein the one or more algorithms determine the cost to obtain the set of exploits to create a budget and obtains the set of exploits constrained by the determined cost to obtain the set of exploits.

Claim 11 depends on claim 10 and is not disclosed for the same reasons as stated in regard to claim 10.

Petitioner references its analysis of claim 5 for this claim. Patent Owner therefore incorporates its response thereto here.

9. Claim 12

Claim 12. The method of claim 11, wherein obtaining the set of exploits is based on maximizing the quantifying of the risk level posed to the computer system related to the cost to obtain the set of exploits.

Claim 12 depends on claim 11 and is not disclosed for the same reasons as stated in regard to claim 11.

Petitioner references its analysis of claim 6 for this claim. Patent Owner therefore incorporates its response thereto here.

10. Claim 16

Claim 16. The one or more non-transitory computer-readable storage media of claim 15, wherein analyzing the application of the set of exploits on the computer system comprises quantifying a risk level posed to the computer system related to a cost to obtain the set of exploits.

Claim 16 depends on claim 15 and is not disclosed for the same reasons as stated in regard to claim 15.

Petitioner references its analysis of claim 2 for this claim. Patent Owner therefore incorporates its response thereto here.

11. Claim 17

Claim 17. The one or more non-transitory computer-readable storage media of claim 15, wherein obtaining a set of exploits comprises applying one or more algorithms to obtain the set of exploits from one or more darknet marketplaces on the Internet.

Claim 17 depends on claim 15 and is not disclosed for the same reasons as stated in regard to claim 15.

Petitioner references its analysis of claims 1.e, 4, and 10 for this claim. Patent Owner therefore incorporates its response thereto here.

12. Motivation to Combine

A POSITA would have had no motivation to combine *Oberheide* with *Roytman* as detailed above in ground 2.

Similarly, “a POSITA would have had no motivation to combine *Oberheide* with *Juels*.” EX2009 at ¶195. “*Oberheide* focuses on identifying and exploiting vulnerabilities that will give it elevated access privileges in order to alter the system configuration to apply its own patches.” *Id.* “A determination of the cost of exploits is not relevant to this analysis.” *Id.* “A POSITA would not be motivated to incur the extra expense and complexity of implementing *Juels* since it would provide no benefit to *Oberheide*’s system.” *Id.* “*Juels*’ game-theoretic approach (called FlipIt) may actually conflict with the desired result of *Oberheide*, since *Juels* considers the cost of a single exploit based on acquisition (*Juels* at 13:25–29) and does not consider *Oberheide*’s requirement that the exploit allow escalated privileges.” *Id.* (citing *Oberheide* at Fig. 2 and [0025]).

D. GROUND 4: Obviousness in Light of *Oberheide*, *Patsos*, *Roytman*, and *Wolff*

Petitioner asserts that claims 7 and 13 are obvious in light of *Oberheide*, *Patsos*, *Roytman*, and *Wolff*. Notably, *Wolff* was specifically considered by the examiner during prosecution. EX1002 at 89–90.

1. Claim 7

Claim 7. The system of claim 1, wherein altering the configuration of the computer system comprises removing a software component of the computer system that is utilized by the set of exploits to attack the computer system.

Claim 7 depends on claim 1 and is not disclosed for the same reasons as stated in regard to Claim 1.

2. Claim 13

Claim 13. The method of claim 8, wherein altering the configuration of the computer system comprises removing a software component of the computer system that is utilized by the set of exploits to attack the computer system.

Claim 13 depends on claim 8 and is not disclosed for the same reasons as stated in regard to claim 8.

3. Motivation to Combine

A POSITA would have had no motivation to combine *Oberheide* with *Roytman* as detailed above in ground 2. Similarly, a POSITA would have had no motivation to combine *Oberheide* with *Wolff*. EX2009 at ¶201. “*Oberheide* focuses on identifying and exploiting vulnerabilities that will give it elevated access privileges in order to alter the system configuration to apply its own patches.” *Id.* “The removal of software exhibiting certain vulnerabilities would be contrary to *Oberheide*’s goal of using the vulnerability to enter an operating mode of escalated privileges through the identified vulnerability.” *Id.* (citing *Oberheide* at Fig. 2). “*Oberheide* would exclude its own escalated system access by removing the

vulnerability it used to access the system.” *Id.* “A POSITA would not be motivated to incur the extra expense and complexity of implementing *Juels* since it would provide no benefit to *Oberheide’s* system.” *Id.*

IX. SECONDARY CONSIDERATIONS OF NON-OBVIOUSNESS

Certain secondary considerations of non-obviousness support the ’385 patent. These must be evaluated in assessing obviousness, to provide real-world insight into the novelty of the invention. *Graham v. John Deere Co. of Kan. City*, 383 U.S. 1, 17–18 (1966). These important factors counter hindsight bias and are particularly important in close cases. *Id.*

A. Long-Felt Need and Government Support

The ’385 patent addressed a long-felt need for effective prediction of cyber exploits in a rapidly evolving threat landscape, critical to U.S. national security. EX2011 at ¶6. For that reason, Dr. Shakarian’s research was funded by competitive government grants totaling over \$7 million, including from DARPA, IARPA, and ONR, focused on mission-critical cybersecurity. EX2011 at ¶4. This support highlights the invention’s role in solving persistent challenges in threat prediction, where prior approaches lacked the integration of dark web data and machine learning for accurate prediction.

B. Praise By Others

The fact that Dr. Shakarian's research that led to the '385 patent was repeatedly funded by competitive government grants totaling over \$7 million, including from DARPA⁷, IARPA⁸, and ONR⁹, supports that industry experts acknowledged the significance of the invention. EX2002 at ¶4.

C. Copying By Competitors

Skysong's suits against the CrowdStrike and Fortinet confirm that competitors have knowingly copied the invention, which suggests they recognized its value and that it was not obvious to them.

X. CONCLUSION

The Board should deny institution because CrowdStrike has failed to demonstrate a reasonable likelihood of prevailing as to any challenged claim.

⁷ Defense Advanced Research Projects Agency, U.S. Department of Defense agency that funds national security technology.

⁸ Intelligence Advanced Research Projects Activity, R&D arm of U.S. Director of National Intelligence.

⁹ Office of Naval Research, U.S. Navy and Marine Corps' science and technology programs.

Dated: November 19, 2025

Respectfully submitted,

/s/ Jay P. Kesan

Jay P. Kesan (Reg. No. 37,488)
KEY KESAN DALLMANN PLLC
1050 Connecticut Avenue, N.W. Suite
500 Washington, D.C. 20036
jay.kesan@kkd-law.com
Telephone: (202) 772-1100

Alfonso G. Chan
Andrew Cochran
Cahill Gordon & Reindel LLP
900 16th Street, N.W., Suite 500
Washington, D.C. 20006
achan@cahill.com
acochran@cahill.com
Telephone: (202) 862-8942

Attorneys for Patent Owner
SKYSONG INNOVATIONS, LLC

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§ 42.6(e)(4) and 42.25(b), the undersigned certifies that on November 19, 2025, a complete copy of Patent Owner's Preliminary Response was filed electronically through the Patent Trial and Appeal Board's PTABE2E System and provided, via electronic service, to the Petitioner by serving the correspondence address of record.

Dated: November 19, 2025

/s/ Jay P. Kesan

Jay P. Kesan

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24, the undersigned certifies that the foregoing Patent Owner's Preliminary Response contains less than 14,000 words excluding the table of contents, table of authorities, certificate of service, certificate of word count, and appendix of exhibits. Patent Owner has relied on the word count feature of the word processing software used to create this paper in making this certification.

Dated: November 19, 2025

/s/ Jay P. Kesan
Jay P. Kesan