

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CROWDSTRIKE, INC.,
Petitioner

v.

SKYSONG INNOVATIONS, LLC,
Patent Owner

Case No. IPR2025-01397
Patent No. 10,313,385

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,313,385**

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
II.	THE '385 PATENT.....	1
	A. THE '385 PATENT'S ALLEGED INVENTION	1
	B. THE '385 PATENT'S PROSECUTION	3
	C. A PERSON OF ORDINARY SKILL IN THE ART.....	3
III.	REQUIREMENTS UNDER 37 C.F.R. § 42.104	4
	A. GROUNDS FOR STANDING UNDER 37 C.F.R. § 42.104(A)	4
	B. IDENTIFICATION OF CHALLENGE UNDER 37 C.F.R. § 42.104(B)	4
	C. CLAIM CONSTRUCTION UNDER 37 C.F.R. § 42.104(B)(3)	5
IV.	PRIOR ART SUMMARY	5
	A. VALLONE	5
	B. OBERHEIDE	7
	C. PATSOS	9
	D. ROYTMAN	10
	E. JUELS	12
	F. WOLFF	13
V.	GROUND 1: CLAIMS 1, 8, 14, AND 15 ARE OBVIOUS OVER VALLONE	15
	A. CLAIM 1	15
	B. CLAIM 8	28
	C. CLAIM 14	31
	D. CLAIM 15	32
VI.	GROUND 2: CLAIMS 1, 8, 14, AND 15 ARE OBVIOUS OVER OBERHEIDE, PATSOS, AND ROYTMAN.....	33
	A. CLAIM 1	33
	B. CLAIM 8	49
	C. 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."	52
	D. CLAIM 15	53
VII.	GROUND 3: CLAIMS 2-6, 9-12, 16, AND 17 ARE OBVIOUS OVER OBERHEIDE, PATSOS, ROYTMAN, AND JUELS	54

A.	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.”	54
B.	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.”	56
C.	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.”	57
D.	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.”	58
E.	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.”	60
F.	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.”	62
G.	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.”	62
H.	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.”	63
I.	CLAIM 12: “THE METHOD OF CLAIM 11, WHERE 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.”	64

J.	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.”	64
K.	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.”	64
VIII. GROUND 4: CLAIMS 7 AND 13 ARE OBVIOUS OVER <i>OBERHEIDE, PATSOS, ROYTMAN, AND WOLFF</i>.....		64
A.	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.”	64
B.	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.”	67
IX.	CONCLUSION.....	67
X.	MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(A)(1)	68
A.	REAL PARTY-IN-INTEREST	68
B.	RELATED MATTERS UNDER 37 C.F.R. § 42.8(B)(2).....	68
C.	LEAD AND BACK-UP COUNSEL	68
D.	37 C.F.R. § 42.8(B)(4) – SERVICE INFORMATION	69

TABLE OF AUTHORITIES

Cases:

<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005).....	5
<i>Skysong Innovations, LLC v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.</i> , Case No. 7:25-cv-00040-DC-DTG.....	69
<i>Skysong Innovations, LLC v. Fortinet, Inc.</i> , Case No. 2:25-cv-00098.....	69

Statutes:

35 U.S.C. § 102(a)(2)	5, 7, 10, 12
35 U.S.C. § 103	4
35 U.S.C. §§ 102(a)(1)	7, 9, 10, 13

Regulations:

37 C.F.R. § 42.100(b).....	5
37 C.F.R. § 42.104	4
37 C.F.R. § 42.104(a)	4
37 C.F.R. § 42.104(b).....	4
37 C.F.R. § 42.104(b)(3)	5
37 C.F.R. § 42.24	78
37 C.F.R. § 42.8	78
37 C.F.R. § 42.8(A)(1)	68
37 C.F.R. § 42.8(b)(1)	68
37 C.F.R. § 42.8(b)(2)	68
37 C.F.R. § 42.8(b)(3)	68
37 C.F.R. § 42.8(b)(4)	69
37 C.F.R. §§ 42.6(e)	79

I. INTRODUCTION

CrowdStrike, Inc. (“Petitioner”) respectfully requests *inter partes* review (“IPR”) of Claims 1-17 (“Challenged Claims”) of U.S. Patent No. 10,313,385 (“’385 Patent”).

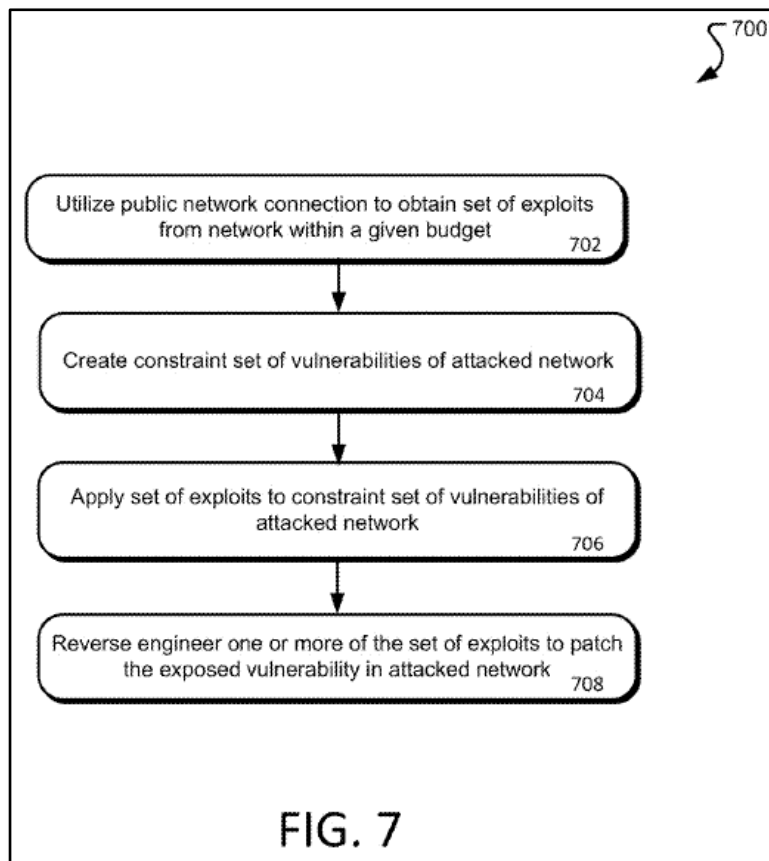
II. THE ’385 PATENT

A. The ’385 Patent’s Alleged Invention

Generally directed to a framework for “addressing penetration testing” “fed with real-world exploit market data,” the ’385 Patent purports to improve the “game theoretic approach to host-based defense where activities of the attacker are informed from an ‘un-conventional’ source...specifically information from darknet markets[.]” ’385 *Patent* (Ex. 1001), 2:20-30, 3:4-6; ’385 *File History* (Ex. 1002), 125. The ’385 Patent describes a game-theoretic model where an attacker purchases exploits to maximize her access to a target system, and a defender adjusts her targeted system to minimize the effect of the attack. ’385 *Patent*, 2:63-3:3. The model operates on a cost-optimization basis, requiring the attacker to purchase exploits within a budget. *Id.*, 2:63-66, 4:7-8.

The claims broadly recite a typical penetration test using real-world exploit information mined from the darknet. *Id.*, 2:30-31, 3:4-7. The claimed penetration test comprises: (1) accessing dark net information associated with exploits, (2) obtaining a set of exploits from that dark net information, (3) applying an exploit function to the set of exploits to identify a set of vulnerabilities, (4) creating a

constraint set of vulnerabilities from the returned set of vulnerabilities, (5) applying the set of exploits to the constraint set, (6) analyzing the previous step to identify a particular vulnerability, and (7) patching the computer system based on the analysis to reduce the effect of a future attack. *Id.*, claim 1; *see also id.*, 3:16-4:5, 9:5-33, 13:3-18. This test is depicted below in Figure 7:



Id., Fig. 7.

B. The '385 Patent's Prosecution

The '385 Patent claims priority to a provisional application filed on November 30, 2015. For purposes of this Petition only, Petitioner applies November 30, 2015, as the priority date for the Challenged Claims.

The originally presented claims faced a single rejection. '385 *Patent File History* (Ex. 1002), 87-93, 118-129. In response, Applicant amended the claims “to clarify that data is accessed through the dark net[,]” “to clarify that a set of exploits is obtained from the dark net information,” that “an exploit function is implemented to return a set of vulnerabilities[,]” and to “further clarify that the claimed subject matter is implemented to reduce potential damage of a cyberattack.” *Id.*, 126-128. A Notice of Allowance issued thereafter. *Id.*, 131-141.

As explained in the Grounds below, *Vallone*, *Oberheide*, *Patsos*, *Roytman*, *Juels*, and *Wolff* expressly disclose the allegedly patentable system and methods for using exploit data mined from the dark net to exploit vulnerabilities in a computer system and then remediate those vulnerabilities. *Infra*.

C. A Person of Ordinary Skill in the Art

A POSITA at the time of the '385 Patent would have had a bachelor's degree in computer science, electrical or computer engineering, or an equivalent degree; and 2-3 years of work or research experience in computer security and networking.

Dec., ¶¶53-55.¹ Additional education (e.g., MS degree) might substitute for less experience and vice versa. *Id.*

III. REQUIREMENTS UNDER 37 C.F.R. § 42.104

A. Grounds for Standing Under 37 C.F.R. § 42.104(a)

Petitioner certifies the '385 Patent is eligible for IPR and Petitioner is not barred or estopped from requesting IPR challenging the claims of the '385 Patent. Petitioner is not the owner of the '385 Patent, has not filed a civil action challenging the validity of any claim of the '385 Patent, and this Petition is filed less than one year after Petitioner was served with a complaint alleging infringement of the '385 Patent.

B. Identification of Challenge Under 37 C.F.R. § 42.104(b)

Petitioner requests the Challenged Claims be found unpatentable on the following grounds:

Proposed Grounds of Unpatentability	Exhibits
Ground 1: Claims 1, 8, 14, and 15 are obvious under AIA 35 U.S.C. § 103 over <i>Vallone</i>	Ex. 1004
Ground 2: Claims 1, 8, 14, and 15 are obvious under AIA 35 U.S.C. § 103 over <i>Oberheide</i> in view of <i>Patsos</i> and <i>Roytman</i>	Ex. 1005; Ex. 1006; Ex. 1007
Ground 3: Claims 2-6, 9-12, 16, and 17 are obvious under AIA 35 U.S.C. § 103 over <i>Oberheide</i> in view of <i>Patsos</i> , <i>Roytman</i> , and <i>Juels</i>	Ex. 1005; Ex. 1006; Ex. 1007; Ex. 1008

¹ All citations to “Dec.” are to paragraph numbers in Ex. 1003, Declaration of Dr. Erez Zadok.

Ground 4: Claims 7 and 13 are obvious under AIA 35 U.S.C. § 103 over <i>Oberheide</i> in view of <i>Patsos</i> , <i>Roytman</i> , and <i>Wolff</i>	Ex. 1005; Ex. 1006; Ex. 1007; Ex. 1009
--	--

C. Claim Construction Under 37 C.F.R. § 42.104(b)(3)

In this proceeding, claims are interpreted under the same standard applied by Article III courts (the *Phillips* standard). 37 C.F.R. § 42.100(b); *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (*en banc*). Petitioner applies the ordinary and customary meaning of the claim terms as understood by a POSITA.²

IV. PRIOR ART SUMMARY

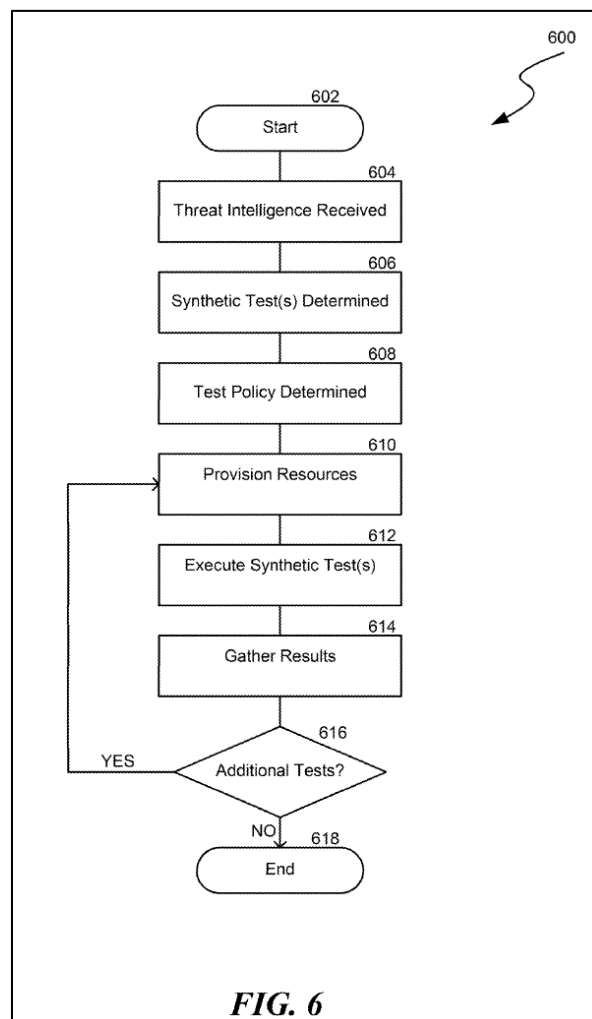
A. Vallone

Vallone was filed on November 11, 2014, and published on May 12, 2016, qualifying as prior art to the '385 Patent under 35 U.S.C. § 102(a)(2). *Vallone* (Ex. 1004).

Vallone teaches a “device...for assessing a target network’s vulnerability to a real cyberthreat [using] synthetic tests configured to model the behavior of the cyberthreat.” *Vallone*, Abstract. *Vallone*’s device “access[es] cyberthreat

² Petitioner’s challenge is not a waiver of any argument in litigation that any claim term in the '385 Patent is indefinite or invalid. Petitioner does not waive its rights to seek to construe claim terms in litigation, including in the pending district-court litigation.

intelligence networks...[such as] the dark net, etc.” to determine cyberthreat objectives, obtains “threat characteristics” including the cyberthreat’s sequence of actions, executes instructions at a host computer “with the intent of generating the threat’s characteristics into the target environment,” “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.” *Vallone*, Abstract, [0016], [0019], [0039], [0040]. This method is illustrated in Figure 6:



Id., Fig. 6. *Vallone*'s synthetic penetration test is performed by "agents" installed on "host" computers for testing the vulnerabilities on the target network. *Id.*, [0015]-[0017]. In this way, *Vallone*'s system is "able to synthesize the cyberthreat...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." *Id.*, [0038].

Vallone, like the '385 Patent, discloses systems and methods for exploiting vulnerabilities and patching vulnerabilities. *Compare* '385 Patent, 3:4-15, 3:61-64, 13:12-18, Figs. 5 and 7 *with Vallone*, Abstract, [0021]-[0022], [0038]-[0040], Fig. 6. Thus, *Vallone* is in the same field of endeavor as the '385 Patent. *Dec.*, ¶88.

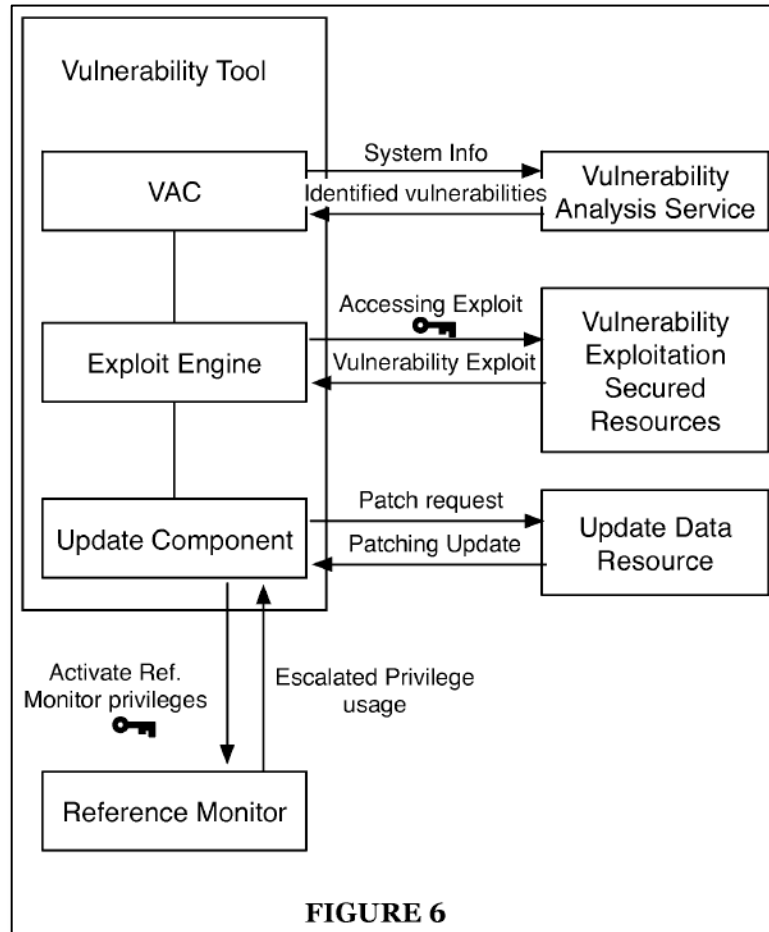
Vallone is also reasonably pertinent to certain problems addressed by the '385 Patent, including developing patches for system vulnerabilities. *Compare* '385 Patent, 2:63-3:3, 13:15-18 *with Vallone*, Abstract, [0020]-[0022]. *Vallone* is therefore analogous art to the '385 Patent. *Dec.*, ¶¶89-90.

B. Oberheide

Oberheide was filed on February 24, 2014, and published on August 28, 2014, qualifying as prior art to the '385 Patent under 35 U.S.C. §§ 102(a)(1) and 102(a)(2). *Oberheide* (Ex. 1005).

Oberheide teaches a "system and method that includes identifying a vulnerability in a computing device; accessing a vulnerability exploitation mapped

to the identified vulnerability; at the computing device, executing the vulnerability exploitation and...updating the system with a vulnerability resolution.” *Oberheide*, Abstract, [0013], [0019]. This process is depicted below in Figure 6:



Id., Fig. 6. *Oberheide* teaches a “vulnerability tool” downloadable to a device for performing the described functions. *Id.*, [0014], Fig. 6. *Oberheide*’s vulnerability tool includes a “vulnerability assessment component (VAC)” that identifies vulnerabilities using “any suitable approach,” including “malware detection approaches.” *Id.*, [0015], [0020]. The vulnerability tool also includes an exploit engine that accesses “exploitation scripts” to exploit vulnerabilities. *Id.*, [0017]. An

update component then applies “a patch or system update to resolve or reduce an existing vulnerability.” *Id.*, [0018], [0024], [0026].

Oberheide, like the ’385 Patent, discloses systems and methods for exploiting vulnerabilities and patching vulnerabilities. *Compare* ’385 Patent, 3:4-15, 3:61-64, 13:12-18, Figs. 5 and 7 *with Oberheide*, Abstract, [0013], [0017], [0019], [0026], Figs. 5-8. Thus, *Oberheide* is in the same field of endeavor as the ’385 Patent. *Dec.*, ¶164.

Oberheide is also reasonably pertinent to certain problems addressed by the ’385 Patent, including developing patches for system vulnerabilities. *Compare* ’385 Patent, 2:63-3:3, 13:15-18 *with Oberheide*, Abstract, [0003], [0013], [0019]. *Oberheide* is therefore analogous art to the ’385 Patent. *Dec.*, ¶165.

C. Patsos

Patsos published in October 2010, qualifying as prior art to the ’385 Patent under 35 U.S.C. § 102(a)(1). *Patsos* (Ex. 1006); *Munford Dec.* (Ex. 1031), ¶¶6-9.

Patsos teaches an “incident response intelligence system (IRIS)” that is a security-management platform that “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.*, 291, 293. To accomplish this, IRIS “collects security information related to vulnerabilities,” including real world exploit

information from various sources including “underground web sources” and “hacking web sites[.]” *Id.*, 295-297 (“[i]nformation related to exploits (and actual exploit code) can be obtained from hacking web sites” using an “exploit gathering module”). IRIS then uses a “vulnerability exploit correlation module (VECM)” to correlate exploits to vulnerabilities and countermeasures. *Id.*, 298 (describing “providing a full correlation between all attack components, i.e.[,] the vulnerabilities, the exploits that can potentially realize these vulnerabilities and the associated countermeasures”).

Patsos, like the ’385 Patent, discloses systems and methods for exploiting vulnerabilities and patching vulnerabilities. *Compare* ’385 Patent, 3:4-15, 3:61-64, 13:12-18, Figs. 5 and 7 *with Patsos*, 292, 295, 300. Thus, *Patsos* is in the same field of endeavor as the ’385 Patent. *Dec.*, ¶169.

Patsos is also reasonably pertinent to certain problems addressed by the ’385 Patent, including improving the security of a computer system or device. *Compare* ’385 Patent, 2:63-3:3, 13:15-18 *with Patsos*, 292, 293, 296, 303. *Patsos* is therefore analogous art to the ’385 Patent. *Dec.*, ¶170.

D. Roytman

Roytman was filed on March 9, 2015, and published on August 20, 2015, qualifying as prior art to the ’385 Patent under 35 U.S.C. §§ 102(a)(1) and 102(a)(2). *Roytman* (Ex. 1007).

Roytman teaches techniques for “ranking a set of vulnerabilities of a computing asset and set of remediations for a computing asset[.]” *Roytman*, Abstract. “In one technique, vulnerabilities...are received at a vulnerability intelligence platform” along with “[b]reach data indicating set of breaches that occurred[.]” *Id.* Then, a “subset of the set of vulnerabilities that are most vulnerable to a breach is identified[.]” a “risk score is generated[.]” and “remediations are ordered based on the remediations that would reduce the risk score the most[.]” *Id.*; *see also id.*, [0025], Fig. 3.

Specifically, in *Roytman*, vulnerabilities are identified from one or more computing assets in a customer network. *Id.*, [0025], Fig. 3. The computing asset may be one that is “absolutely critical” for day-to-day operations. *Id.*, [0044]. A server then receives data on breaches (e.g., successful exploits) to determine “[a] subset of the set of vulnerabilities that are most vulnerable to the breach[.]” *Id.*, [0025], [0032]. A risk score is assigned to the vulnerabilities. *Id.*, [0026]-[0027], [0034]. Using the vulnerabilities and their corresponding risk scores, remediations are selected to “reduc[e] the risk posed by one or more vulnerabilities of a computing asset or a group of computing assets.” *Id.*, [0092]-[0093], [0100].

Roytman, like the ’385 Patent, discloses systems and methods for exploiting vulnerabilities and patching vulnerabilities. *Compare* ’385 Patent, 3:4-15, 3:61-64, 13:12-18, Figs. 5 and 7 *with Roytman*, [0027], [0038], [0060]-[0061], [0069],

[0092]. Thus, *Roytman* is in the same field of endeavor as the '385 Patent. *Dec.*, ¶174.

Roytman is also reasonably pertinent to certain problems addressed by the '385 Patent, such as improving the security of a computer system or device. *Compare* '385 Patent, 2:63-3:3, 13:15-18 *with Roytman*, [0002], [0027], [0038], [0100]. *Roytman* is therefore analogous art to the '385 Patent. *Dec.*, ¶175.

E. Juels

Juels was filed on February 24, 2012, qualifying as prior art to the '385 Patent under 35 U.S.C. § 102(a)(2). *Juels* (Ex. 1008).

Juels teaches using a game-theoretic model to protect a processing system from security threats. *Juels*, 1:7-10, 2:7-10. The model may include both “attacker and defender entities [that] can take actions at any time, with the actions having associated costs[.]” *Id.*, 2:34-38. The processing system identifies “a plurality of defensive security actions to be taken to address a persistent security threat to a system comprising information technology infrastructure[.]” *Id.*, 2:19-22. Optimal defensive security actions are decided based on the actions taken in the game-theoretic model, such as “key update operations, virtual machine refresh operations, cloud service auditing operations, password reset operations, and many others.” *Id.* 2:41-43, 15:24-27 (describing password reset based on attacker purchasing password in “underground forum”). *Juels*, like the '385 Patent, discloses systems

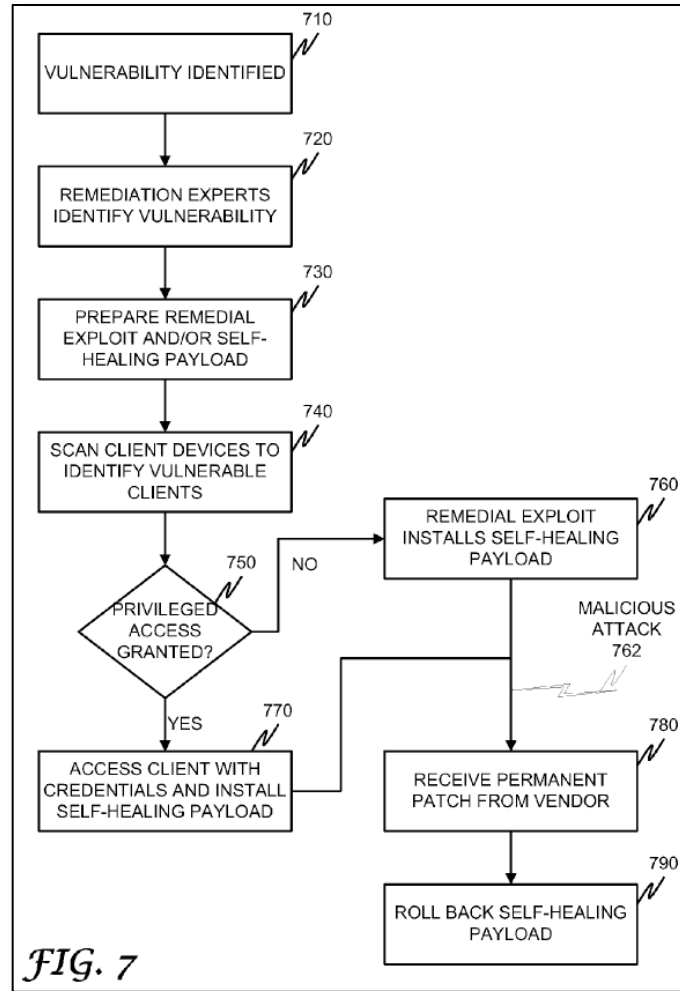
and methods for exploiting vulnerabilities and patching vulnerabilities. *Compare* '385 Patent, 3:4-15, 3:61-64, 13:12-18, Figs. 5 and 7 *with Juels*, 2:39-43, 12:60-63, 13:6-11, 13:25-34. Thus, *Juels* is in the same field of endeavor as the '385 Patent. *Dec.*, ¶273.

Juels is also reasonably pertinent to certain problems addressed by the '385 Patent, such as improving the security of a computer system or device. *Compare* '385 Patent, 2:63-3:3, 13:15-18 *with Juels*, Abstract, 1:7-10, 2:7-10, 2:39-47. *Juels* is therefore analogous art to the '385 Patent. *Dec.*, ¶274.

F. Wolff

Wolff was filed on October 17, 2011, and published on August 1, 2013, qualifying as prior art to the '385 Patent under 35 U.S.C. §§ 102(a)(1) and 102(a)(2). *Wolff* (Ex. 1009).

Wolff teaches an approach to remediating “security vulnerabilities in computer software[.]” *Id.*, [0015]. *Wolff* describes a “computer security vulnerability remediation system (CSVRS)” that includes a CSVRS client coupled to a remediation server over a network, the CSVRS being a “computer or computing device.” *Id.*, [0014], [0029]. “The CSVRS client includes software having a security vulnerability” that may be exploited by a malicious actor. *Id.* Figure 7 below depicts an exemplary method for this vulnerability remediation:



Wolff, Fig. 7.

In *Wolff*'s method, a Remediation Service Provider (RSP), which operates the remediation server, receives information on an exploit that may attack the identified vulnerability in the CSVRS client and “may modify the exploit to create a remedial exploit, including a self-healing payload.” *Id.*, [0014], [0027]; *see also id.*, [0019] (describing buying exploits on sale on “black market ‘underground economy servers’”). The self-healing payload is deployed on the CSVRS client to take actions such as “downloading programs, executing programs, downloading software

updates, installing patches, *removing software*, modifying privileges,” and many others. *Id.*, [0028]. If the actions are adequate, the self-healing payload will ameliorate the effects of an incoming malicious attack on the CSVRS client. *Id.*, [0028], [0047]. *Wolff*, like the ’385 Patent, discloses systems and methods for exploiting vulnerabilities and patching vulnerabilities. *Compare* ’385 Patent, 3:4-15, 3:61-64, 13:12-18, Figs. 5 and 7 *with Wolff*, Abstract, [0014]-[0016], [0028], [0033]. Thus, *Wolff* is in the same field of endeavor as the ’385 Patent. Dec., ¶328.

Wolff is also reasonably pertinent to certain problems addressed by the ’385 Patent, such as improving the security of a computer system or device. *Compare* ’385 Patent, 2:63-3:3, 13:15-18 *with Wolff*, Abstract, [0014], [0021] [0047]. *Wolff* is therefore analogous art to the ’385 Patent. Dec., ¶329.

V. GROUND 1: CLAIMS 1, 8, 14, AND 15 ARE OBVIOUS OVER VALLONE

A. Claim 1

1. Claim 1(Pre): “A system for protecting a computer system from attack, the system comprising:”

To the extent the preamble is limiting, *Vallone* teaches systems for “modeling risk of cyber-attacks[.]” *Vallone*, [0001]. *Vallone*’s system “assess[es] a target network’s vulnerability to a real cyberthreat” and uses feedback “to find patches for known vulnerabilities[.]” *Id.*, Abstract. This process is used to “dynamically update the target network’s security capabilities to, e.g., protect

against the same and/or an other vulnerability.” *Id.*, [0020]-[0021]. Patching vulnerabilities protects a computer system from attack. *Dec.*, ¶¶93-94.

2. Claim 1(a): “a network connection;”

Vallone’s system is connected to a network. *Vallone*, [0037]. Specifically, “[t]hreat simulation technology 500 is configured to communicate to **external network 250** to...receive information regarding cyberthreats” by “**access[ing] cyberthreat intelligence networks**, local and/or remote data repositories (e.g., data 520), the dark net, etc. to determine whether a new cyberthreat has been released[.]” *Id.* [0037], [0040]³; *see also*, Fig. 2B.

3. Claim 1(b): “a processing device; and”

Vallone’s system includes a processing device. *Vallone*, [0028]. *Vallone* teaches its system includes “routines executed by a general- or special-purpose **data processing device** (e.g., a server or client computer).” *Vallone*, [0028]. *Vallone’s* “threat simulation technology...includes data 520 and server(s) 510[.]” *Id.*, [0040], Figs. 2B, 5. The server (i.e., processing device) of *Vallone’s* threat simulation technology performs the claimed steps, as described below. *Dec.*, ¶¶97-98.

4. Claim 1(c): “a non-transitory computer-readable medium connected to the processing device configured to store

³ All emphasis added unless otherwise noted.

instructions that, when executed by the processing device, performs the operations of:

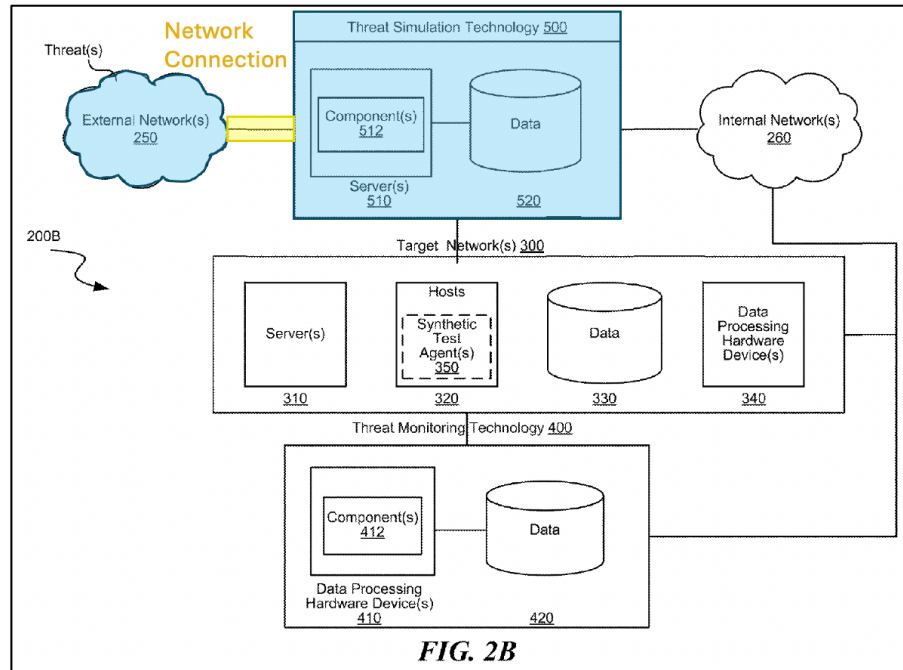
Vallone teaches “computer executable instructions” “stored on a machine-readable storage medium [that are] executed by one or more general-purpose or special-purpose programmable microprocessors.” *Id.*, [0028], [0045]. *Vallone* provides example “non-transitory” machine-readable storage media, including ROM, RAM, magnetic disc storage, optical storage, and flash memory devices. *Id.*; *Dec.*, ¶¶99-100. With respect to *Vallone*’s specific embodiments, *Vallone*’s threat simulation technology includes “data 520” connected to “server 510[.]” *Vallone*, Figs. 2B, 5. *Vallone*’s ROM with data 520 is “a non-transitory computer-readable medium connected to the processing device [i.e., server 510] configured to store instructions” that are executed by server 510. *Dec.*, ¶¶99-101.

5. ***Claim 1(d): “accessing data through the network connection, the data comprising dark net information associated with exploits of a computer system, comprising:”***

Vallone teaches its threat simulation technology “fetch[es]...information regarding the existence and details regarding cyberthreats.” *Vallone*, [0040]. “Cyberthreats” include “viruses, Trojans, rootkits, **zero-day exploits**[.]” *Id.*, [0005]. *Vallone* teaches that “threat intelligence component 513 can access” “the **dark net** [] to determine whether a new cyberthreat has been released and, if so, what its capabilities and objectives are[.]” *Id.*, [0040]. Threat simulation technology “receive[s] information regarding [dark net] cyberthreats” over

network connection 250. *Id.*; *see also id.*, [0037]. This process is depicted below in

Figure 2B:



Id., Fig. 2B (annotated).

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

In litigation, Skysong accuses CrowdStrike’s Falcon Platform and specifically the Falcon Adversary Intelligence module of infringing this limitation. Ex. 1030, 45-46 (further accusing older modules including “Falcon Platform Falcon Intelligence” and “False Intelligence Recon+”). To “obtain[] a set of exploits from the dark net information,” Skysong alleges these modules “implement ‘dark web monitoring’ to ‘develop a more comprehensive understanding of attackers and their methods’.” *Id.*, 45 (citing CrowdStrike’s

website that states “[d]ark web monitoring continuously searches the dark web and pulls in raw intelligence in near real time... The data captured by the dark web monitoring solution can be fed into automated threat intelligence systems and used to enrich that data”).

For application of alleged exploit code, Skysong accuses the CrowdStrike Falcon Sandbox that “‘establishes a space where potentially harmful code can be safely executed’...in which ‘investigators can ensure the rest of the system remains unaffected, thereby minimizing the risk of a security breach’.” *Id.*, 53-60. That is, Skysong alleges CrowdStrike’s Falcon Platform obtains exploit code/script from the dark web and applies it in a sandboxed environment. *Dec.*, ¶107. To the extent the claims are interpreted such that “obtaining a set of exploits” includes crawling the net for exploit code, *Vallone* teaches this limitation.⁴

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.” *Vallone*, [0016]. These “[c]haracteristics are mapped to instructions via logic...that when executed by an agent would achieve...an objective of the cyberthreat.” *Id.*, [0017] (“synthetic characteristics” are “intended to exhibit, via an

⁴ CrowdStrike does not waive its right to construe this limitation or contest Skysong’s interpretation in litigation, including in pending district court litigation.

agent, the real threat's characteristics"). For example, *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." *Id.*, [0019], [0038]. By obtaining threat characteristics, such as scripts or programs regarding multiple cyberthreats from the dark net, *Vallone*'s system "obtain[s] a set of exploits from the dark net information[.]" *Dec.*, ¶¶108-110.

These exploits are "configured to penetrate the computer system[.]" *Dec.*, ¶¶111-112. Specifically, *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." *Vallone*, [0040]. These objectives may penetrate a computer system to, e.g., "steal usernames and passwords[.]" *Id.* (teaching determining multiple steps for an exploit, including creating files, opening communication tunnels, etc., for penetrating a host computer).

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

Vallone teaches applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities by determining the "sequence of actions" an exploit takes to penetrate the computer system. *Vallone*, [0016], [0040]; *Dec.*, ¶¶115-121. *Vallone*'s synthetic test generator inputs exploits and determines

“one or more steps of instructions” for emulating the exploits in synthetic tests. *Vallone*, [0041] (“synthetic tests are determined by **mapping the exploit’s characteristics (e.g., any files created/accessed, communications spawned) into one or more steps of instructions**”).

For example, *Vallone* teaches obtaining sets of cyberthreats (i.e., exploits). *Id.*, [0037] (“receive information regarding cyberthreats”), [0040]. “With this information, the synthetic test generator component 514...determines...one or more steps of an instruction for emulating that portion of the objective.” *Id.*, [0040], Fig. 5. A POSITA would have understood that the steps for emulating each exploit would return one or more vulnerabilities associated with the application of each exploit. *Dec.*, ¶¶117-118. *Vallone* describes an example exploit attack path, including instructions that require both execution of a malicious Flash script and the establishment of a data exfiltration tunnel. *Vallone*, [0040]. This example involves exploitation of a client-side vulnerability (e.g., a Flash Player exploit) and a separate network-level vulnerability enabling unauthorized outbound communication. *Id.* (describing multiple steps of an exploit, including “infiltrating a network via running a Flash browser-based script ‘badscript.fl’ and creating a tunnel between data 520 and an Internet server”). *Vallone*’s system thus inputs an exploit and returns one or more vulnerabilities. *Dec.*, ¶119.

Vallone’s teachings are consistent with the ’385 Patent’s description of its

“exploit function,” which maps the attacker’s strategy. *See, e.g.*, ’385 Patent, 3:61-63 (“The set of vulnerabilities produced by [exploit function] ExF(A), for a given set of exploits A, represents the vulnerabilities that are exploited by the exploits in A”), 4:9-11 (“identifies a set of vulnerabilities that he is willing to expose (often referred to as the ‘presented attack surface’”), 4:53-54 (“vulnerabilities generated by an attack strategy’s exploits”), 10:6-9 (“An exploit function (denoted Ex) maps the attacker’s strategy to a set of vulnerable software applications, which can be easily obtained from darknet marketplace data”); *see also id.*, 3:59-4:5 (“many possible variations of an exploit function are possible”). Similarly, *Vallone* maps an attack path of an exploit with instructions for emulating the attack in a synthetic test, where this attack path identifies the vulnerable components and/or software exploited by the vulnerability. Dec., ¶120.

8. *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 litigation, Skysong accuses CrowdStrike’s Falcon Platform and specifically the Falcon Sandbox module of infringing this limitation. Ex. 1030, 53-58. To “creat[e] a constraint set of the computer system from the set of vulnerabilities,” Skysong alleges Falcon Sandbox “establishes a space where potentially harmful code can be safely executed.” *Id.*, 53. Skysong contends that by “creat[ing] a controlled and isolated environment” such that “sandboxes restrict

an application's access to certain system resources," Falcon Sandbox's alleged "constraint set compris[es] a minimum set of dependencies to operate the computer system." *Id.* (accusing sandboxes that "provide enough resources for testing malicious code while still preventing that code from causing wider damage"). That is, Skysong accuses constraining harmful code in a controlled environment as the claimed "creating a constraint set." To the extent the claims are interpreted such that "creating a constraint set" where "the constraint set comprises a minimum set of dependencies to operate the computer system" includes testing malicious code against computer vulnerabilities in an environment with enough resources to test the malicious code while preventing wider damage, *Vallone* teaches this limitation.⁵

Vallone's system "creat[es] a constraint set of the computer system from the set of vulnerabilities" by "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[.]" *Vallone*, [0038]; *see also, id.*, [0014] (teaching "methods, devices, and systems ... related to generating synthetic security events, representative of actual cyberthreat activity, for testing the effectiveness of a

⁵ CrowdStrike does not waive its right to construe this limitation or contest Skysong's interpretation in litigation, including in pending district court litigation.

network's detective and preventative information security controls"). To accomplish this, *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)." *Vallone*, [0038]. In other words, by mimicking the sequence of events of an attack on the host computer, *Vallone* is "creating a constraint set of the computer system[.]" *Dec.*, ¶¶124-126.

Vallone's constraint set "compris[es] a minimum set of dependencies to operate the computer system" because the "executable instructions [] 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[.]" *Vallone*, [0038]; *see also*, *id.*, [0040] (teaching emulating a cyberthreat "without the hazard of the objective actually being achieved[,]" such as "steal[ing] usernames and passwords"); *Dec.*, ¶127. Accordingly, *Vallone's* system is restricting access at the host for, e.g., posting credit card information on the Internet in the same way that Skysong accuses testing malicious code while preventing wider damage. Infringement Contentions (EX1030), 53-54; *Dec.*, ¶127.

Vallone's system forms a controlled and isolated environment with the agents on the host computer (i.e., "creating a constraint set of the computer system from the set of vulnerabilities") that allows for testing potentially harmful code up

until completion such that the ultimate objective is not committed (i.e., “the constraint set comprising a minimum set of dependencies to operate the computer system”). *Dec.*, ¶¶122-128.

9. Claim 1(h): “applying the set of exploits to the constraint set of the computer system;”

Once the instructions are compiled for simulating the exploit, the synthetic test is executed (i.e., *applying the set of exploits to the constraint set of the computer system*). See, e.g., *Vallone*, [0041] (“a test is executed, at step 612”), Fig. 6. *Vallone*’s “synthetic test” “synthesize[s] 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, while still providing an administrator with knowledge (e.g., via feedback from agent(s) 350 and/or network equipment 340) regarding the depth of vulnerability of the target network 300 to an actual attack.” *Id.*, [0038]. For example, “synthetic test generator component 514, in various embodiments, determines, for at least a portion of a cyberthreat’s objective, one or more steps of an instruction for emulating that portion of the objective.” *Id.*, [0040]. “[I]f a cyberthreat’s objective is to steal usernames and passwords and includes steps for infiltrating a network via running a Flash browser-based script ‘badscript.fl’ and creating a tunnel between data 520 and an Internet server, then synthetic test generator 514 can determine corresponding instructions for execution by an agent

350 at a host 320 of a target network 300 for simulating that objective, without the hazard of the objective actually being achieved.” *Id.*

In this way, *Vallone*’s threat simulation technology “appl[ies] the set of exploits to the constraint set of the computer system” by installing and instructing an agent to execute the instructions at the host computer. *Dec.*, ¶¶129-134. Specifically, threat simulation technology 500 “installs and/or instantiates one or more agents 350 on a host 320.” *Vallone*, [0038]. These agents “are configurable by the technology to execute instructions on a host 320 for performing characteristics of a cyberthreat[.]” *Id.* (agents “contain[] features that when configured with instructions can synthesize the characteristics of the cyberthreat”). This includes running “several simultaneous synthetic tests at a [] time” for multiple exploits. *Vallone*, [0041]; *Dec.*, ¶¶131-133.

10. 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’s system includes “sensor component 416” which “monitors target network 300 activity during a synthetic test and passes feedback data back to...threat simulation technology 500. *Vallone*, [0039]. The “[f]eedback data (e.g., real-time and/or post-instruction data) from the agent(s) 350 and/or network equipment 340...is **sent to and/or accessed by the threat simulation technology [500], which can further analyze the data[.]**” *Id.*, [0038]. To this end, *Vallone*’s

threat simulation technology includes “validation feedback component 516” which “receives the feedback data...from the target network 300. The feedback data can be stored as data 520 and used by the other components 512 (e.g., the synthetic test policy component 515) to improve the efficiency of **detecting**, assessing, responding to, and fixing **vulnerabilities**.” *Vallone*, [0040]. *Vallone* specifies that “if an instruction has multiple steps...each step can send feedback regarding its progress (e.g., whether the step was successful, failed, was interrupted, etc.).” *Id.*, [0019]. As such, *Vallone*’s threat simulation technology analyzes the application of the exploits by analyzing the feedback data and detects a particular vulnerability if execution of a certain step was successful. *Dec.*, ¶¶136-137.

11. *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.”*

Vallone teaches that “[b]ased on the analysis of the validation feedback data...the technology can dynamically update the target network’s security capabilities to, e.g., protect against the same and/or an other vulnerability.” *Vallone*, [0020]. *Vallone* calls these dynamic updates “self-healing process[.]” and provides examples such as “downloading and/or installing updates, firmware, new releases, antivirus rules, etc.” *Id.* Based on feedback data, “the target networks 300 can be (re)provisioned (e.g., **by updating** servers 310 and **hosts 320 with recent security updates**)[.]” *Id.*, [0040] (“[t]he feedback data can be stored as data 520

and used by the other components 512 (e.g., the synthetic test policy component 515) to improve the efficiency of detecting, assessing, responding to, and fixing vulnerabilities”). Thus, *Vallone*’s system updates “hosts” (i.e., “*altering a configuration of the computer system*”) based on the feedback data from the synthetic test (i.e., “*in response to the analysis of the application of the set of exploits*”) which protects the computer system from future cyberattacks (i.e., “reduce[s] potential damage of a cyberattack”). *Dec.*, ¶¶139-140.

B. Claim 8

1. Claim 8(Pre): “A method for improving a computer device, the method comprising:”

To the extent the preamble is limiting, *Vallone* teaches “[a] method for improving a computer device[.]” *Vallone*, [0020]-[0021] (teaching its process “dynamically update[s] the target network’s security capabilities to, e.g., protect against the same and/or an other vulnerability), Claims 1-23 (teaching “[a] method,” and more specifically “a computer-implemented method”); *see also*, Claim 1(Pre). Protecting a computer device from vulnerabilities improves the computer device. *Dec.*, ¶¶92-93, 141.

2. Claim 8(a): “accessing data comprising dark net information associated with a computer system;”

Vallone teaches “accessing data comprising dark net information[.]” *Vallone*, [0005] (teaching accessing “**the dark net** [] to determine whether a new cyberthreat has been released and, if so, what its capabilities and objectives are”); *supra*, Claim

1(d). The cyberthreats (i.e., dark net information) are “associated with a computer system[.]” *Vallone*, [0015] (describing “[h]osts” as “laptops” that “can be affected by a cyberthreat”).

3. ***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;”***

Vallone teaches “obtaining a set of exploits from the dark net information[.]” See Claim 1(e). These “set of exploits [are] configured to bypass a security feature of the computer system[.]” *Vallone*, [0040] (describing exemplary scenarios in which the cyberthreat bypasses a security system, such as “steal[ing] usernames and passwords”); *supra*, Claim 1(e).

4. ***Claim 8(c): “applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities;”***

Vallone applies an exploit function which takes the set of exploits as input and returns a set of vulnerabilities (i.e., vulnerable components and/or software in an exploit attack path).). *Supra*, Claim 1(f).

5. ***Claim 8(d)(1): “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,”***

Vallone teaches “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” by testing malicious code against computer vulnerabilities in an environment with enough resources to test the malicious code

while preventing wider damage. *Supra*, Claim 1(g) (comparing *Vallone* to Skysong’s infringement allegations in litigation); *compare* Ex. 1030 (infringement contentions) 53-58 *with* 120-127 (alleging Falcon Sandbox infringes Claims 1(g) and 8(d)(1) in the same way).

6. ***Claim 8(d)(2): “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;”***

Vallone teaches applying the set of exploits. *Supra* Claims 1(h). *Vallone* further teaches “determining the effect of the set of exploits on the constraint set of vulnerabilities of the computer system[.]” *See, e.g., Vallone*, [0038]-[0039] (teaching a sensor for monitoring the network activity during the synthetic test and passing feedback data to threat simulation technology for analysis); *Dec.*, ¶148.

7. ***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; and”***

Vallone teaches “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[.]” *Supra*, Claim 1(i). “[I]f an instruction has multiple steps...each step can send feedback regarding its progress (e.g., whether the step was successful, failed, was interrupted, etc.)” such that *Vallone* analyzes the feedback data and detects a particular vulnerability if execution of a

certain step was successful. *Vallone*, [0019]; *Dec.*, ¶¶136-138, 149; *supra*, Claim 1(i).

8. ***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.”***

Vallone alters 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 by “updating...hosts 320 with recent security updates” based on feedback data. *Vallone*, [0040]; *supra*, Claim 1(j).

C. Claim 14

1. ***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.”***

Vallone teaches altering the configuration of the computer system. *Vallone*, [0040] (teaching, based on feedback, “updating...hosts 320 with recent security updates”); Claims 8(f), 1(j). This includes “downloading and/or installing updates, firmware, new releases, antivirus rules, etc....[which] alleviates the time-consuming measures...when there are hundreds of computing devices[.]” *Vallone*, [0020]; *see also, id.*, [0022] (teaching healing the network includes “addressing the vulnerability”). As such, *Vallone* teaches “installing a patch [i.e., updates, firmware, new releases] to a software component of the computer system that is exposed by the set of exploits to attack the computer system.” *Dec.*, ¶¶151-152.

D. Claim 15

1. ***Claim 15(Pre): “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:”***

To the extent the preamble is limiting, *Vallone* teaches “[o]ne or more non-transitory tangible computer-readable storage media storing computer-executable instructions for performing a computer process on a machine[.]” *Vallone*, [0028] (teaching “computer-executable instructions, such as routines executed by a...processing device” “stored...on tangible computer-readable media”), Claims 10-23 (teaching “[a] computer-readable storage device having instructions that, when executed by at least one processor, causes the at least one processor to perform a computer-implemented method”); *supra*, Claim 1(c).

2. ***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;”***

Vallone teaches obtaining exploits from dark net information which are configured to bypass a security feature. *Supra*, Claims 1(e), 8(b).

3. ***Claim 15(b): “applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities;”***

Vallone teaches applying an exploit function as claimed. *Supra*, Claims 1(f), 8(c).

4. ***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;”***

Vallone teaches 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. *Supra*, Claims 8(d)(1), 1(g).

5. ***Claim 15(d): “applying the set of exploits to the constraint set of vulnerabilities of the computer system;”***

Vallone teaches applying the set of exploits to the constraint set of vulnerabilities of the computer system. *Supra*, Claim 1(h).

6. ***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”***

Vallone teaches analyzing the application of the set of exploits as claimed. *Supra* Claims 1(i), 8(e).

7. ***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.”***

Vallone teaches altering a configuration of the computer system as claimed. *Supra* Claims 1(j), 8(f).

VI. GROUND 2: CLAIMS 1, 8, 14, AND 15 ARE OBVIOUS OVER OBERHEIDE, PATSOS, AND ROYTMAN

A. Claim 1

1. ***Claim 1(Pre): “A system for protecting a computer system from attack, the system comprising:”***

To the extent the preamble is limiting, *Oberheide* discloses a system for

protecting a computer system from attack. *Oberheide*, [0013] (disclosing “a system for patching a device”). *Oberheide*’s system is used to “reduce the vulnerabilities of a system” and “apply a persistent fix on a computer system[.]” *Id.* A vulnerability is “a flaw or weakness” of a system that risks exploitation in an attack. *Oberheide*, [0003]. By reducing weaknesses and applying persistent fixes to a computer system, *Oberheide* teaches “[a] system for protecting a computer system from attack[.]” *Dec.*, ¶¶177-178.

2. Claim 1(a): “a network connection;”

Oberheide teaches its system is connected to a network. *Oberheide*, [0016], Fig. 1. For example, an “analysis cloud service may communicate with the VAC using any variety of protocols using any type of network interface deemed appropriate.” *Oberheide*, [0016] (“Such a network may be public or private in nature”).

3. Claim 1(b): “a processing device; and”

Oberheide teaches “a general or application specific processor” that executes instructions for performing *Oberheide*’s embodiments. *Oberheide*, [0031] (specifying that the instructions are “executed by computer-executable components preferably integrated with the vulnerability tool” of the computing device, such that the “computer-executable component is preferably a general or application specific processor”).

4. ***Claim 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:”***

Oberheide teaches its “system and methods of the preferred embodiment and variations thereof can be embodied and/or implemented at least in part as a machine configured to receive a computer-readable medium storing computer-readable instructions.” *Oberheide*, [0031]. “The instructions are preferably executed by computer-executable components” and the “computer-executable component is preferably a general or application specific processor[.]” *Id.*, *Dec.*, ¶¶184-185.

5. ***Claim 1(d): “accessing data through the network connection, the data comprising dark net information associated with exploits of a computer system, comprising:”***

As above, *Oberheide* teaches its system is connected to a network. *Supra*, Claim 1(a); *Oberheide*, [0016], Fig. 1. *Oberheide* also teaches a process for “access[ing] a resource storing a collection of exploitation scripts or routines.” *Oberheide*, [0017]. While *Oberheide* does not expressly state it accesses dark net exploit data over a network connection, *Patsos* teaches this.

Patsos teaches a security management system that “collects security information related to...exploits⁶...from a large number of sources” for local

⁶ *Patsos* defines an “exploit” as a “piece of code that realizes the existence of a particular vulnerability and takes some action[.]” *Patsos*, 295.

storage, including “underground web sources” such as “hacking web sites[.]” *Patsos*, 295 (“Information regarding exploits can be found [] directly in a large number of underground web sources”), 296 (“hacking web sites”). A POSITA would have understood hacking web sites on underground web sources include “dark net information associated with exploits[.]” *Dec.*, ¶190. Namely, the dark net is an “underground web source[.]” and hacking web sites on the dark net include information associated with exploits of a computer system. *Dec.*, ¶190.

It would have been obvious to modify *Oberheide* to update its local repository of exploit scripts with exploits from the dark net as taught by *Patsos* for several reasons. Adding dark net information associated with exploits of a computer system to *Oberheide*’s local repository of exploits would enhance *Oberheide*’s security coverage. *Dec.*, ¶¶191-192. *Oberheide*’s system would beneficially access the latest exploits being developed and marketed by malicious actors on the dark net to patch vulnerabilities associated with those exploits. *Dec.*, ¶192. Enhancing access to a broader selection of exploits would have provided additional tools (i.e., exploits) to gain the necessary privileges for patching vulnerabilities. *Dec.*, ¶193. Moreover, *Oberheide* recognizes many “different approaches” are necessary to patch certain vulnerabilities, and with that different “exemplary exploits” are used. *Oberheide*, [0025] (providing list of exemplary exploits). Updating *Oberheide*’s exploit repository with dark net exploits expands the range of approaches *Oberheide*’s

system may choose from, allowing it to resolve more vulnerabilities. *Dec.*, ¶¶194-196 (describing standards for using real-world exploit data and noting triviality to process dark net exploits compared to exploits from other sources).

A POSITA would have appreciated a reasonable expectation of success making this modification. *Oberheide* already teaches a network connection (see Claim 1(a)) through which *Oberheide*'s modified system would interface with the dark net, as taught by *Patsos*, to obtain exploits. *Dec.*, ¶197. To obtain exploits, *Patsos* utilizes known dark net crawlers to locally store exploit data. *Dec.*, ¶¶198-200. It would have been a simple addition well within the skill of a POSITA to implement known crawlers into *Oberheide*'s system to update *Oberheide*'s local exploit repository. *Dec.*, ¶¶201-204.

6. ***Claim 1(e): “obtaining a set of exploits from the dark net information, the set of exploits configured to penetrate the computer system;”***

Oberheide-Patsos teaches obtaining a set of exploits from the dark net information. *Supra*, Claim 1(d). Namely, *Oberheide*'s modified system utilizes a dark net crawler to obtain exploits to update its local exploit repository. *Id.*; *Oberheide*, [0017], Fig 6; *Patsos*, 295-296 (“[i]nformation related to exploits (and **actual exploit code**) can be **obtained from hacking web sites**”), 299 (providing “direct link to the actual exploit”).

These exploits are configured to penetrate the computer system. *Oberheide*'s

system “identif[ies] vulnerabilities and exploit[s] such vulnerabilities to enable capabilities to fix the vulnerabilities.” *Oberheide*, [0013]. To accomplish this, *Oberheide* uses the exploits to “[e]nter[] an operating mode of escalated privileges[,]” which will “grant[] the vulnerability tool the necessary privileges to patch the vulnerability.” *Id.*, [0024]-[0025]. In other words, *Oberheide* uses its stored exploits to penetrate its computer system to force a patch. *Dec.*, ¶207

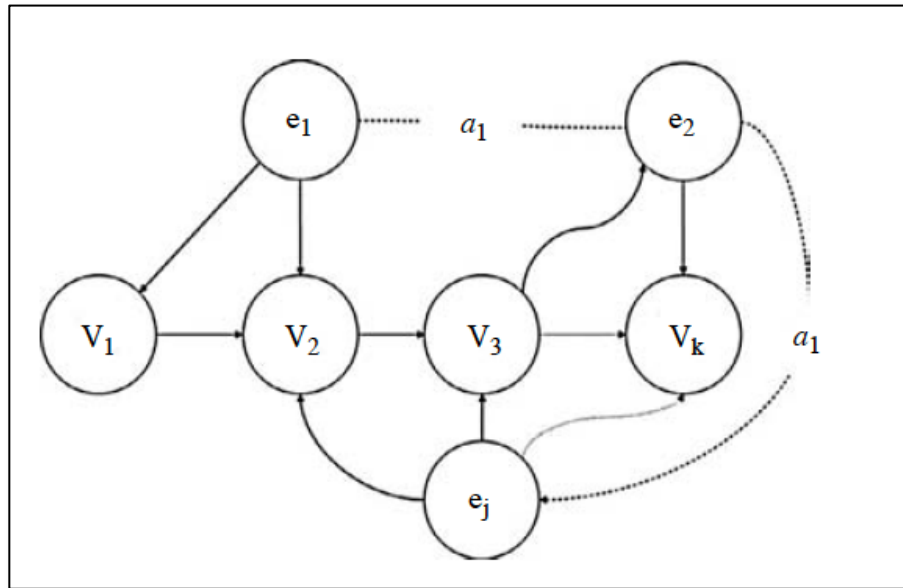
7. *Claim 1(f): “applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities;”*

Oberheide teaches a vulnerability assessment component (VAC) that returns a set of vulnerabilities using “any suitable approach to identifying a vulnerability[,]” such as “anti-virus and/or malware detection approaches as would be known by those skilled in the art.” *Oberheide*, [0015]; *see also id.*, [0019]-[0021], Fig. 6. While *Oberheide* teaches returning sets of vulnerabilities using known malware approaches (e.g., exploitation of known weaknesses), *Oberheide* does not expressly discuss using an exploit function to return a set of vulnerabilities. *Dec.*, ¶210.

Patsos teaches a “vulnerability exploit correlation module (VECM)” that correlates exploits to vulnerabilities. *Patsos*, 295-298 (“the correlation of exploits to vulnerability identifiers is performed by the vulnerability exploit correlation module VECM”). The VECM correlation module is “part[] of the system’s correlation engine.” *Id.*, 298. “This engine...provid[es] a full correlation between all attack components, i.e.,[] the vulnerabilities, the exploits that can potentially realize these

vulnerabilities and the associated countermeasures[.]” *Id.* (describing taking an exploit and correlating the exploit to vulnerabilities using vulnerability identifiers).

An example of this process is shown below in Figure 1:



Patsos, 295. *Patsos*’s system creates a “sample attack path (a_1)” and correlates exploits (e_j) with vulnerabilities (v_k). *Id.*, 295-296. In other words, *Patsos*’s VECM takes exploits as input and returns a set of vulnerabilities that correlate to those exploits. *Dec.*, ¶¶211-212.

Patsos’s teachings are consistent with the ’385 Patent’s description of its “exploit function[.]” *See, e.g.*, ’385 Patent, 10:6-9 (“An exploit function (denoted Ex) maps the attacker’s strategy to a set of vulnerable software applications, which can be easily obtained from darknet marketplace data”); *see also id.*, 3:59-4:5 (“many possible variations of an exploit function are possible” including associating vulnerabilities with exploits listed in online marketplaces). Similarly, *Patsos* maps

an attack path and correlates exploits with vulnerabilities. *Dec.*, ¶213.

It would have been obvious to modify *Oberheide*'s VAC to implement *Patsos*' VECM exploit function to identify the vulnerabilities associated with newly acquired exploits for patching. *Dec.*, ¶¶215-221. *Oberheide* teaches its "VAC may use any suitable approach to identify[] a vulnerability." *Oberheide*, [0015]. *Patsos*' VECM correlation engine employs an exemplary "suitable approach to identifying a vulnerability" that constructs attack paths and correlates vulnerabilities to exploits. *Dec.*, ¶¶216-217. More specifically, *Oberheide* teaches several vulnerability assessment techniques, including leveraging a remote analysis cloud service or using a "local analysis engine" to identify vulnerabilities locally. *Oberheide*, [0016]. A POSITA would have understood *Patsos*' VECM exploit function would facilitate this local analysis using known and reliable correlation techniques. *Dec.*, ¶217. (explaining benefits of computer-based correlation where *Patsos* explains this process was often a "human-oriented task" "performed by high-skilled security analysts").

Moreover, a POSITA would have been motivated to modify *Oberheide* with *Patsos* to enhance patching vulnerabilities. *Dec.*, ¶¶218-219. *Oberheide* aims to provide a variation of exploits to gain device privileges to patch vulnerabilities. *Dec.*, ¶218; *see generally*, *Oberheide*, [0025]. A POSITA would have understood that correlating vulnerabilities to exploits using *Patsos*'s VECM exploit function

would improve *Oberheide*'s process for gaining the different types of privileges needed for patching. *Dec.*, ¶219. And a POSITA would have appreciated success making this modification because *Oberheide* already teaches using "any suitable approach" to identify vulnerabilities and *Patsos*' VECM exploit function is a known suitable approach. *Dec.*, ¶220.

8. *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;"*

Oberheide-Patsos teaches identifying sets of vulnerabilities in the computer system. *Supra*, Claim 1(f); *Oberheide*, [0015], [0019]-[0021], Fig. 6. As detailed below, *Oberheide* creates a "constraint set" by constraining its vulnerability analysis to specific vulnerabilities based on: (1) vulnerabilities associated with a specific component of the system; or (2) system capabilities after exploitation of a vulnerability. 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.

Oberheide teaches that "[p]referably, vulnerabilities are identified for an entire system, but the vulnerabilities may alternatively be a set of vulnerabilities for a particular component or set of components of a device such as for a service or an application within an operating system of a computing device." *Oberheide*, [0020]. For example, the constrained set of vulnerabilities may be a particular application in

the system, such as “a browser, communication framework, or any suitable component[.]” *Oberheide*, [0020]. Moreover, *Oberheide* teaches constraining to specific vulnerabilities “based on the...enabled capabilities after the exploitation[.]” *Id.*, [0024]. In other words, *Oberheide* constrains its vulnerability analysis to vulnerabilities associated with a particular computer component or vulnerabilities that significantly alter the computer’s capabilities after exploitation. *Dec.*, ¶¶222-227.

Oberheide therefore teaches or suggests creating a constraint set comprising “a minimum set of dependencies to operate the computer system” such that 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. *Dec.*, ¶¶225-227 (describing exploited vulnerabilities that cause a user to lose operation of some or all of the computer system).

Oberheide teaches or suggests constraining its analysis to specific subsets of vulnerabilities from the set of system vulnerabilities (i.e., *constraint set*⁷), including

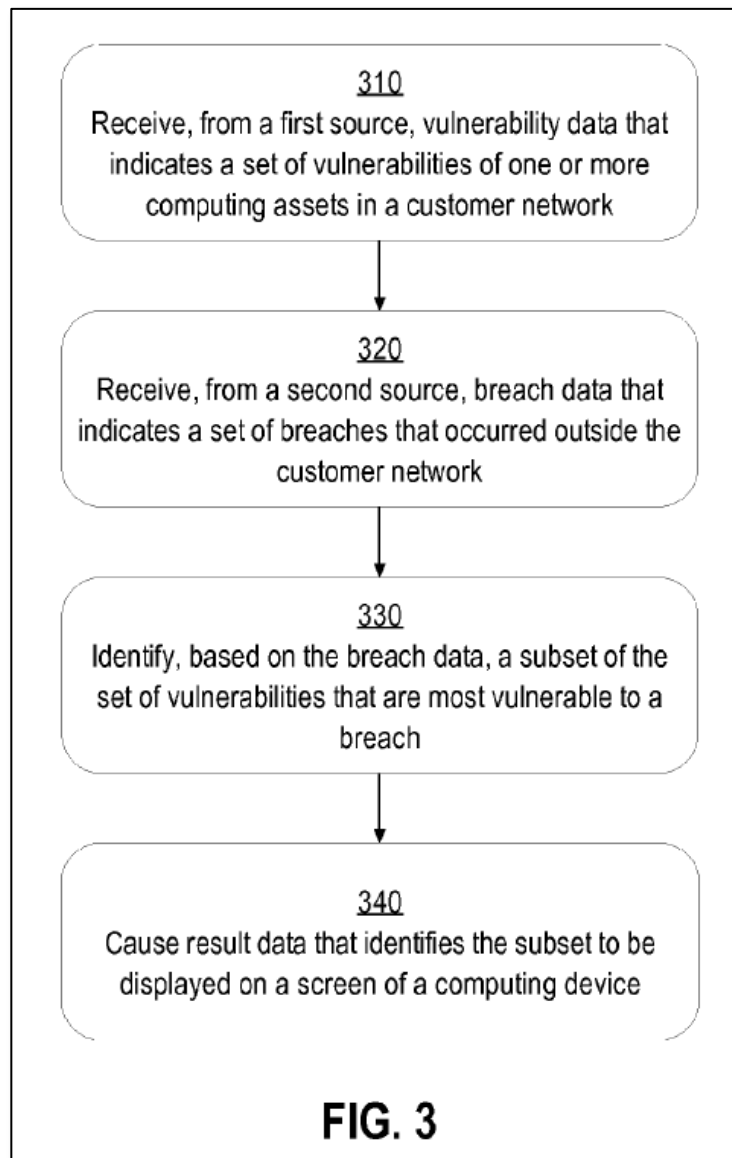
⁷ The ’385 Patent describes a “constraint set” as, for example, a “subset” of the total vulnerabilities such that the “vulnerabilities in a constraint set represent the

to vulnerabilities that would shut down the computer system. Should Patent Owner argue *Oberheide* does not teach an express process for “creating” this constraint set, *Roytman* teaches this.

Roytman teaches a “vulnerability threat management platform” that “rank[s] a set of vulnerabilities of a computing asset” and identifies “[a] subset of the set of vulnerabilities that are most vulnerable to the breach[.]” *Roytman*, Abstract, [0025], [0032]-[0033], [0064]-[0068], Fig. 3. A “breach is a successful exploit[.]” that is, “a breach is a successful attack on a computing asset by successfully exploiting a vulnerability of the computing asset.” *Id.*, [0032]. To identify the subset of vulnerabilities most vulnerable to exploitation, *Roytman*’s threat management platform receives vulnerability data, receives breach data, correlates vulnerability and breach data, and then constructs a “ranked or ordered list of vulnerabilities.” *Id.*, [0025], [0033]. *Roytman* may “identify one or more of the most important vulnerabilities of a customer (based on breach data associated with the vulnerabilities). For example, the top five vulnerabilities with the highest number of breaches (or the vulnerabilities that are associated with at least a threshold number of breaches) in a most recent time interval are identified[.]” *Id.*, [0059].

vulnerabilities required for some system functionality.” ’385 Patent, 3:29-32, 13:7-12.

An overview of this process is depicted below in Figure 3:



Id., Fig. 3. “FIG. 3 illustrates an example method for identifying a subset of vulnerabilities from a set of vulnerabilities that are most vulnerable to a breach based on breach data.” *Id.*, [0064]. “In step 330, a subset of the set vulnerabilities that are most vulnerable to a breach are identified based on the breach data received.” *Id.*, [0067], [0068].

A POSITA would have been motivated to modify *Oberheide* to rank vulnerabilities correlated with an exploit pursuant to *Roytman* to create “[a] subset of the set of vulnerabilities that are most vulnerable to the breach[.]” *Dec.*, ¶¶234-238. *Roytman*’s ranking techniques would effectuate and improve *Oberheide*’s process to constrain its vulnerability analysis to specific vulnerabilities based on “enabled capabilities after the exploitation[.]” *Dec.*, ¶¶235-236. It would have therefore been obvious to modify *Oberheide* to analyze and patch a subset of the highest ranked vulnerabilities ranked according to which vulnerabilities are most vulnerable to successful exploitation. *Dec.*, ¶¶235-236. Further, *Roytman*’s rankings include both risk score and optionally how important the component being exploited is. *Roytman*, [0044] (ranking accounting for whether a particular computing asset is “absolutely critical for...day-to-day operations”), [0050] (discussing “risk assessment unit”). Based on this, *Roytman* suggests creating a subset of vulnerabilities with the highest rankings, including vulnerabilities with the highest risks or that would render inoperable critical components in the system and/or the system as a whole. *Dec.*, ¶¶237-238. *Roytman*’s subset of highest ranked vulnerabilities thus includes vulnerabilities associated with “a minimum set of dependencies to operate the computer system[.]” *Dec.*, ¶238.

A POSITA would have had a reasonable expectation of success modifying *Oberheide* to rank vulnerabilities to create a subset of the set of vulnerabilities that

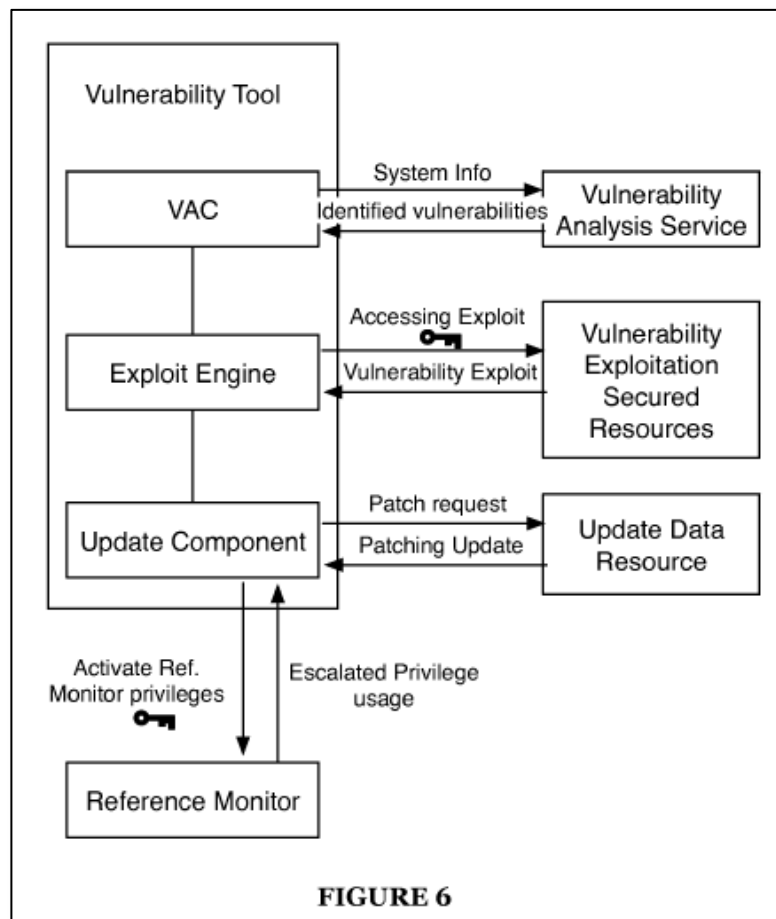
are most vulnerable to the breach. *Dec.*, ¶239. Such a modification would have required simple programming changes well within the skill of a POSITA to a system that already constrains analysis to specific vulnerabilities. *Dec.*, ¶239.

9. Claim 1(h): “applying the set of exploits to the constraint set of the computer system;”

Oberheide’s system applies exploits to the identified subset of vulnerabilities to determine which exploit will provide the necessary privileges for patching.

Oberheide, [0017]-[0019], [0024]-[0025]. This process is illustrated below in

Figure 6:



Oberheide, Fig. 6. Namely, after *Oberheide*'s modified system identifies the constraint set of vulnerabilities, *Oberheide* teaches an "exploit engine" that "functions to execute an exploitation process." *Id.*, [0017]. "The exploit engine preferably uses at least one of the vulnerabilities identified by the VAC to grant the vulnerability tool heightened permissions within the operating system" to facilitate a patch. *Id.*; *see also id.*, [0025] ("The exploit preferably grants the vulnerability tool the necessary privileges to patch the vulnerability"). To accomplish this, the "exploit engine may access a resource storing a collection of exploitation scripts or routines" such that an "exploitation routine preferably targets at least one vulnerability...identified by the analysis engine." *Id.*, [0017]; *see also id.*, [0019] (describing a "method for patching a device through exploitation"); [0024] ("Entering an operating mode of escalated privileges preferably includes selecting an appropriate vulnerability exploitation and executing the vulnerability exploitation").

In other words, *Oberheide*'s modified system selects sets of dark net exploits that will grant the vulnerability tool heightened permissions within the operating system to patch the vulnerabilities identified in its constraint set. *Dec.*, ¶243. *Oberheide* applies these sets of exploits against critical vulnerabilities in its constraint set to patch those vulnerabilities (i.e., *applying the set of exploits to the constraint set of the computer system*). *Dec.*, ¶¶241-244.

10. 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”

Oberheide’s modified system teaches analyzing the application of the set of exploits on the computer system to detect a particular vulnerability to patch, such as the highest ranked vulnerability. *Supra*, Claims 1(g)-1(h); *Oberheide*, [0024]-[0025].

As above, *Oberheide* exploits the vulnerabilities in its constraint set to gain access to privileges necessary to patch those vulnerabilities. *Supra*, Claim 1(h). *Oberheide*’s system is therefore “analyzing the application of the set of exploits on the computer system” by analyzing which exploits provide the system with the requisite privileges for patching a specific vulnerability. *Dec.*, ¶246. As modified by *Roytman*, *Oberheide*’s system would analyze and apply exploits to identify the highest ranked vulnerability to patch. *Dec.*, ¶247; *Roytman*, [0033] (“the vulnerability threat management platform may then provide a ranked or ordered list of vulnerabilities which represent the order in which the vulnerabilities should be addressed, such that the vulnerability that poses the most significant threat is addressed first and the one that poses the least significant threat is addressed last”). *Oberheide*’s system thus “detect[s] a particular vulnerability of the computer system” by analyzing the exploitation of those vulnerabilities to identify the highest ranked vulnerability for patching. *Dec.*, ¶¶245-248.

11. ***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.”***

Oberheide’s vulnerability tool includes an “update component” that applies “a patch or system update to resolve or reduce an existing vulnerability.” *Oberheide*, [0018], [0019]. A “patch” or “system update” alters a configuration of the computer system. *Dec.*, ¶249. The computer system is altered in response to the analysis of the application of the set of exploits. *Id.* *Oberheide* teaches the applied exploits “grant[] the vulnerability tool the necessary privileges to patch the vulnerability.” *Oberheide*, [0025]. For instance, after applying exploits, *Oberheide*’s system enters an escalated privileges mode and updates the system with a “vulnerability resolution” to resolve or reduce vulnerabilities. *Id.*, [0026]. “Updating the system preferably includes downloading, accessing, and/or selecting a patch or software update that can be applied to the system.” *Id.* And “at least the vulnerability that is exploited is patched.” *Id.* Fixing, updating, or reducing vulnerabilities reduces the potential damage of a cyberattack. *Dec.*, ¶250.

B. Claim 8

1. ***Claim 8(Pre): “A method for improving a computer device, the method comprising:”***

To the extent the preamble is limiting, *Oberheide* teaches “[a] method for improving a computer device[.]” *Supra*, Claim 1(Pre); *Oberheide*, [0019] (disclosing “a **method** for patching a device”); *Oberheide*, claims 1, 11, 19.

Protecting a computer device from vulnerabilities improves the computer device.

Dec., ¶¶177, 252.

2. ***Claim 8(a): “accessing data comprising dark net information associated with a computer system;”***

Oberheide-Patsos teaches “accessing data comprising dark net information associated with a computer system[.]” *Supra*, Claim 1(d); *Patsos*, 295-296 (describing collecting security information related to exploits from underground web sources such as “hacking web sites”).

3. ***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;”***

Oberheide-Patsos teaches obtaining a set of exploits from the dark net information using a dark net crawler. *Supra*, Claim 1(e). These dark net exploits are “configured to bypass a security feature of the computer system” to “grant[] the vulnerability tool the necessary privileges to patch the vulnerability.” *Oberheide*, [0024]-[0025]; *Dec.*, ¶¶254-255 (explaining Claim 8(b)’s “bypass” and Claim 1(e) “penetrate”: both refer to exploits that circumvent security features or break into computer systems and software).

4. ***Claim 8(c): “applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities;”***

Oberheide-Patsos teaches applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities. *Supra*, Claim 1(f).

5. ***Claim 8(d)(1): “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,”***

Oberheide-Patsos teaches “creating a constraint set of vulnerabilities of the computer system” by ranking and constraining its vulnerability analysis to specific vulnerabilities. *Supra*, Claim 1(g). This “constraint set” comprises “a minimum set of dependencies to operate the computer system” such that *Oberheide-Patsos*’ 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. *Supra*, Claim 1(g).

6. ***Claim 8(d)(2): “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;”***

Oberheide’s system applies exploits to the identified subset of vulnerabilities to determine which exploit will provide the necessary privileges for patching. *Supra*, Claim 1(h); *Oberheide*, [0017]-[0019], [0024]-[0025]. *Oberheide* determines the effect of the set of exploits on the constraint set of vulnerabilities of the computer system by determining which exploit provides privileges for patching. *Id.*; *Dec.*, ¶¶241-244, 258.

7. ***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; and”***

Oberheide’s modified system teaches analyzing the application of the set of exploits on the computer system to detect a particular vulnerability to patch, such as the highest ranked vulnerability. *Supra*, Claim 1(i).

8. ***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.”***

Oberheide’s vulnerability tool includes an “update component” that applies “a patch or system update to resolve or reduce an existing vulnerability” such that fixing, updating, or reducing vulnerabilities reduces the potential damage of a cyberattack. *Supra*, Claim 1(j); *Oberheide*, [0018], [0019].

- C. **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.”**

As above in Claim 1(j), *Oberheide* installs patches to a software component of the computer system which *alter the configuration of the computer system*. See Claim 1(j). *Oberheide* teaches installing patches for the vulnerable component that was exploited. *Oberheide*, [0026] (“us[ing] the privileges granted through the exploit to resolve or reduce vulnerabilities[,]” and “**at least the vulnerability that is exploited is patched**”). Particularly, “[u]pdating the system preferably includes **downloading, accessing, and/or selecting a patch or software update** that can

be applied to the system...The updating can target only the identified vulnerability[.]” *Oberheide*, [0026]; *Dec.*, ¶261.

D. Claim 15

1. ***Claim 15(Pre): “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:”***

To the extent the preamble is limiting, *Oberheide* teaches one or more non-transitory tangible computer-readable storage media storing computer-executable instructions for performing a computer process on a machine. *Supra*, Claim 1(c).

2. ***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;”***

Supra, Claims 1(e), 8(b) (identical language).

3. ***Claim 15(b): “applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities;”***

Supra, Claims 1(f), 8(c) (identical language).

4. ***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;”***

Supra, Claims 1(g), 8(d)(1) (identical language).

5. ***Claim 15(d): “applying the set of exploits to the constraint set of vulnerabilities of the computer system;”***

Oberheide’s modified system teaches applying the set of exploits to the constraint set of vulnerabilities of the computer system. *Supra*, Claim 1(h).

6. ***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”***

Oberheide’s modified system teaches analyzing the application of the set of exploits on the computer system to detect a particular vulnerability to patch, such as the highest ranked vulnerability. *Supra*, Claims 1(i), 8(e).

7. ***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.”***

Supra, Claims 1(j), 8(f) (identical language).

VII. GROUND 3: CLAIMS 2-6, 9-12, 16, AND 17 ARE OBVIOUS OVER OBERHEIDE, PATSOS, ROYTMAN, AND JUELS

- A. ***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.”***

Oberheide-Patsos-Roytman analyzes and applies exploits against the highest ranked vulnerabilities. *Supra*, Claim 1(i); *Dec.*, ¶¶245-248, 275. To rank vulnerabilities, *Roytman* teaches that its “vulnerability threat management platform may provide a quantified measure, known as a risk score herein, to indicate how vulnerable a particular asset may be of being successfully exploited or breached.” *Roytman*, [0034]. For the reasons above, it would have been obvious to implement *Roytman’s* ranking process, including calculating risk scores, into *Oberheide’s* system. *Supra*, Claim 1(i); *Dec.*, ¶¶245-248, 276. However, *Oberheide’s* modified

system does not expressly quantify risk level related to a cost to obtain exploits. It would have been obvious to include a cost metric for exploits as taught by *Juels*, when generating a risk score in *Oberheide's* modified system.

Juels teaches a game-theoretic model where an attacker acts to compromise a target computing device, and a defender acts to protect the target device. *Juels*, 2:32-38, 13:6-16. Because “[a]n action by either side carries a cost[,]” the actions taken must be cost-effective. *Juels*, 13:12-24. In one example, the attacker chooses between 1) a less expensive, yet less effective published exploit; or 2) a more expensive, yet more effective zero-day exploit. *Juels*, 13:25-29. A POSITA would have recognized, as taught by *Jules*, the well-known link between the effectiveness of the exploit (i.e., *a risk level posed to the computer system*) is *related to a cost to obtain the set of exploits*. *Dec.*, ¶277.

A POSITA would have been motivated to further modify *Oberheide* to consider exploit cost, as taught by *Juels*, when generating a risk score. *Juels*, 13:21-34 (teaching zero-day exploits cost more but are more effective than less-expensive published exploits); *Dec.*, ¶¶279-284. This modification would increase efficiency of identifying, exploiting, and patching the highest priority vulnerabilities in *Oberheide's* system. *Dec.*, ¶¶280-281. For example, the modified system would distinguish a more effective, but cost-prohibitive exploit, from a slightly less effective, but cheap exploit. *Dec.*, ¶¶280-281. This would enhance selectivity of

the vulnerabilities to be patched, leading to an overall enhanced security process.

Dec., ¶281.

A POSITA would have appreciated a reasonable likelihood of success modifying *Oberheide-Patsos-Roytman* to generate a risk score for a vulnerability that accounts for the cost to exploit that vulnerability. *Dec.*, ¶283. *Roytman*'s risk score already accounts for how likely the vulnerability may be exploited and how important the system component is that could be exploited. *Dec.*, ¶283. Modifying this risk score to account for a third metric, cost to exploit, would be well within the skill of a POSITA at the time. *Dec.*, ¶283.

B. 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.”

As above, *Oberheide-Patsos-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. *Supra*, Ground 2, Claim 1(g). *Roytman* ranks vulnerabilities in part by receiving “breach data, sourced from across the internet, and correlating that data with vulnerabilities discovered in computing assets[.]” *Roytman*, [0030], [0032] (“a breach is a successful exploit”). Breach data “indicates a set of breaches that occurred outside the customer network” and is used to determine “the likelihood that an exploit of a particular vulnerability of a particular

asset will be successful.” *Id.*, [0025], [0030]. Breach data therefore includes “Internet data” that is “based on information obtained about the set of exploits” (i.e., breaches), and is aligned with the vulnerabilities such that breach data is used to determine, in part, the vulnerability’s rank. *Dec.*, ¶285 (understanding plain meaning of “aligning Internet data” to at least include correlating Internet data with the vulnerabilities based on information obtained about the set of exploits).

A POSITA would have been motivated to modify and would have appreciated a reasonable expectation of success in modifying *Oberheide* with *Roytman* for the same reasons above. *Supra*, Claim 1(g); *Dec.* ¶¶234-240, 288.

C. 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.”

Oberheide’s modified system obtains exploits through a network connection using, e.g., a dark net crawler. *Supra*, Claim 1(d) (describing *Patsos*’ techniques for obtaining exploit code from underground websites); *Dec.*, ¶¶197-200. An automated crawler includes one or more algorithms for obtaining exploits from various dark net websites. *Dec.*, ¶¶290-292. Internet crawlers operate algorithmically; they require a defined set of logical instructions to systematically navigate web content and extract data. *Dec.*, ¶¶290-292. Indeed, dark net crawling algorithms were well known in the art. Ex. 1015 (*Olston*), 230; Ex. 1018 (*Fu*), 1218; *Dec.*, ¶293. A POSITA would have understood *Oberheide*’s modified system

would have implemented known algorithms to crawl the dark net and obtain exploits as taught by *Patsos. Dec.*, ¶293.

D. 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.”

As discussed above, *Oberheide*’s modified system benefits from updating its local repository of exploit scripts with exploits from the dark net. *Supra*, Claim 1(d). Moreover, *Oberheide*’s modified system determines the cost to obtain the set of exploits, as taught by *Juels*, for ranking vulnerabilities (e.g., by likelihood of exploitation). *Supra*, Claim 2. Cost information would have been collected by the dark net crawler (i.e., *one or more algorithms*) while obtaining the set of exploits. *Supra*, Claims 1(e), 2; *Dec.*, ¶295. As such, *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. *Dec.*, ¶295.

It would have been obvious to create a budget and obtain the set of exploits constrained by the determined cost to obtain the set of exploits. *Juels*’ “Host Takeover” version of its game-theoretic model involves an attacker trying to “compromise [a] device by exploiting a software vulnerability[.]” *Juels*, 13:5-34. The compromises available to the attacker for exploitation have “varying costs[.]” such as a “zero-day exploit” costing more than “a published exploit,” forcing the players to “choose among a set of actions with varying costs and effectiveness.” *Id.*

Juels specifies the players have certain “[b]udgets” they can’t exceed, providing an “upper bounds...on the players’ respective move rates.” *Id.*, 36:28-33. For example, a player may be limited in their play rate by a “financial budget[.]” *Id.*, 37:4-9. In other words, *Juels*’s model links the cost of an exploit to a financial budget that constrains the player’s ability to obtain exploits. *Dec.*, ¶¶296-297.

A POSITA would have understood, based on *Juels*’s teachings, that for obtaining costly exploits in *Oberheide*’s modified system, it would be beneficial to have a budget constraining the system from purchasing all exploits it found. *Dec.*, ¶¶300-302. Such a system would otherwise suffer from runaway purchasing power, spending an unforeseen amount on updating exploits in its repository. *Dec.*, ¶¶300-302. Moreover, by constraining the system by cost, *Oberheide* would only purchase exploits it knew would substantially benefit the system. *Dec.*, ¶¶300-302. For example, in ranking exploits, *Oberheide* may identify two top exploits providing nearly the same privilege level, or risk level, required for patching a subset of vulnerabilities. However, *Oberheide*’s system, including cost, determines the top-ranked exploit was much more expensive than the second-ranked exploit. In this example, *Oberheide*’s system would benefit by purchasing and reranking the second-ranked exploit over the top-ranked exploit. *Dec.*, ¶302.

A POSITA would have had a reasonable likelihood of success in modifying *Oberheide* in this way for the same reasons above. *Supra*, Claim 2.

E. 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.”

Oberheide’s system, as modified by *Juels*, teaches obtaining the set of exploits based on maximizing the quantifying of the risk level posed to the computer system related to the cost to obtain the set of exploits. *Supra*, Claim 5 (providing example of maximizing the quantifying of the risk level of two exploits as related to the cost to obtain the set of exploits); *Dec.*, ¶305.

More specifically, *Oberheide-Patsos-Roytman-Juels* teaches quantifying a risk level posed to the computer system related to the cost to obtain the set of exploits. *Supra*, Claim 2. For example, the risk level “indicates how vulnerable a particular asset may be of being successfully exploited or breached.” *Roytman*, [0069]. “[T]he risk score may represent a value between 0 and 1000, where the **higher the number, the more at risk that particular computing asset** or group of assets is of being successfully exploited or breached.” *Roytman*, [0034].

The attacker in *Juels’* game-theoretic model is best able to exploit a software vulnerability by obtaining exploits that will be “more effective” against the targeted vulnerability. *Juels*, 13:6-15, 13:21-34. Thus, quantifying the risk level is “**maximized**” by obtaining exploits, as permitted by cost, that may most effectively exploit the computer system.

As discussed in Claim 2, a POSITA would have found it obvious to modify *Oberheide*, *Roytman*, and *Juels*. *Supra*, Claim 2. The “more effective” an exploit is against a targeted vulnerability, such as a computer system or asset, the more vulnerable that target is to being “successfully exploited or breached” by that exploit. *Juels*, 13:25-29; *Roytman*, [0034]. This, in turn, means a higher risk level is assigned to that targeted vulnerability. *Dec.*, ¶¶308-309. As discussed above in Claims 2 and 5, however, obtaining exploits from the dark net comes at a cost, and the budget is not limitless. *See* Claims 2 and 5. With that backdrop, a POSITA would have understood that “*maximizing [. . .] the risk level posed to the computer system*” is limited by the effectiveness of the exploits the POSITA may obtain based on the exploit budget. *Dec.*, ¶310.

A POSITA would have appreciated a reasonable expectation of success further modifying *Oberheide*’s modified system to maximize the risk level versus costs. *Roytman*’s “risk score” is determined “based upon several contextual factors[.]” *Roytman*, [0070]. *Roytman*’s factors are non-exhaustive, but include “the prevalence (or number) of available exploits for each vulnerability, popularity of the computing asset or [. . .] difficulty of exploiting the vulnerability[.]” *Id.* It would have required simple programming changes to consider exploit cost as a factor. *Dec.*, ¶¶311-312. This cost may limit the number and/or type of exploits purchased, as well as the effectiveness of the exploits available based on the budget.

A POSITA under these constraints would also understand that these factors directly impact the risk level assigned to the target computer system. *Supra*, Claim 2.

- F. 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.”**

Oberheide’s modified system teaches quantifying a “risk score” related to a cost to obtain a set of exploits. *Supra*, Claim 2.

- G. 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.”**

Oberheide’s modified system teaches applying one or more algorithms to obtain the set of exploits. *Supra*, Claim 4. To the extent Claim 10 is interpreted to require the constraint set of vulnerabilities of the computer system to be determined before obtaining the set of exploits, *Oberheide* teaches this. *Oberheide* teaches “vulnerabilities are identified...for a particular component or set of components” (i.e., constraint set of vulnerabilities). *Oberheide*, [0020]. *Oberheide*’s method then “select[s] an appropriate vulnerability exploitation and execut[es] the vulnerability exploitation” to gain “escalated privileges” through the vulnerability to “fix or lessen the threat.” *Id.*, [0024] (teaching a situation with multiple vulnerabilities where an exploit is selected from a group of exploits “based on the...enabled capabilities after the exploitation[.]”). *Oberheide* therefore obtains a set of exploits from its repository

based on the constraint set of vulnerabilities of the computer system. *Dec.*, ¶¶315-316.

As modified by *Patsos*, *Oberheide* obtains the vulnerabilities “from one or more darknet marketplaces on the Internet.” *Supra*, Claim 1(d); *Dec.*, ¶¶189-204, 312 (describing how “hacking web sites” in *Patsos* would be equivalent to a “darknet marketplace” as claimed). To the extent that Patent Owner alleges a “hacking web site” providing “exploits (and actual exploit code)” is not a darknet marketplace, a POSITA would have found it obvious to modify *Oberheide* to obtain exploits based on cost from a dark web “marketplace[.]” *Supra*, Claim 5; *Dec.*, ¶¶317-318. Indeed, dark net marketplaces were widely known repositories for exploit data with well-understood algorithms available in the industry to obtain exploits from those marketplaces. *Dec.*, ¶318; Ex. 1021 (*Miller*), 2; Ex. 1042 (*Kumar*).

H. 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.”

Oberheide’s modified system teaches crawling exploits from the dark net based on risk and cost and therefore teaches the method of Claim 11. *Supra*, Claim 5.

- I. Claim 12: “The method of claim 11, where 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.”**

Oberheide’s modified system teaches the method of Claim 12. *See* Claim 6.

- J. 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.”**

Oberheide’s modified system teaches one or more non-transitory computer-readable storage media storing instructions for performing this process. *Supra*, Claim 2.

- K. 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.”**

Oberheide’s modified system teaches applying one or more algorithms to obtain the set of exploits from one or more darknet marketplaces on the Internet. *Supra*, Claims 1(e), 4, 10; *Dec.*, ¶¶290-294, 315-319, 323.

VIII. GROUND 4: CLAIMS 7 AND 13 ARE OBVIOUS OVER *OBERHEIDE*, *PATSOS*, *ROYTMAN*, AND *WOLFF*

- A. 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.”**

Oberheide teaches “altering the configuration of the computer system” by applying a patch or software update to a software component of the computer

system that is utilized by the set of exploits to attack the computer system. *Supra*, Claim 1(j); *Oberheide*, [0026] (“[p]referably at least the vulnerability that is exploited is patched” and “patch[ing] **the vulnerable software component**”). While *Oberheide*’s system “enable[s] capabilities to fix, update, and/or reduce the vulnerabilities[,]” (*Oberheide*, [0013]) *Oberheide* does not expressly teach removal of the exploited software component during patching. It would have been obvious to modify *Oberheide* pursuant to *Wolff* to remove the exploited software component.

Wolff teaches remediating a client computer by applying a remedial exploit to a software vulnerability. *Wolff*, [0014]-[0015], [0028]. The remedial exploit carries a self-healing payload, which “inserts itself into the client computer [and] take[s] steps to remedy the vulnerability[.]” *Id.*, [0016], [0027] (“[t]he remedial exploit may rely on the same vulnerability that is the intended target of the original malicious exploit”). Remedies include “**removing software[.]**” *Id.*, [0028]. Removing a software component alters the computer system to “ameliorate or eliminate the malicious exploit’s ability to cause harm[.]” *Id.*, [0028], [0047].

A POSITA would have been motivated to modify *Oberheide* in light of *Wolff*’s teaching to remove the software to prevent future exploitation. *Dec.*, ¶¶333-337. This modification enhances *Oberheide*’s options to facilitate remediation by including other actions such as removing the exploited software. *Wolff*, [0028].

Moreover, *Oberheide*'s preferred actions to "resolve or reduce vulnerabilities" include "downloading, accessing, and/or selecting a patch or software update that can be applied to the system." *Oberheide*, [0026], [0013] (teaching employing either "a persistent fix [or] a non-persistent computing system patch"). Merely reducing a vulnerability or applying a temporary patch may not ultimately resolve the issue. *Dec.*, ¶¶334-335. A POSITA would thus have been motivated to add an option to remove a vulnerable software component, as taught by *Wolff*, in scenarios where a simple patch will not fully resolve the issue. *Dec.*, ¶¶334-335.

A POSITA would have had a reasonable expectation of success making this modification because both *Oberheide* and *Wolff* provide methods of exploiting vulnerabilities for self-healing purposes. *Oberheide*, Abstract, [0013], [0019], Fig. 6; *Wolff*, [0014]-[0016], [0027]-[0028]. As such, modifying *Oberheide*'s method of self-healing to include the ability to remove the vulnerable software component would have required simple programming techniques known to a POSITA (e.g., invoking existing OS tools used to uninstall a software package). *Dec.*, ¶336. This modification would have used known techniques for removing exploited software components to improve *Oberheide*'s system and yield predictable results. *Dec.*, ¶336.

- B. 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.”**

Oberheide as further modified by *Wolff* teaches altering the configuration of the computer system by removing a software component of the computer system that is utilized by the set of exploits to attack the computer system. *Supra*, Claim 7.

IX. CONCLUSION

For the foregoing reasons, Petitioner respectfully requests *inter partes* review of the Challenged Claims.

Respectfully submitted,

BY: /s/ Paul R. Hart

Paul R. Hart, Reg. No. 59,646

COUNSEL FOR PETITIONER

X. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(A)(1)

A. Real Party-in-Interest

Petitioner is the real party-in-interest. 37 C.F.R. § 42.8(b)(1)

B. Related Matters Under 37 C.F.R. § 42.8(b)(2)

The '385 Patent is presently the subject of a patent infringement lawsuit filed against Petitioner in the United States District Court for the Western District of Texas, *Skysong Innovations, LLC v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.*, Case No. 7:25-cv-00040-DC-DTG. The '385 Patent has also been asserted against Fortinet, Inc. in the United States District Court for the Eastern District of Texas, *Skysong Innovations, LLC v. Fortinet, Inc.*, Case No. 2:25-cv-00098.

C. Lead and Back-Up Counsel

Petitioner provides the following designation and service information for lead and back-up counsel. 37 C.F.R. § 42.8(b)(3) and (b)(4).

Lead Counsel	Back-Up Counsel
Paul R. Hart (Reg. No. 59,646) Paul.Hart@eriseip.com <u>PTAB@eriseip.com</u> <u>Postal and Hand-Delivery Address:</u> ERISE IP, P.A. 717 17 th St., Suite 1400 Denver, Colorado 80202 Telephone: (913) 777-5600 Fax: (913) 777-5601	Adam P. Seitz (Reg. No. 52,206) Adam.Seitz @eriseip.com <u>Postal and Hand-Delivery Address:</u> ERISE IP, P.A. 7015 College Blvd., Suite 700 Overland Park, Kansas 66211 Telephone: (913) 777-5600 Fax: (913) 777-5601
	Hunter Horton, <i>pro hac vice</i> Hunter.Horton@eriseip.com

	<u>Postal and Hand-Delivery Address:</u> ERISE IP, P.A. 7015 College Blvd., Suite 700 Overland Park, Kansas 66211 Telephone: (913) 777-5600 Fax: (913) 777-5601 Nathan Byers, (Reg. No. 81,095) Nathan.Byers@eriseip.com <u>Postal and Hand-Delivery Address:</u> ERISE IP, P.A. 717 17th St., Suite 1400 Denver, Colorado 80202 Telephone: (913) 777-5600 Fax: (913) 777-5601
--	--

D. 37 C.F.R. § 42.8(b)(4) – Service Information

Please address all correspondence to the lead and back-up counsel as shown above. Petitioner consents to electronic service by e-mail at the e-mail addresses provided above.

CLAIM LISTING APPENDIX
U.S. Patent No. 10,313,385 for Claims 1-17

Claim Designation	Claim Language
Claim 1(Pre)	A system for protecting a computer system from attack, the system comprising:
Claim 1(a)	a network connection;
Claim 1(b)	a processing device; and
Claim 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:
Claim 1(d)	accessing data through the network connection, the data comprising dark net information associated with exploits of a computer system, comprising:
Claim 1(e)	obtaining a set of exploits from the dark net information, the set of exploits configured to penetrate the computer system;
Claim 1(f)	applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities;
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;
Claim 1(h)	applying the set of exploits to the constraint set of the computer system;
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
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.
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 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 Designation	Claim Language
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 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	The system of claim 5, where 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	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(Pre)	A method for improving a computer device, the method comprising:
Claim 8(a)	accessing data comprising dark net information associated with a computer system;
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;
Claim 8(c)	applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities;
Claim 8(d)(1)	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,
Claim 8(d)(2)	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;
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; and
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.

Claim Designation	Claim Language
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 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	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	The method of claim 11, where 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	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	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(Pre)	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:
Claim 15(a)	obtaining a set of exploits from the dark net information, the set of exploits configured to bypass a security feature of a computer system;
Claim 15(b)	applying an exploit function which takes the set of exploits as input and returns a set of vulnerabilities;
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;

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

APPENDIX OF EXHIBITS

Exhibit 1001	U.S. Patent No. 10,313,385 to Shakarian et al. (“’385 Patent”)
Exhibit 1002	File History of U.S. Patent No. 10,313,385 (“’385 Patent File History”)
Exhibit 1003	Declaration of Erez Zadok, Ph.D. (“Dec.”)
Exhibit 1004	U.S. Patent Application Publication No. 2016/0134653 to Vallone et al. (“Vallone”)
Exhibit 1005	U.S. Patent Application Publication No. 2014/0245450 to Oberheide et al. (“Oberheide”)
Exhibit 1006	Dimitrios Patsos, Sarandis Mitropoulos, and Christos Douligeris, <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) (“Patsos”)
Exhibit 1007	U.S. Patent Application Publication No. 2015/0237065 to Roytman et al. (“Roytman”)
Exhibit 1008	U.S. Patent No. 9,471,777 to Juels et al. (“Juels”)
Exhibit 1009	U.S. Patent Publication No. 2013/0198848 to Wolff (“Wolff”)
Exhibit 1010	U.S. Patent Application Publication No. 2014/0137257 to Martinez et al. (“Martinez”)
Exhibit 1011	Karen Scarfone, Murugiah Souppaya, Amanda Cody, and Angela Orebaugh, <i>Technical Guide to Information Security Testing and Assessment</i> , NIST Special Publication 800-115 (Sept. 2008) (“Scarfone”)
Exhibit 1012	Microsoft Computer Dictionary, Microsoft Press, 4th ed. 201 (1999) (“Microsoft”)
Exhibit 1013	Assane Gueye and Vladimir Marbukh, <i>A Game-Theoretic Framework for Network Security Vulnerability Assessment and Mitigation</i> , In International Conference on Decision and Game Theory for Security (GameSec), (2012) (“Gueye”)
Exhibit 1014	Ritesh Kumar Tiwari and Kamalakar Karlapalem, <i>Cost Tradeoffs for Information Security Assurance</i> , Workshop on the Economics of Information Security (2005) (“Tiwari”)
Exhibit 1015	Christopher Olston and Marc Najork, <i>Web Crawling</i> , Foundations in Information Retrieval, Vol. 4, No. 3, 175-246 (2010) (“Olston”)
Exhibit 1016	Md. Abu Kausar, V.S. Dhaka, and Sanjeev Kumar Singh, <i>Web Crawler: A Review</i> , International Journal of Computer Applications, Vol. 63, No. 2, 31-36 (2013) (“Kausar”)

Exhibit 1017	Seyed M. Mirtaheri, Mustafa Emre Dincturk, Salman Hooshmand, Gregor V. Bochmann, Guy-Vincent Jourdan, and Iosif Viorel Onut, <i>A Brief History of Web Crawlers</i> , arXiv, https://doi.org/10.48550/arXiv.1405.0749 (2014) (“Mirtaheri”)
Exhibit 1018	Tianjun Fu, Ahmed Abbasi, and Hsinchun Chen, <i>A Focused Crawler for Dark Web Forums</i> , Journal of the American Society for Information Science and Technology, Vol. 611, No. 6, 1213-1231 (2010) (“Fu”)
Exhibit 1019	Jiang-Ming Yang, Rui Cai, Chunsong Wang, Hua Huang, Lei Zhang, and Wei-Ying Ma, <i>Incorporating Site-Level Knowledge for Incremental Crawling of Web Forums: A List-wise Strategy</i> , Proceedings of the 15 th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2009) (“Yang”)
Exhibit 1020	Jorge Lucángeli Obes, Carlos Sarraute, and Gerardo Richarte, <i>Attack Planning in the Real World</i> , arXiv:1306.4044 [cs.CR] (June 19, 2013) (“Obes”)
Exhibit 1021	Charlie Miller, <i>The legitimate vulnerability market: inside the secretive world of 0-day exploit sales</i> , Workshop on the Economics of Information Security (May 6, 2007) (“Miller”)
Exhibit 1022	Michael Chertoff and Toby Simon, <i>The Impact of the Dark Web on Internet Governance and Cyber Security</i> , Global Commission on Internet Governance Paper Series No. 6 (Feb. 2015) (“Chertoff”)
Exhibit 1023	Marco Carvalho and Choh Man Teng, <i>Automatic Discovery of Attack Messages and Pre- and Post-Conditions for Attack Graph Generation</i> , Proceedings of the International Conference on Information-Warfare and Security, 378-387 (2010) (“Carvalho”)
Exhibit 1024	Sushil Jajodia, Steven Noel, and Brian O’Berry, <i>Topological Analysis of Network Attack Vulnerability</i> , Managing Cyber Threats, 247-266 (2005) (“Jajodia”)
Exhibit 1025	U.S. Patent Application Publication No. 2006/0021029 to Brickwell et al. (“Brickwell”)
Exhibit 1026	Shinsuke Miwa, Toshiyuki Miyachi, Masashi Eto, Masashi Yoshizumi, and Yoichi Shinoda, <i>Design and Implementation of an Isolated Sandbox with Mimetic Internet used to Analyze Malwares</i> , DETER (2007) (“Miwa”)
Exhibit 1027	Vincent Nicomette, Mohamed Kaaniche, Eric Alata, and Matthieu Herrb, <i>Set-up and Deployment of a High-Interaction Honeypot: Experiment and Lessons Learned</i> , Journal of Computer Virology, Vol. 7, No. 2, 143-157 (2011) (“Nicomette”)

Exhibit 1028	U.S. Patent Application Publication No. 2015/0040230 to Oliphant et al. (“ <i>Oliphant</i> ”)
Exhibit 1029	Joint Task Force, <i>Security and Privacy Controls for Federal Information Systems and Organizations</i> , NIST Special Publication 800-53 (2013) (“ <i>Joint Task Force</i> ”)
Exhibit 1030	Plaintiff’s Preliminary Infringement Contentions, <i>Skysong Innovations, LLC v. CrowdStrike, Inc., CrowdStrike Holdings, Inc.</i> , C.A. No. 7:25-cv-00040 (2025) (“ <i>Infringement Contentions</i> ”)
Exhibit 1031	Declaration of June Munford (“ <i>Munford Dec.</i> ”)
Exhibit 1032	Europol European Law Enforcement Agency, <i>Review</i> , General Report on Europol Activities (2014) (“ <i>Europol</i> ”)
Exhibit 1033	U.S. Patent Application Publication No. 2016/0259943 to Murthy et al. (“ <i>Murthy</i> ”)
Exhibit 1034	Yulei Zhang, Shuo Zeng, Chun-Neng Huang, Li Fan, Ximing Yu, Yan Dang, Catherine A. Larson, Dorthy Denning, Nancy Roberts, and Hsinchun Chen, <i>Developing a Dark Web Collection and Infrastructure for Computational and Social Sciences</i> , Proceedings of the 2010 IEEE International Conference on Intelligence and Security Informatics (2010) (“ <i>Zhang</i> ”)
Exhibit 1035	Michael Felderer and Ina Schieferdecker, <i>A taxonomy of risk-based testing</i> , International Journal on Software Tools for Technology Transfer, Vol. 16, Issue 5, 559-568 (July 23, 2014) (“ <i>Felderer</i> ”)
Exhibit 1036	Massimiliano Albanese, Sushil Jajodia and Steven Noel, <i>Time-Efficient and Cost-Effective Network Hardening Using Attack Graphs</i> , In Proceedings of the 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (June 2012) (“ <i>Albanese</i> ”)
Exhibit 1037	U.S. Patent No. 9,082,126 to Nielsen et al. (“ <i>Nielsen</i> ”)
Exhibit 1038	Luca Allodi and Fabio Massacci, <i>Comparing Vulnerability Severity and Exploits Using Case-Control Studies</i> , ACM Transactions on Information and System Security, Vol. 17, No. 1, Article 1 (Aug. 2014) (“ <i>Allodi</i> ”)
Exhibit 1039	U.S. Patent Application Publication No. 2011/0191854 to Giakouminakis et al. (“ <i>Giakouminakis</i> ”)
Exhibit 1040	Gabriele Gianini, Marco Cremonini, Andrea Rainini, G. Lena Cota, and L. Ghemmogne Fossi, <i>A Game Theoretic Approach to Vulnerability Patching</i> , In International Conference on Information and Communication Technology Research (ICTRC) 88-91 (2015) (“ <i>Gianini</i> ”)

Exhibit 1041	Stefan Frei, <i>Security Econometrics: The Dynamics of (In)Security</i> , Ph.D. dissertation, ETH Zurich (2009) available at https://www.research-collection.ethz.ch/handle/20.500.11850/151394 (“ <i>Frei</i> ”)
Exhibit 1042	Mohit Kumar, <i>New Dark Web Marketplace Offers Zero-Day Exploits to Hackers</i> , The Hacker News (2015) (“ <i>Kumar</i> ”)
Exhibit 1043	<i>Famous Dark Net Marketplaces to but Exploits – 0 day Vulnerabilities – Malwares for Research</i> , International Institute of Cyber Security (2015) (“ <i>International Institute of Cyber Security</i> ”)
Exhibit 1044	Peter Mell, Tiffany Bergeron and David Henning, <i>Creating a Patch and Vulnerability Management Program</i> , NIST Special Publication 800-40, Version 2.0 (Nov. 2005) (“ <i>Mell</i> ”)
Exhibit 1045	Curriculum vitae (CV) of Dr. Erez Zadok.

CERTIFICATION OF WORD COUNT

The undersigned certifies pursuant to 37 C.F.R. § 42.24 that the foregoing Petition for *Inter Partes* Review, excluding any table of contents, mandatory notices under 37 C.F.R. § 42.8, certificates of service or word count, or appendix of exhibits, contains 13,494 words according to the word-processing program used to prepare this document (Microsoft Word).

Dated: August 15, 2025

Respectfully submitted,

ERISE IP, P.A.

BY: /s/ Paul R. Hart

Paul R. Hart, Reg. No. 59,646

COUNSEL FOR PETITIONER

**CERTIFICATE OF SERVICE ON PATENT OWNER
UNDER 37 C.F.R. § 42.105(a)**

Pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(b), the undersigned certifies that on August 15, 2025, a complete and entire copy of this Petition for *Inter Partes* Review and Exhibits were provided via Federal Express to the Patent Owner by serving the correspondence address of record for the '385 Patent:

POLSINELLI PC
PO Box 140310
Kansas City, MO 64114-0310

Further, a courtesy copy of this Petition for *Inter Partes* Review was sent via email to Patent Owner's litigation counsel:

Jay P. Kesan (jay.kesan@kkd-law.com)
Cecil E. Key (cecil.key@kkd-law.com)
Alfonso Garcia Chan (achan@cahill.com)
Britton F. Davis (bdavis@cahill.com)
Christopher C. Campbell (ccampbell@cahill.com)
Halima Shukri Ndai (hndai@cahill.com)

Respectfully submitted,

ERISE IP, P.A.

BY: /s/ Paul R. Hart
Paul R. Hart, Reg. No. 59,646

COUNSEL FOR PETITIONER