

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

PALO ALTO NETWORKS, INC.,

Petitioner,

v.

QOMPLX LLC,

Patent Owner.

IPR2026-00431

U.S. Patent No. 12,301,627

**PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 12,301,627**

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION.....	1
II. MANDATORY NOTICES AND FEES	2
A. Real Party In Interest.....	2
B. Related Litigation.....	2
C. Counsel and Service Information.....	3
D. Payment of Fees	4
E. Requirements for <i>Inter Partes</i> Review	4
III. GROUNDS	4
IV. LEVEL OF SKILL IN THE ART, AND STATE OF THE ART	4
A. Person Of Ordinary Skill In The Art.....	4
B. State Of The Art	5
V. THE '627 PATENT	5
A. The '627 Patent Disclosures In Common With Crabtree	7
B. The Prosecution History.....	9
C. The Challenged Claims' Earliest Effective Filing Date Is January 30, 2020 10	
VI. CLAIM CONSTRUCTION	12
VII. GROUND 1: CLAIMS 9, 10, 15, and 19 ARE OBVIOUS OVER STOKES AND CRABTREE	12
A. Stokes	12
B. Crabtree	15
C. Combining Stokes And Crabtree	15
D. Claim 1	18
1. Element 1.1.....	19
2. Element 1.2.....	19
3. Element 1.3.....	20
4. Element 1.4.....	23
5. Element 1.5.....	23
6. Element 1.6.....	24

7. Element 1.7.....	26
8. Element 1.8.....	29
9. Element 1.9.....	32
10. Element 1.10.....	32
11. Element 1.11.....	36
12. Element 1.12.....	39
E. Claim 9.....	40
F. Claim 10.....	42
G. Claim 15.....	42
H. Claim 18.....	43
1. Elements 18.1-18.5, 18.7, and 18.9.....	43
2. Element 18.6.....	44
3. Element 18.8.....	46
4. Element 18.10.....	47
5. Element 18.11.....	48
I. Claim 19.....	52
VIII. GROUND 2: CLAIMS 20 AND 21 ARE OBVIOUS OVER STOKES IN VIEW OF CRABTREE AND SEKAR	52
A. Sekar (EX1006).....	53
B. Combining Stokes, Crabtree, And Sekar	53
C. Claim 20.....	55
1. Element 20.1.....	55
2. Element 20.2.....	56
3. Element 20.3.....	58
D. Claim 21	59
1. Element 21.1.....	59
2. Element 21.2.....	59
3. Element 21.3.....	61
4. Element 21.4.....	62
IX. NO OBJECTIVE INDICIA OF NON-OBVIOUSNESS.....	63
X. CONCLUSION.....	63

TABLE OF CONTENTS

Page

EXHIBIT LIST

Ex.	Description
1001	U.S. Patent No. 12,301,627 (“’627 patent”)
1002	File History of U.S. Patent No. 12,301,627
1003	Declaration of Paul S. Min, dated July 20, 2026 (“Min Decl.” or “Min”)
1004	U.S. Patent App. Pub. No. 2018/0367548 (“Stokes”)
1005	U.S. Patent App. Pub. No. 2018/0219919 (“Crabtree”)
1006	U.S. Patent App. Pub. No. 2020/0059481 (“Sekar”)
1007	U.S. Patent App. Pub. No. 2020/0296137 A1 (“publication of the ’270 application”)
1008	U.S. Patent App. Pub. No. 2022/0014560 A1 (“publication of the ’346 application”)

1009	Leman Akoglu, et al., Graph Based Anomaly Detection and Description: a Survey, Data Mining and Knowledge Discovery, Vol. 29, Issue 3 (Springer 2014)
1010	Steven Noel & Sushil Jajodia, Advanced Vulnerability Analysis and Intrusion Detection Through Predictive Attack Graphs, Center for Secure Info. Sys., George Mason Univ. (2009) available at https://csis.gmu.edu/noel/pubs/2009_c4i.pdf
1011	Managing Cyber Threats (Vipin Kumar, et al. eds, Springer Science+Business Media, Inc., 2005)
1012	William Eberle & Lawrence Holder, Incremental Anomaly Detection in Graphs, 2013 IEEE 13th Int'l Conference on Data Mining Workshops (2013)
1013	Plaintiff Qomplx's Preliminary Infringement Contentions, Qomplx LLC v. Microsoft Corp., Case No. 1:25-cv-01383-ADA (W.D. Tex.) (Nov. 24, 2025)

1014	Plaintiff Qomplx LLC's Preliminary Disclosure of Proposed Claim Constructions, Qomplx LLC v. Microsoft Corp., Case No. 1:25-cv-01383-ADA (W.D. Tex.) (Mar. 2, 2026)
1015	Defendant's Preliminary Claim Constructions, Qomplx LLC v. Microsoft Corp., Case No. 1:25-cv-01383-ADA (W.D. Tex.) (Mar. 2, 2026)
1016	Claims Listing of Challenged '627 Patent Claims
1017	U.S. Patent No. 9,225,730
1018	Microsoft Computer Dictionary, 5th Ed. (2002)
1019	Redline Comparison of Crabtree to '627 Patent Specification
1020	Complaint For Patent Infringement, Qomplx LLC v. Microsoft Corp., Case No. 1:25-cv-01383-ADA (W.D. Tex. Aug. 28, 2025)
1021	U.S. Patent App. Pub. No. 2019/0081876A1
1022	Seth Webster, et al., Experience Using Active and Passive Mapping for Network Situational Awareness, IEEE 5th Int'l Symposium on Network Computing and Applications (2006)

1023	Genevieve Bartlett, et al., Understanding Passive and Active Service Discovery, IMC '07 (2007)
1024	Declaration of Dr. John R. Black, Jr., Exhibit 1003 from Microsoft Corp. v. Qomplx LLC, IPR2026-00325
1025	Curriculum Vitae of Dr. Paul S. Min

LIST OF CHALLENGED CLAIMS

Claim	Limitation
[1.1]	A computer system comprising:
[1.2]	a hardware memory, wherein the computer system is configured to execute software instructions stored on nontransitory machine-readable storage media comprising software instructions that:
[1.3]	store in the hardware memory a representation of a first graph, wherein the representation of the first graph comprises representations of a first plurality of nodes corresponding to a first plurality of entities and further comprises representations of a first plurality of edges, wherein the first graph is a directed graph,
[1.4]	wherein the first plurality of entities comprises a plurality of accounts and a plurality of resources, and
[1.5]	wherein each edge of the first plurality of edges corresponds to a respective relationship between a respective pair of entities;

Claim	Limitation
[1.6]	receive streaming data comprising time-stamped data about events relating to one or more entities of the first plurality of entities,
[1.7]	based on a first portion of the streaming data, identify a first entity that does not correspond to any of the first plurality of nodes, wherein the first entity is not of the first plurality of entities,
[1.8]	based on a second portion of the streaming data, wherein the second portion is not identical to the first portion, identify a first relationship between a pair of entities of the first plurality of entities that does not correspond to any of the first plurality of edges,
[1.9]	modify, in the hardware memory, the representation of the first graph to generate a modified representation of the first graph, wherein the modified representation of the first graph comprises a representation of a first node corresponding to the first entity and a representation of a first edge corresponding to the first

Claim	Limitation
	relationship, wherein the first node is not of the first plurality of nodes and the first edge is not of the first plurality of edges,
[1.10]	identify, based on the modified representation of the first graph, an attack path that could be involved in an attack involving the first entity,
[1.11]	wherein identifying the attack path comprises: identifying a second entity that can be reached using the first entity, wherein the second entity corresponds to a second node, and the second node is related by one or more edges to the first node corresponding to the first entity in the modified representation of the first graph; and, identifying a third entity that can be reached using the second entity, wherein the third entity corresponds to a third node, and the third node is related by one or more edges to the second node in the modified representation of the first graph; and
[1.12]	generate a report comprising an identification of the first entity and at least one of the second entity and the third entity.

Claim	Limitation
[9]	The computer system of claim 1, wherein the first graph is a cyber-physical graph.
[10]	The computer system of claim 9, wherein the report is an alert.
[15]	The computer system of claim 1, wherein the computer system comprises a plurality of virtual computing machines.
[18.1]	A computer system comprising:
[18.2]	a hardware memory, wherein the computer system is configured to execute software instructions stored on nontransitory machine readable storage media comprising software instructions that:
[18.3]	store in the hardware memory a representation of a first graph, wherein the representation of the first graph comprises

Claim	Limitation
	representations of a first plurality of nodes corresponding to a first plurality of entities and further comprises representations of a first plurality of edges, wherein the first graph is a directed graph,
[18.4]	wherein the first plurality of entities comprises a plurality of accounts and a plurality of resources, and
[18.5]	wherein each edge of the first plurality of edges corresponds to a respective relationship between a respective pair of entities;
[18.6]	identify, based on the representation of the first graph, a first plurality of attack paths comprising a first entity of the first plurality of entities, wherein each attack path of the first plurality of attack paths targets a second entity of the first plurality of entities,
[18.7]	receive streaming data comprising time-stamped data about events relating to one or more entities of the first plurality of entities,

Claim	Limitation
[18.8]	based on a first portion of the streaming data, identify a third entity that does not correspond to any of the first plurality of nodes, wherein the third entity is not of the first plurality of entities,
[18.9]	based on a second portion of the streaming data, wherein the second portion is not identical to the first portion, identify a first relationship between a pair of entities of the first plurality of entities that does not correspond to any of the first plurality of edges,
[18.10]	modify, in the hardware memory, the representation of the first graph to generate a modified representation of the first graph, wherein the modified representation of the first graph comprises a representation of a node corresponding to the third entity and an edge corresponding to the first relationship, wherein the node is not of the first plurality of nodes and the edge is not of the first plurality of edges, and

Claim	Limitation
[18.11]	identify, based on the modified representation of the first graph, a second plurality of attack paths comprising the first entity, wherein each attack path of the second plurality of attack paths targets the second entity, and wherein an attack path of the second plurality of attack paths comprises the third entity.
[19]	The computer system of claim 18, wherein the first graph is a cyber-physical graph.
[20.1]	The computer system of claim 18, wherein identifying the second plurality of attack paths comprises:
[20.2]	identifying a first plurality of event flows that could involve the first entity, and
[20.3]	identifying a first plurality of dependencies associated with the first plurality of event flows.

Claim	Limitation
[21.1]	The computer system of claim 20, further configured to execute software instructions stored on nontransitory machine-readable storage media comprising software instructions that:
[21.2]	based on a third portion of the streaming data, identify an anomalous event,
[21.3]	store in the hardware memory a dependency tree for the anomalous event, and
[21.4]	based on the dependency tree, determine a point of origin for the anomalous event.

I. INTRODUCTION

Palo Alto Networks, Inc. (“Petitioner”) respectfully requests *inter partes* review (“IPR”) of claims 9, 10, 15, 19, 20, and 21 of U.S. Patent 12,301,627 (“’627 patent,” EX1001), allegedly assigned to Qomplx LLC (“Patent Owner”). As explained below, these claims should be found unpatentable and cancelled.

The challenged claims relate to a computer system storing and updating a graph structure comprising nodes and edges that represent, respectively, entities in a computer network and their relationships. The computer system uses this graph to identify attack paths and generates a report identifying entities implicated by those attack paths.

Years before the challenged claims’ effective filing date, Stokes (EX1004) disclosed the same concept of using streaming event data to perform graph-based cybersecurity analysis, including identifying suspicious paths through a graph in which nodes represent entities and edges represent relationships between entities. EX1004, [0037]-[0040], [0047].

Stokes was not previously considered by the Office and discloses certain features relied on during prosecution to allegedly distinguish the challenged claims from the prior art. The combination of Stokes and Crabtree (EX1005), which includes many disclosures identical to the ’627 patent, renders challenged claims 9,

10, 15, and 19 obvious. Additionally, the combination of Stokes-Crabtree and Sekar (EX1006) renders challenged claims 20 and 21 obvious.

II. MANDATORY NOTICES AND FEES

A. Real Party In Interest

The real party in interest is Palo Alto Networks, Inc.

B. Related Litigation

Patent Owner is currently asserting the '627 patent against Petitioner in *Qomplx LLC v. Palo Alto Networks, Inc.*, Case No. 2-25-cv-00913 (E.D. Tex.), filed August 28, 2025. Patent Owner has also asserted the '627 patent against Microsoft in *Qomplx LLC v. Microsoft Corporation*, Case No. 1-25-cv-01383 (W.D. Tex.), filed August 28, 2025.

Microsoft has filed an IPR petition against the '627 patent in IPR2026-00325. Microsoft's petition challenges claims 1-3, 5-6, 11-14, 16-18, 22, 24, and 27-28. Petitioner Palo Alto Networks filed a "copycat" petition, IPR2026-00364, and concurrently filed a motion to join Microsoft's IPR proceedings in the event trial is instituted. This petition challenges claims 9, 10, 15, 19, 20, and 21 of the '627 patent. These claims were not included in the prior petitions against the '627 patent by Microsoft and Petitioner Palo Alto Networks.

C. Counsel and Service Information

Petitioner provides the following counsel and service information. Petitioner consents to electronic service at the email addresses listed in the table below.

Pursuant to 37 C.F.R. § 42.10(b), Powers of Attorney accompany this Petition.

<u>Lead Counsel</u>	<u>Back-Up Counsel</u>
James M. Glass (Reg. No. 46,729) jimglass@quinnemanuel.com QUINN EMANUEL URQUHART & SULLIVAN, LLP 295 5 th Avenue, 9 th Floor New York, NY 10016 Tel: (212) 849-7000 Fax: (212) 849-7100	Quincy Lu (Reg. No. 76,954) quincylu@quinnemanuel.com QUINN EMANUEL URQUHART & SULLIVAN, LLP 1109 First Avenue, Suite 210 Seattle, WA 98101 Tel: (206) 905-7000 Fax: (206) 905-7100 Brian Biddinger (<i>pro hac vice</i> forthcoming) brianbiddinger@quinnemanuel.com QUINN EMANUEL URQUHART & SULLIVAN, LLP 295 5 th Avenue, 9 th Floor New York, NY 10016 Tel: (212) 849-7000 Fax: (212) 849-7100 Jeremy Thomas (<i>pro hac vice</i> forthcoming) jeremythomas@quinnemanuel.com QUINN EMANUEL URQUHART & SULLIVAN, LLP 50 California Street, 22 nd Floor San Francisco, California 94111 Tel: (415) 875-6600 Fax: (415) 875-6700

D. Payment of Fees

The undersigned authorizes the Office to charge the fee required for this Petition (and any additional fees) to Deposit Account No. 50-5708.

E. Requirements for *Inter Partes* Review

Petitioner certifies the '627 patent is available for IPR, and Petitioner is not barred or estopped from requesting this proceeding.

III. GROUNDS

Petitioner requests *inter partes* review of Claims 9, 10, 15, 19, 20, and 21 of the '627 patent, on the following statutory grounds:

Ground	Basis	Reference(s)	Claims
1	§103	Stokes, Crabtree	9, 10, 15, 19
2	§103	Stokes, Crabtree, Sekar	20, 21

These are the same grounds raised in Microsoft's IPR2026-00325 and Petitioner Palo Alto Networks' IPR2026-00364.

IV. LEVEL OF SKILL IN THE ART, AND STATE OF THE ART

A. Person Of Ordinary Skill In The Art

The person of ordinary skill in the art for purposes of the '627 patent in January 2020 ("POSITA") would have had a bachelor's degree in computer science, or equivalent, with two to four years' experience designing, implementing, or otherwise working with cybersecurity technologies. More education, e.g., a master's degree or Ph.D., would compensate for less work experience and vice versa.

EX1003, ¶36. For purposes of this Petition, the knowledge and skill of a POSITA would not change if the relevant priority date were April 22, 2021. EX1003, ¶36. The state of the art, immediately below, identifies certain technologies within the knowledge and skill of a POSITA. *Id.*

B. State Of The Art

Graph structures have long been used to represent and analyze data, including entities and their behaviors. *See* EX1012, 522. Such techniques were applied to computer networks more than a decade before the '627 patent's filing date. *Id.*, 521. EX1003, ¶37. These types of graphical network representations have been commonly used to identify anomalies in a computer network, such as by identifying deviations from normative behavior patterns. *See id.* ("In this work, we describe methods for graph-based anomaly detection"). *See also* EX1009, 673-74 (2014 survey of graph based methods for identifying "[a]nomalies in computer networks"). EX1003, ¶38. Graph-based techniques have been used to predict, represent, and analyze attack paths in computer networks since more than a decade before the '627 patent's filing date. *See* EX1010, 1 (2009 publication explaining that "[a]ttack graphs map paths of vulnerability, showing how attackers can incrementally penetrate a network,""). *See also* EX1011, 247-48. EX1003, ¶39.

V. THE '627 PATENT

The '627 patent issued May 13, 2025, from an application filed September 20, 2024, which is a continuation of Application No. 17/237,346 (the "'346 application"), filed April 22, 2021. The '346 application is a continuation-in-part of Application No. 16/777,270 (the "'270 application"), filed January 30, 2020. EX1001, Cover, 1:9-18. The '627 patent alleges priority through a series of continuation and continuation-in-part applications to an application filed October 28, 2015. However, as discussed below, the challenged claims of the '627 patent are not entitled to an effective filing date any earlier than January 30, 2020. EX1003, ¶40.

The '627 patent relates to cybersecurity and, within a computer network, identifying "attack paths and points of origin." EX1001, 2:45-48; *id.*, 2:52-57. It uses network graphs to do so, explaining that "[p]ossible attack paths may be analyzed using the cyber-physical graph." *Id.*, 28:46-50. It describes a "cyber-physical graph" ("CPG") that represents network devices "as nodes" and "the physical or logical connections between them as edges between the nodes." *Id.*, 28:15-18. Figure 22 shows an example CPG, with network entities as nodes (circles), the relationships between entities as edges (arrowed lines), and three "exemplary attack paths ranked by likelihood":

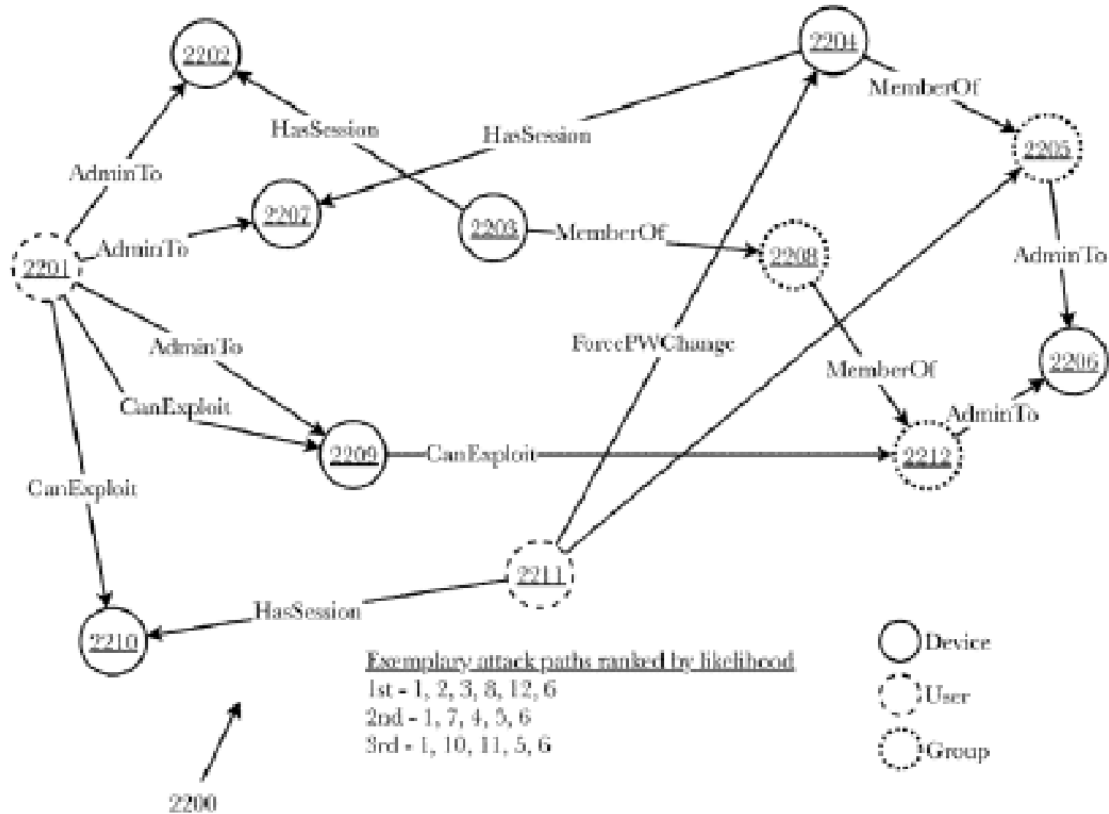


Fig. 22

EX1001, Fig. 22. The '627 patent describes “running graph analysis algorithms” to analyze and rank possible attack paths. *Id.*, 28:46-50. EX1003, ¶¶41-45.

A. The '627 Patent Disclosures In Common With Crabtree

The '627 patent and Crabtree share numerous disclosures. *See* EX1019 (redline comparison showing differences); EX1003, ¶¶55-56. Crabtree lists as inventors two of the '627 patent's three inventors and Crabtree issued from Application No. 15/823,363, which is an ancestor to the '627 patent through several continuation-in-part applications (EX1001, 1:9-26).

The '627 patent and Crabtree both describe “a cyber-physical system graph (CPG)” that provides “a visualization of hierarchies and relationships between devices and resources in a security infrastructure, contextualizing security information with physical device relationships that are easily understandable for security personnel and users.” EX1001, 23:55-62; EX1005, [0076]. Both also describe such a “graph” as a “representation of information and relationships, where each primary unit of information makes up a ‘node’ or ‘vertex’ of the graph and the relationship between two nodes makes up an edge of the graph.” EX1001, 7:45-49; EX1005, [0044].

The '627 patent and Crabtree disclose receiving time-series information and using it “to produce [] a graph visualization of users, servers, devices, and other resources correlating physical relationships ... with logical relationships (such as access privileges or database connections), to produce a meaningful and contextualized visualization of a security infrastructure that reflects the current state of the internal relationships present in the infrastructure.” EX1001, 23:34-24:13; EX1005, [0076]. Both also describe “updating CPG information as changes occur and events are logged.” EX1001, 24:2-5; EX1005, [0076]. Time-series data “may be received continuously ... for use in producing time-series graphs [] of states and changes over time”, thereby “providing live monitoring” of the state of the network. EX1001, 23:40-52; EX1005, [0075].

The '627 patent and Crabtree also disclose the graph including information used to evaluate and identify security risks in the network, such as “behavior analytics information” for recognizing deviations from established patterns of network behavior. EX1001, 22:43-23:14, 23:62-65; EX1005, [0072], [0076]. The graph may also contain “impact assessment score[s]” reflecting access capabilities of user accounts in the network and “representing the potential risk should that account be compromised.” EX1001, 23:14-27, 23:65-24:2; EX1005, [0074], [0076].

B. The Prosecution History

Following a first Office action (EX1002, 188-197/290) the Applicant substantially amended claim 1 (which the challenged claims 9, 10, and 15 depend upon), among others, and added several new claims including independent claim 27 (issued as claim 18, which the challenged claims 19, 20, and 21 depend upon) (*Id.*, 220-239/290). The Applicant argued that the cited art failed to teach or disclose four newly-added/amended limitations of claim 1: [1] receiving “streaming data comprising time-stamped data about events”; [2] “based on a first portion of the streaming data, identifying a first entity that does not correspond to any node of the first plurality of nodes”; [3] generating a modified representation of the first graph including “a representation of a first node corresponding to the first entity”; and [4] using the modified graph representation to identify “an attack path involving the first entity,” including “identifying a second entity that can potentially be accessed using

the first entity.” *Id.*, 235-238/290. These limitations correspond to as-issued elements [1.6], [1.7], [1.9], [1.10], and [1.11] of the as-issued patent. EX1016, 2-3.¹ The Applicant argued that pending claim 27 (issued as independent claim 18) was “allowable for similar reasons as claim 1.” *Id.*, 238/290. Thereafter, the claims were allowed, subject to an Examiner’s amendment (*id.*, 267-281/290), and the ’627 patent issued May 13, 2025 (EX1001, Cover).

C. The Challenged Claims’ Earliest Effective Filing Date Is January 30, 2020

In its Complaint and preliminary infringement contentions in district court, Patent Owner has identified the challenged claims’ priority date as January 30, 2020, or later. EX1013, 9; EX1020, ¶124.

Patent Owner cannot show entitlement to an effective filing date for any Challenged Claim before January 30, 2020, because the earlier applications lack written description support for the Challenged Claims, including at least elements [1.10], [1.11], [18.6], and [18.11]. As explained above, the Applicant relied in part on elements [1.10] and [1.11] in arguing for allowance over the prior art in prosecution (*supra* Section V.B), and elements [18.6] and [18.11] recite similar

¹ Limitations [1.10] and [1.11] were slightly amended via Examiner’s amendment at allowance. *Id.*, 276/290.

subject matter concerning identifying attack paths in a graph (EX1016, 1-3). When amending the claims to add these elements during prosecution, Applicant identified alleged specification support in certain paragraphs and figures of the as-filed '283 application. EX1002, 231-33/290. As explained below, corresponding disclosures to this alleged support were first included in either the '270 application (filed January 30, 2020) or '346 application (filed April 22, 2021) and not found in any earlier application identified by the '627 patent.

For each of limitations [1.10], [1.11], and [18.11], Applicant cited as support ¶¶ [85], [86], and [103], and Figure 31 of the as-filed specification. *Id.*, 231/290, 233/290. For limitation [18.6], Applicant cited as support only ¶ [103]. *Id.*, 232/290. Disclosures corresponding to ¶ [103] first appeared in the '270 application and are found in no earlier applications. *Compare id.*, 113-114/290 (¶ [103]) with EX1007, [0100] (publication of the '270 application). Disclosures corresponding to ¶¶ [85], [86], and Figure 31 first appeared in the '346 application. *Compare* EX1002, 60/290 (FIG. 31), 104-105/290 (¶¶ [85]-[86]), with EX1008, FIG. 31, [0101]-[0102] (publication of the '346 application). No earlier application supports these claim limitations.

To the extent Patent Owner contends otherwise, it “must demonstrate entitlement to a priority date” that would render Crabtree not prior art. *See Penumbra, Inc., v. Rapidpulses, Inc.*, IPR2021-01466, Paper 34 at 18 (PTAB Mar.

10, 2023) (Precedential as to § II.E.3) (“[T]he patentee must demonstrate that the claims meet the requirements of [Section 120, including Section 112].” (quoting *In re NTP, Inc.*, 654 F.3d 1268, 1276–77 (Fed. Cir. 2011)); *see also Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1379-80 (Fed. Cir. 2015) (discussing burdens in *inter partes* review to show entitlement to earlier filing dates).

VI. CLAIM CONSTRUCTION

Unless otherwise expressly discussed below, Petitioner applies the plain and ordinary meaning of all claim terms. EX1003, ¶46.

VII. GROUND 1: CLAIMS 9, 10, 15, AND 19 ARE OBVIOUS OVER STOKES AND CRABTREE

As explained below, claims 9, 10, 15, and 19 are obvious over Stokes and Crabtree. EX1003, ¶47.

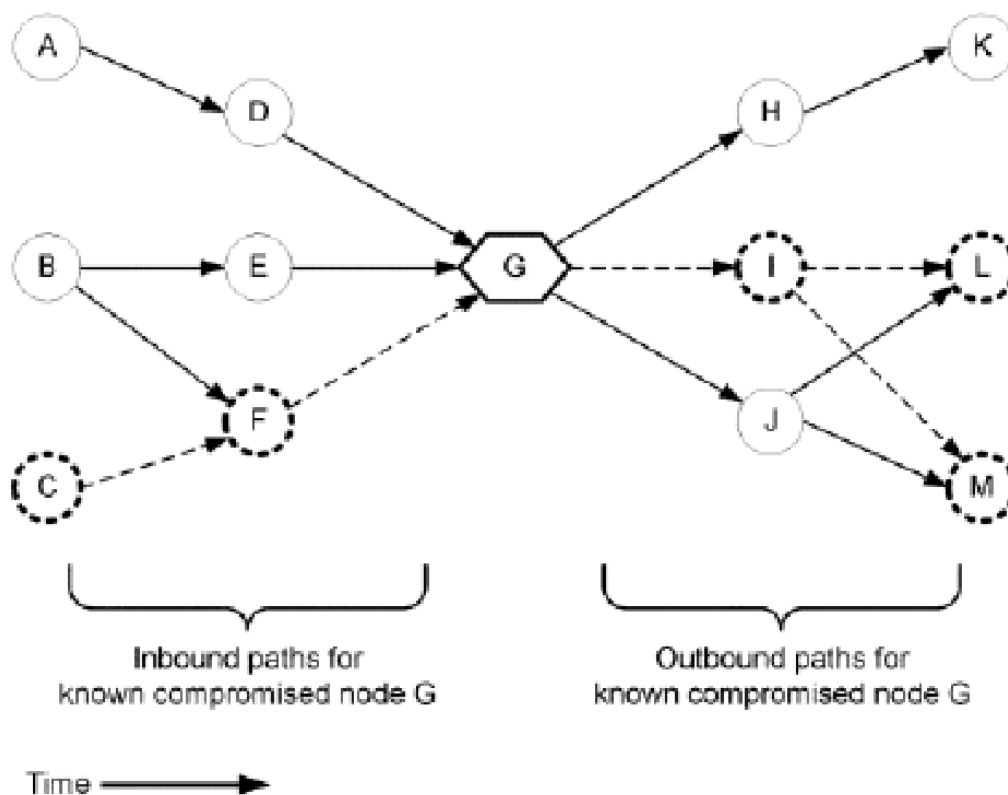
A. Stokes

Stokes, a U.S. patent application, published December 20, 2018, and is prior art under at least 35 U.S.C. § 102(a)(1). EX1004. Stokes was not cited during prosecution and is not referenced by the ’627 patent. EX1003, ¶48.

Stokes describes using a graph to detect malicious activity in computer networks. It does so, for example, by identifying potentially malicious (or “rare”) paths within a large graph of network connections. EX1004, [0003], [0005], [0006], [0044]. Stokes frames these teachings in two scenarios: (i) *forensic analysis*, which

starts from at least one known compromised computer or user account and identifies inbound/outbound paths; and (ii) *general detection*, which seeks previously unknown compromised nodes. *Id.*, [0005], [0028], [0029], [0048]. Stokes further explains that events may be processed “in real-time” and that the system “may be implemented ... in a streaming environment.” *Id.*, [0023], [0038]. EX1003, ¶49.

Stokes constructs a network graph with computers and user accounts modeled as nodes, and computer connections and user logon events modeled as directed edges. EX1004, [0024], [0025], [0037]-[0038]. Stokes Figure 3 shows an example graph “for describing forensic analysis with respect to tracking malicious lateral movement across a computer network” (*id.*, [0063]):



300

FIG. 3

Id., FIG. 3. In Figure 3, “a known compromised node (denoted as nod G [sic]) is discovered” and the system identifies “the unknown compromised computers and user accounts (dashed nodes C, F, I, L, and M) along the malicious access path (dashed lines).” *Id.*, [0066]. EX1003, ¶¶50-51.

The Stokes system uses path-rate scores for each path to identify rare, (i.e., anomalous) paths that are “more likely to correspond to malicious lateral movement.” *Id.*, [0047]; *see also id.*, [0044], [0058].

B. Crabtree

Crabtree (EX1005) published August 2, 2018, from an application filed November 27, 2017 (EX1005, 1), and is prior art under at least 35 U.S.C. § 102(a)(1) because Patent Owner cannot meet its burden in showing an effective filing date any earlier than January 30, 2020. *Supra* Section V.C. EX1003, ¶¶54-55.

Crabtree shares many of the same disclosures as the '627 patent. *Supra* Section V.A. EX1003, ¶¶55-58.

C. Combining Stokes And Crabtree

A POSITA would have been motivated to incorporate Crabtree's graph-updating and reporting teachings into Stokes's graph-based attack-path analysis system, and would have had a reasonable expectation of success in doing so. EX1003, ¶¶59, 66.

Stokes and Crabtree address closely related network-security problems using similar graph-based analysis and complementary teachings. Each teaches processing streaming network event data to generate and update graphs in which nodes represent entities (e.g., user accounts and computers) and edges represent relationships, and each uses those graphs to analyze network security. *See* EX1004,

[0020], [0024]-[0025], [0028], [0037]-[0040], [0047]-[0048], [0064]-[0071]; EX1005, [0063]-[0064], [0072], [0076], [0080]-[0081]. Crabtree’s more detailed teachings on graph updating and reporting (EX1005, [0063]-[0064], [0076], [0080]-[0081]), therefore complement Stokes’s attack-path identification techniques. EX1003, ¶60. A POSITA would have recognized those teachings as directly applicable to, and straightforward to implement in, the highly-similar Stokes system. *Id.*, ¶63.

A POSITA implementing Stokes would have understood the importance—if not imperative—of analyzing security issues based on the most current state of the network. Thus, a POSITA would have been motivated to implement Stokes’s streaming-environment teachings by continually updating the graph to add newly detected nodes and edges based on incoming event data. EX1003, ¶61. A POSITA would have understood that Stokes’s attack-path identification benefits from a complete and current graph because cybersecurity applications typically required rapid identification of, and response to, anomalies, intrusions, and attacks. EX1021, [0003] (explaining, in background, that “for some security-related applications ... *the ability to identify [] anomalous data fairly quickly may be critical*”) (emphasis added, parenthetical omitted); EX1003, ¶61. For example, Stokes describes how network attacks exploit an initial vulnerability and then expand “to additional

computers” (EX1004, [0002]), which illustrates the importance of quickly detecting attacks. EX1003, ¶61; *see also* EX1004, [0019].

Crabtree expressly teaches updating graph information “as changes occur and events are logged” and using received time-series and event information to produce graph representations of users, servers, devices, and other resources and their relationships. EX1005, [0072], [0076]. It further confirms the benefit of continuous updating, explaining that updating “[w]hen a new connection is detected,” allows for identification of “potential risk” and can lead to “quick handling” of any risky connections. *Id.*, [0081]. Crabtree also reveals other benefits, explaining that such updates “continuously reflect the current state of the organization, reflecting any recent changes, newly-discovered or announced vulnerabilities, ...” *Id.*, [0062]. And it notes that rapid detection can beneficially “alert an enterprise of ongoing cyberattack early in the attack cycle, possibly thwarting it but at least mitigating the damage.” *Id.*, [0063]. Accordingly, a POSITA would have recognized that Stokes benefited from graph-updating techniques that provide a more current graph, thereby enabling quicker and more comprehensive identification of potential attacks. EX1003, ¶¶61-62.

A POSITA would have had a reasonable expectation of success in combining Stokes and Crabtree. The two references operate in the same technical field of graph-based cybersecurity analysis and rely on compatible inputs, structures, and

objectives. Stokes already uses event-derived graph relationships to identify suspicious paths (*e.g.*, EX1004, [0003], [0026], [0038]-[0039], [0063]-[0067]), and Crabtree likewise uses event and time-series information to update and analyze a graph of users, devices, servers, and other resources (*e.g.* EX1005, [0044], [0075]-[0076]). Because Crabtree's graph-updating and reporting techniques would have been applied to the same type of graph-based cybersecurity system disclosed in Stokes, a POSITA would have understood that incorporating Crabtree's teachings into Stokes would involve the predictable use of known techniques in a closely related system, without technological incompatibility or undue experimentation. A POSITA therefore would have reasonably expected that combining Crabtree's graph-updating and reporting concepts with Stokes's attack-path identification framework would yield a more current and informative graph-based cybersecurity system. EX1003, ¶65.

Additional aspects of the Stokes-Crabtree combination are discussed below, as pertinent for individual claim elements. EX1003, ¶67.

D. Claim 1

Although this petition does not challenge the validity of Claim 1, Challenged Claims 9, 10, and 15 are dependent of Claim 1. Each limitation of Claim 1 is disclosed and/or rendered obvious by Stokes in view of Crabtree.

1. Element 1.1²

[1.1] A computer system comprising:

Stokes discloses a “server” (*computer system*) within its “exemplary environment ... for tracking malicious lateral movement across a computer network.” *E.g.*, EX1004, [0021]. Stokes teaches implementing its server using a “computing device” such as that illustrated in FIG. 7. *Id.*, [0022]; *see also id.*, FIG. 7, [0082]; *see generally* [0079]-[0086] (describing “computing device environment”). EX1003, ¶68.

2. Element 1.2

[1.2] a hardware memory, wherein the computer system is configured to execute software instructions stored on nontransitory machine-readable storage media

Stokes teaches implementing its server using a computing device that includes “memory” (EX1004, [0082]) and “additional storage” (*id.*, [0083]). *See also id.*, [0022]; *id.*, FIG. 7 (memory 704, other storage 708, 710). EX1003, ¶¶69-70.

Stokes further teaches implementing its methods and devices using “program code (i.e., instructions) embodied in tangible media” to practice the disclosed subject matter when “executed by a machine.” *Id.*, [0087]; *see also id.*, [0030] (describing server modules), [0081] (describing computing devices’ execution of “computer-

² Reference numbers have been added throughout, in the format [claim#.element#].

executable instructions, such as program modules” to “perform particular tasks” and how “program modules ... may be located in both local and remote computer storage media.”), [0084]-[0085]; EX1003, ¶71.

3. Element 1.3

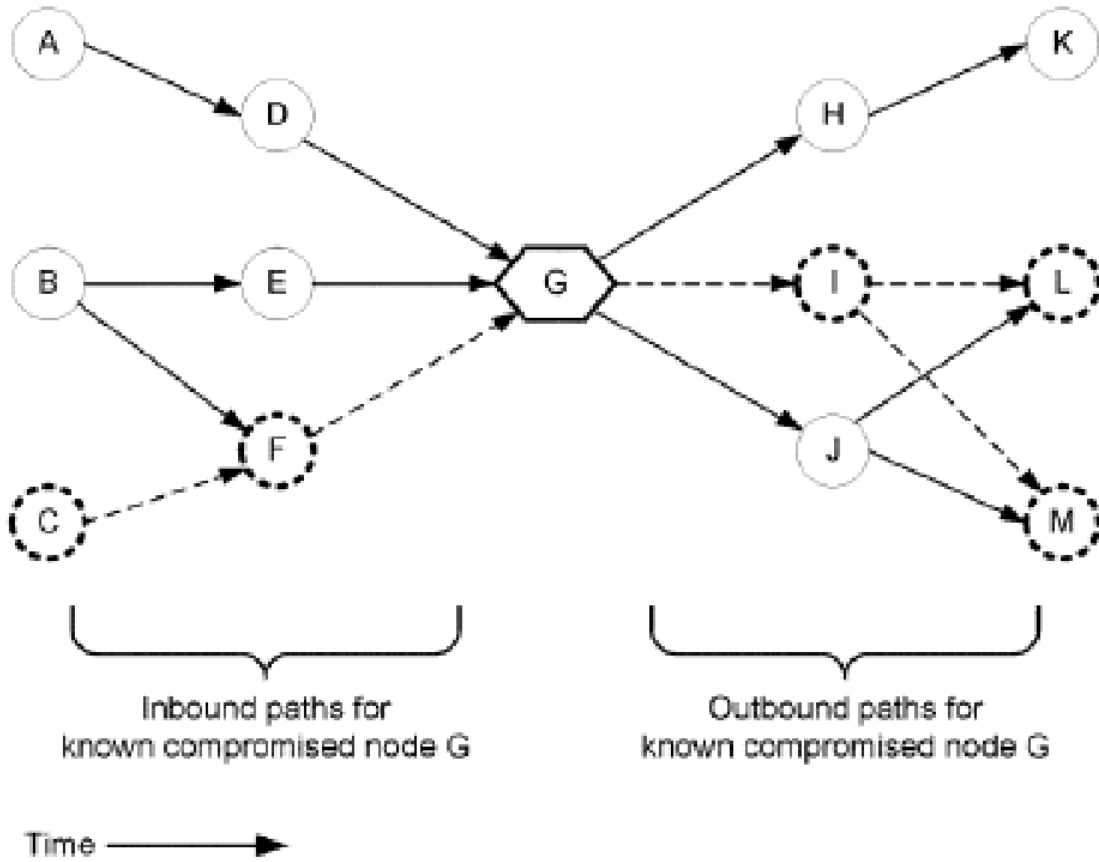
[1.3.a] store in the hardware memory a representation of a first graph,

The Stokes server includes “a network graph construction module 165” (EX1004, [0030], [0025]-[0027]) that “uses system and security events to construct a network connection graph 173” (*id.*, [0037]; FIG. 1). EX1003, ¶72. Stokes discloses the server using the network connection graph 173 for subsequent processing and operations. *See, e.g.*, EX1004, [0057]-[0062], FIG. 2, [0047], [0068]-[0071], FIG. 4, [0075]-[0078], FIG. 6; *see also id.*, [0030] (describing specific server modules that use or operate on the graph); [0026] (mathematical representation of graph). A POSITA would understand that the Stokes server *stores the data representing the graph in hardware memory*, both during its creation and during subsequent operations involving the graph. *See e.g., id.*, [0085] (describing memory storing “data structures ... or other data”), [0081], [0049]; EX1003, ¶¶73-75.

[1.3.b] wherein the representation of the first graph comprises representations of a first plurality of nodes corresponding to a first plurality of entities and further comprises representations of a first plurality of edges, wherein the first graph is a directed graph,

Stokes discloses “network connection graph 173” with plural “nodes” representing “computers and user accounts” (*first plurality of entities*) and plural

edges representing “computer-to-computer connections or user logon events.” *Id.*, [0025]; *see also id.*, [0024], [0026] (mathematical representation of graph, with V representing nodes and E representing edges), and [0063]-[0067] (further describing graphs); *infra* Section VII.D.4 (further describing plurality of entities). Stokes Figure 3 shows an example graph with plural nodes (A-M) and plural edges, shown as arrows between the nodes:



300

FIG. 3

Id., FIG. 3. EX1003, ¶76.

Stokes discloses using *directed graphs* with “order between the connection of a pair of nodes” and “directed edges.” *Id.*, [0038]; *see also id.*, [0039] (“*directed acyclic graphs (DAGs)*”) (emphasis added); *id.*, [0027] (explaining “directed edges”), [0063]; *see also* EX1001, 7:67-8:4 (’627 patent explaining directed graphs).

Stokes Figures 3 and 5, its two figures showing graphs, depict *directed graphs*, with arrows indicating the direction of each edge. EX1004, FIG. 3, FIG. 5, [0072]; EX1003, ¶77.

4. Element 1.4

[1.4] wherein the first plurality of entities comprises a plurality of accounts and a plurality of resources, and

In the Stokes graph, nodes represent “computers and user accounts” (*resources* and *accounts*). *E.g.*, EX1004, [0025]; *see also id.*, [0006]-[0007], claim 1, claim 12. A POSITA would understand the “computers” as resources at least because they perform computing functions on the network. EX1003, ¶78. The challenged ’627 patent itself characterizes computers and similar devices as “resources.” *E.g.*, EX1001, 21:13-14, 21:33-35, 24:2-14, 28:15-23; *see also id.*, 28:34-40 (graph with nodes representing “seven computers”); *id.* FIG. 22; *id.*, 21:30-35. EX1003, ¶¶79-80. In Stokes, the plural “computers and user accounts” make up a *first plurality of entities*. EX1004, [0025]. EX1003, ¶80.

5. Element 1.5

[1.5] wherein each edge of the first plurality of edges corresponds to a respective relationship between a respective pair of entities;

Stokes discloses that its graph “edges” model “computer-to-computer connections or user logon events.” EX1004, [0025]. Stokes also teaches using the known mathematical representation of a graph, with “E” representing “the set of

connection *relationships* ... between pairs of nodes, called edges.” EX1004, [0026]; *see also* [0037] (“These connections can be used to generate a graph 173 that shows the connections between the various nodes (e.g., the accounts and computers and their connections and *relationships*).”); *supra* Section VII.A. Of note, the ’627 patent mentions that edges can represent “the *physical or logical connections* between” nodes. EX1001, 28:15-19; *see also* Fig. 22 (showing nodes can have a “HasSession” relationship). Thus, in the Stokes graph, each of the plurality of edges corresponds to a relationship, such a connection relationship, between a respective pair of nodes. EX1003, ¶81.

6. Element 1.6

[1.6] receive streaming data comprising time-stamped data about events relating to one or more entities of the first plurality of entities,

Stokes-Crabtree satisfies this element, as each discloses real-time processing of streaming data for building or revising graphs. EX1003, ¶82, ¶88.

Stokes expressly teaches implementing its system “in a streaming environment.” *Id.*, [0038]. After disclosing that its system “processes event logs generated by a network of computers such as client,” Stokes explains that “events may be processed in real-time instead of being processed from event logs.” EX1004, [0023]. As the server uses these events to generate the graph (*e.g., id.*, [0024]-[0026], [0037]), Stokes teaches its server receiving and processing a real-time stream of event data. EX1003, ¶83; *see also* EX1018, 499.

Stokes discloses that this *streaming event data comprises time-stamped data about events relating to entities represented in a graph*. E.g., EX1004, [0037]-[0039]. For example, Stokes describes its server analyzing event data for computers and user accounts, such as Kerberos records, where “[e]ach Kerberos record represents an edge in the graph, with a source node, a destination node, and a timestamp indicating when this connection occurred.” EX1004, [0038]. These records thus comprise time-stamped data about *events* (e.g., connection events) *relating to one or more entities of the first plurality of entities* (computers and user accounts). *Id.*; *see also id.*, [0037]-[0038], [0045] (describing use of edge timestamps), [0023]; EX1003, ¶84.

Crabtree likewise discloses “converting streams” of event data/information “into graph representations of that data” (EX1005, [0050]), further confirming a POSITA would have implemented Stokes-Crabtree with a Stokes-like server that received and processed streaming time-stamped event data. EX1003, ¶¶85-87; *see also* EX1005 [0075]-[0076] (describing “live monitoring” and updating a graph “as changes occur and events are logged”); EX1002, 231/290, 106-108/290; EX1005, [0075]; EX1019, 33-34 (showing identical disclosures in Crabtree as ’627 patent). Crabtree further teaches associating timestamps with changes in a time-series graph. EX1005, [0061]. For example, Crabtree explains that “timestamps may be

associated with ongoing changes to reveal these updates over time.” *Id.* EX1003, ¶88.

7. Element 1.7

[1.7] based on a first portion of the streaming data, identify a first entity that does not correspond to any of the first plurality of nodes, wherein the first entity is not of the first plurality of entities,

In the streaming system taught by Stokes, the server receives and processes event data that arrives in portions, one after another, and uses this event data “to generate a graph.” EX1004, [0037]-[0038]; *supra* Section VII.D.6. For example, Stokes describes exemplary Kerberos records and how each “record” (portion of event data) “represents an edge in the graph, with a source node, a destination node, and a timestamp.” EX1004, [0038]. These events are “processed in real-time” (*id.*, [0023]) by the server’s network graph construction module, which uses them “in the generation of the network connection graph” (*id.*, [0037]). EX1003, ¶89. Stokes describes ongoing generation of events corresponding to Kerberos records, over a period of hours or days, in a streaming environment. *E.g.*, EX1004, [0038], [0051], [0023].³ Thus, Stokes teaches its server receiving streaming data made up of time-stamped portions (e.g., Kerberos records). EX1004, [0037]-[0038]; EX1003, ¶89.

³ Although the “hourly, daily” collection of “event logs” refers to Stokes’s offline implementation (*id.*, [0038]), it shows ongoing receipt of event data over time

Stokes further teaches using this event data “to generate a graph 173 that shows the connections between the various nodes (e.g., the accounts and computers and their connections and relationships).” EX1004, [0037]. Given these teachings, a POSITA would expect the obvious use case of event data for certain later events arriving after initial graph generation, with this later event data indicating a new account or computer recently added to the network. EX1003, ¶90. Stokes teaches—and thus renders obvious—a server configured to process this later portion of streaming event data, identify any newly added accounts or computers, and add corresponding nodes to the graph as part of its ongoing graph generation. *Id.*; EX1004, [0037]-[0038]; *see also id.*, [0051] (describing scenario when “[t]he graph can be implemented or extended”). As these nodes are “new,” they are not of the first plurality of nodes previously in the graph and, similarly, the corresponding computer or account is *a first entity that does not correspond to any of the first plurality of nodes* and *is not of the first plurality of entities*. EX1003, ¶90.

While Stokes teaches a server configured to satisfy this claim element, it does not expressly describe the obvious scenario of identifying entities not already represented in the graph. To the extent Stokes alone does not render this element

and Stokes teaches that these same “events may be processed in real-time instead of being processed from event logs.” *Id.*, [0023]. EX1003, ¶89.

obvious, Crabtree expressly teaches using incoming event data to update its graph to include nodes and edges corresponding to entities and relationships not yet represented in the existing graph. EX1005, [0075] (“As new events are recorded..., these events are automatically incorporated”); [0076] (describing “updating CPG [graph] information as changes occur”). Crabtree also describes identifying changes such as “new users” based on analysis of time-series data. *Id.*, [0078]; *see also id.*, [0079]-[0081]; EX1003, ¶91. These graph modifications involve identifying entities not already represented by an existing node in the graph. EX1003, ¶92.

In the combined system, Stokes provides the streaming graph environment and Crabtree provides the explicit teachings on identifying new entities for inclusion in the graph. *Supra* Section VII.C. In the combined system, a first incoming portion of event data reveals a previously unrepresented entity, and a later, non-identical portion reveals a previously unrepresented relationship involving already-graphed entities, after which the graph is modified accordingly. Notably, the alleged support identified by Applicant when adding this claim element during prosecution (application paragraphs [93], [94], [96]) is identical to Crabtree paragraphs [0078]-[0079] and [0081]. EX1002, 108-109/290, 231/290; EX1005, [0078]-[0079] and [0081]; EX1003, ¶93.

Thus, Stokes-Crabtree includes a server that uses a first portion of streaming data to identify an entity that is not of the first plurality of entities, such as a new

user account or other entity not previously represented in the graph. EX1003, ¶94; *see also supra* Section VII.C (explaining the combination).

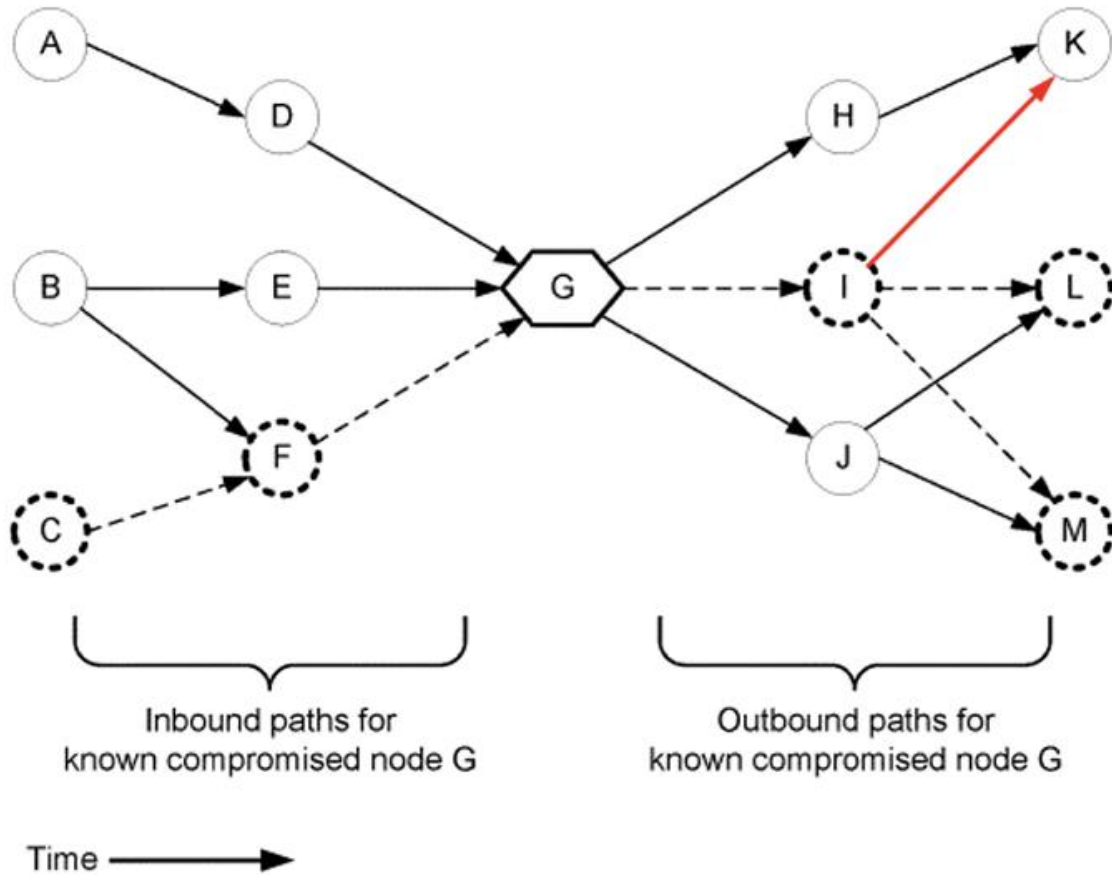
8. Element 1.8

[1.8] based on a second portion of the streaming data, wherein the second portion is not identical to the first portion, identify a first relationship between a pair of entities of the first plurality of entities that does not correspond to any of the first plurality of edges,

As explained above, Stokes teaches its server receiving streaming data made up of time-stamped portions (e.g., corresponding to Kerberos records). *Supra* Section VII.D.6. Because those portions are received at different times for different events, later portions (e.g., a second portion) are not identical to earlier portions (e.g., a first portion). EX1003, ¶95.

Stokes-Crabtree includes a server configured to update its graph based on streaming time-series data for events reflecting new entities or relationships. *Supra* Section VII.D.7. This includes updating the graph with “new connection information” “[w]hen a new connection is detected,” as taught by Crabtree. EX1005, [0081]; FIG. 16. Thus, the Stokes-Crabtree server identifies any new connections (i.e., new relationships) in incoming portions of event data, including new relationships between pairs of entities already represented by nodes in the graph. EX1003, ¶96. Crabtree describes its “new connection” generically, without limiting it to new devices (EX1005, [0081]; FIG. 16) and a POSITA would recognize the

obvious use case of a new relationship between entities already represented by nodes in the graph (i.e., a pair of the first plurality of entities), because a POSITA understood that pre-existing network entities would often have new connections/relationships with each other. EX1003, ¶¶96-97. The Stokes-Crabtree server is configured to identify such relationships for inclusion in its graph as new edges. EX1003, ¶¶96-97. Figure 3 of Stokes shows that each node in the network may connect with multiple other nodes, thus allowing for new edges to reflect new relationships, including between previously graphed nodes. The following modified version of Stokes's Figure 3 shows an obvious scenario of the graph of Stokes being modified to add a new edge (red arrow) added based on a new relationship identified between entities corresponding to nodes I and K.



EX1004, FIG. 3 (modified to add new edge in red), [0013]; EX1003, ¶96. Any newly-identified relationship does not correspond to any of the initially graphed (i.e., first) plurality of edges, as confirmed by the fact that it results in a new edge. EX1003, ¶96.

Finally, given that Stokes-Crabtree receives ongoing streaming event data, a POSITA would contemplate the server identifying a new entity based on one portion of event data (first portion, *supra* Section VII.D.7) and a new relationship based on a different portion of event data (second portion). *E.g.*, EX1005, [0081], FIG. 16; EX1003, ¶¶98-99.

9. Element 1.9

[1.9] modify, in the hardware memory, the representation of the first graph to generate a modified representation of the first graph, wherein the modified representation of the first graph comprises a representation of a first node corresponding to the first entity and a representation of a first edge corresponding to the first relationship, wherein the first node is not of the first plurality of nodes and the first edge is not of the first plurality of edges,

Stokes discloses storing its graph representations in hardware memory. *Supra* Section VII.D.3. Thus, in the Stokes-Crabtree system, any graph modifications would be reflected in hardware memory. EX1003, ¶¶100-102. The newly-identified entity and relationship(s) discussed above would result in a modified representation of the graph that included 1) a new node (*first node*) corresponding to the first entity (*supra* Section VII.D.7 (element 1.7)) and 2) a new edge (*first edge*) corresponding to the first relationship (*supra* Section VII.D.8 (element 1.8)). Because the first entity is a newly detected entity, the corresponding *first node* is not of the first plurality of nodes; and, because the first relationship is a newly detected relationship, the corresponding *first edge* is not of the first plurality of edges. EX1003, ¶¶103-104.

During prosecution, the alleged support identified for element 1.9 is identical to Crabtree paragraphs [0078]-[0079] and [0081]. EX1002, 108-109/290, 231/290; EX1005, [0078]-[0079], [0081]; EX1003, ¶105.

10. Element 1.10

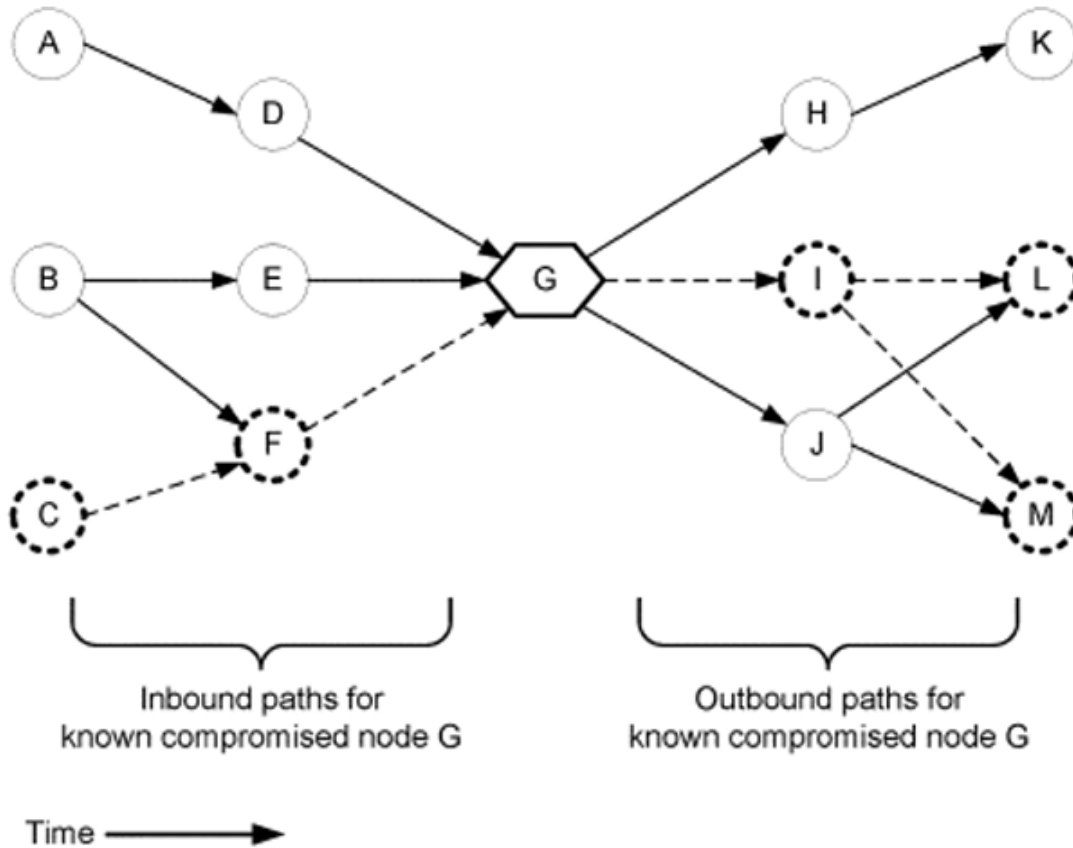
[1.10] identify, based on the modified representation of the first graph, an attack path that could be involved in an attack involving the first entity,

Stokes teaches using its graph to identify lateral movement paths associated with malicious nodes (*attack paths*). It explains that, “[w]ith forensic analysis, given a malicious node, **possible** lateral movement leading into or out of the node is identified,” and “other compromised computers” are identified. EX1004, [0028]; *see also id.*, [0038] (“[A] series of lateral movements can be represented as a path through the network connection graph.”). Stokes identifies lateral movements by first identifying a malicious node and then analyzing inbound and outbound paths from that node, whether actual or potential, *i.e.* a path that **could** be involved in an attack involving the first entity. *Id.*, [0068]-[0071], FIG. 4; *see also* [0020] (“Graph-based detection systems and techniques are provided to identify **potential** malicious lateral movement paths.”). Stokes in fact describes how to avoid false positives when identifying the potential attack paths. EX1004, [0044]-[0046]. An infected computer “may be the source of the initial infection, an intermediate node in the lateral movement path, or the final destination in the attack.” *Id.*, [0062]. EX1003, ¶106.

Stokes expressly associates the lateral movement paths with attacker activity, describing them “as a set of malicious paths corresponding to the attacker’s activity.” EX1004, [0003]; *see also id.*, [0002] (“[T]he attacker begins to use lateral movement techniques to explore the network.”). EX1003, ¶107.

The modified representation of the first graph includes the *first node* corresponding to the *first entity* (*supra* Section VII.D.9 (element 1.9)), and the server would carry out Stokes’s forensic analysis techniques to identify any malicious path, including those that involve the newly-identified first node. EX1004, [0038]-[0039], [0047], [0066]; EX1003, ¶108. This includes implementations in which the first entity was not represented in the initial graph because it had not previously appeared within the relevant graph time period, but is later added based on incoming event data. *E.g.*, *supra* Section VII.D.9; EX1004, [0038], [0045], [0048].

Stokes Figure 3 shows an exemplary attack path identification using a network connection graph with nodes A-M. In this example, nodes A-M include compromised (i.e., infected) node G among other nodes one or two hops away from node G. EX1004, FIG. 3, [0063]. Stokes describes performing inbound and outbound path analysis for node G to identify “compromised computers and user accounts (dashed nodes C, F, I, L, and M) *along the malicious access path (dashed lines).*” *Id.*, [0066]-[0071] (emphasis added). In the example shown in Figure 3, reproduced below, the identified malicious access paths (dashed lines) include a path that starts at node C, traverses nodes F, G, and I, and ends at node L, as well as a path that starts at node C, traverses nodes F, G, and I, and ends at node M.



300

FIG. 3

Id., FIG. 3. EX1003, ¶¶109-110.

The Stokes-Crabtree system would apply that same path-identification analysis to a modified graph, including paths involving any newly added nodes. For example, where the modified graph includes a newly added node corresponding to a first entity (*e.g.*, *supra* Sections VII.D.7-9) that is potentially malicious, Stokes's

forensic analysis would identify the dashed paths of Figure 3 as attack paths involving that first entity. EX1003, ¶110; *see also* EX1004, [0045], [0048]. In the combined Stokes-Crabtree system, with Crabtree's teachings on modifying the graph to add new nodes for new entities, identified attack paths would involve new entities not represented in the initial graph (additional examples of *first entities*). EX1003, ¶110.

Accordingly, Stokes-Crabtree teaches a server executing code to *identify, based on the modified representation of the first graph, an attack path that could be involved in an attack involving the first entity*. EX1003, ¶111.

11. Element 1.11

[1.11] wherein identifying the attack path comprises:

identifying a second entity that can be reached using the first entity, wherein the second entity corresponds to a second node, and the second node is related by one or more edges to the first node corresponding to the first entity in the modified representation of the first graph; and,

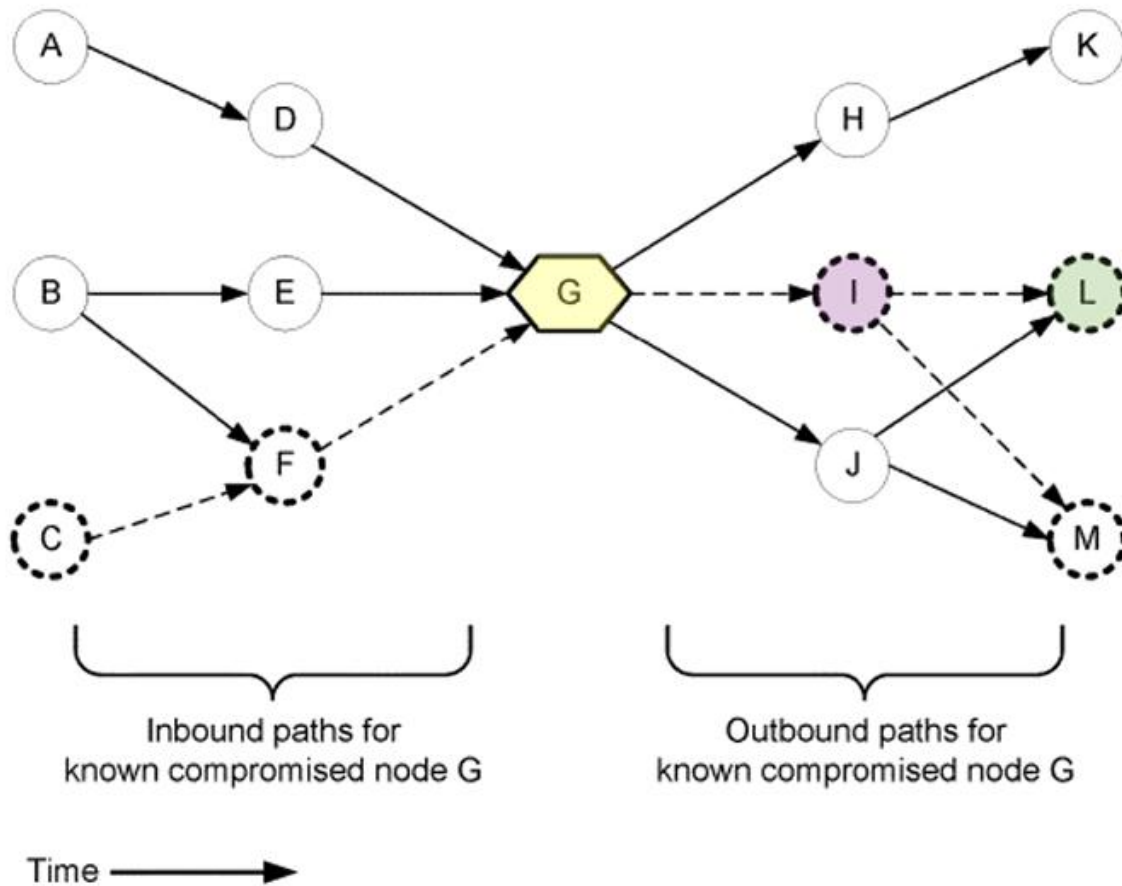
identifying a third entity that can be reached using the second entity, wherein the third entity corresponds to a third node, and the third node is related by one or more edges to the second node in the modified representation of the first graph; and

Stokes teaches that its malicious path identification (*supra* Section VII.D.10 (element 1.10)) involves identifying multiple entities that can be reached using a first entity. For example, Stokes teaches malicious paths with 2, 3, or more “hops,” from one node to the next. EX1004, [0041] (describing parameter “K, the number of hops in suspicious paths which can also be considered the desired subpath length,” and that K may be 2, 3, or more). EX1003, ¶113. Stokes further describes “possible

lateral movement leading into or out of the node,” including direct connections one hop away from a first node and indirect connections two hops away from a first node. *Id.*, [0005], [0028], [0048], [0064]-[0065]; EX1003, ¶114. For K=2, an identified malicious path would include both a second entity/node *related by one or more edges to the first entity/node*, and a third entity/node *related by one or more edges to the second entity/node*. EX1003, ¶¶113-115. And, as discussed above for element [1.10], Section VII.D.10 *supra*, Stokes’s identified malicious paths disclose the claimed “attack path.”

Stokes therefore teaches identifying attack paths that traverse from the first node to a second node and then to a third node, thereby identifying a second entity/node that can be reached using, and is related by an edge to, the first entity/node, and a third entity/node that can be reached using, and is related by an edge to, the second entity/node. EX1003, ¶¶112-115.

Stokes’s teaching of element [1.11] is further reflected in Figure 3, reproduced and annotated below.



EX1004, FIG. 3 (annotated); EX1003, ¶116. As explained for element 1.10, Section VII.D.10 *supra*, Stokes's Figure 3 teaches a first node (compromised node G, highlighted in yellow) and identifying the malicious access paths (i.e., attack paths, shown as dashed arrows) flowing through that node and the unknown compromised entities (computers and accounts) connected by those paths. As shown in Figure 3, above, this includes identifying a second node (node I, highlighted in purple) that corresponds to a second entity and is connected (i.e., related) by an edge to the first node (compromised node G), as well as identifying a third node (node L, highlighted

in green) that corresponds to a third entity and is connected by an edge to the second node. EX1003, ¶116.

12. Element 1.12

[1.12] generate a report comprising an identification of the first entity and at least one of the second entity and the third entity.

Stokes discloses a ranking module that is “configured to rank the plurality of paths ... and output results of the ranking,” EX1004, [0007]; *see also id.*, [0047] (“The results 190 are output”), FIG. 1 (depicting results 190 output from the server). Such output allows an “analyst” or “system administrator” to then investigate identified suspicious paths manually. *Id.*, [0048]. A POSITA would understand such outputting of results to a user to constitute a “report.” *See* EX1018, 450 (defining a “report” as “[t]he presentation of information about a given topic”); EX1003, ¶¶117-118. Because the ranked paths are sequences of edges connecting sequences of nodes, and because Stokes teaches that a user can investigate certain paths and focus on certain nodes, the output results would identify the entities included in each path (i.e., the first entity, second entity, and third entity included in the identified malicious path). *Id.*, [0027]. EX1003, ¶118.

Crabtree reinforces this understanding by teaching that cybersecurity analysis results may be presented to users in the form of reports identifying relevant entities and resources. For example, Crabtree discloses the ability to “format and deliver customized reports and dashboards.” EX1005, [0063]-[0064]. Crabtree further

discloses producing “a graph visualization of users, servers, devices, and other resources” and a “recommendation report.” *Id.*, [0076], [0080]-[0081]. Such reports therefore identify the relevant entities implicated by the underlying cybersecurity analysis. EX1003, ¶119. While Stokes already teaches one form of reports, a POSITA would have been motivated with a reasonable expectation of success to enhance the Stokes reporting with the formatted and customized reports taught by Crabtree. *Supra* Section VII.C. This would have involved minor changes to include the reporting modules taught by Crabtree. EX1003, ¶120. Indeed, the ’627 patent provides no more description than Crabtree on how to implement its reporting features, confirming a POSITA’s reasonable expectation of success in implementing those features in a Stokes-like system. *See* EX1019 (redline comparison showing differences between the ’627 patent and Crabtree); EX1003, ¶121.

E. Claim 9

[9] The computer system of claim 1, wherein the first graph is a cyber-physical graph.
--

The ’627 patent describes a “cyber-physical graph” as a graph “comprising nodes representing entities associated with the organization and edges representing relationships between entities associated with the organization.” EX1001, 3:16-21; EX1003, ¶122. One example of a “cyber-physical graph” is a graph that “represent[s] the relationships between devices, users, resources, and processes in

the organization, and contextualiz[es] cybersecurity information with physical and logical relationships that represent the flow of data and access to data within the organization including, in particular, network security protocols and procedures.” EX1001, 19:46-52.

As discussed above in elements 1.3 through 1.5, Stokes discloses a “network connection graph 173” with “nodes” representing “computers and user accounts,” which are organizational entities, and edges representing “computer-to-computer connections or user logon events,” which are relationships between these entities. EX1004, [0025]; EX1003, ¶123. This graph is therefore a ***cyber-physical graph*** because it “represent[s] entities associated with the organization and edges representing relationships between entities associated with the organization.” EX1001, 3:16-21. It is also a ***cyber-physical graph*** because it “represent[s] the relationships between devices, users, resources, and processes in the organization.” EX1001, 19:46-52. As explained above in element 1.6, the graph is built using streamed “real-time” data (EX1004, [0023]), including “Kerberos record[s]” of connection events (EX1004, [0038]), and therefore “represent[s] the flow of data and access to data within the organization.” EX1001, 19:46-52. As explained above in element 1.10, this graph is used to analyze “lateral movement[s]” into and out of a “malicious node” (EX1004, [0028]), and therefore “contextualiz[es] cybersecurity

information with physical and logical relationships.” EX1001, 19:46-52; EX1003, ¶123.

F. Claim 10

[10] The computer system of claim 9, wherein the report is an alert.

As discussed above in element 1.12, Crabtree discloses the ability to “finely customize reports and dashboards to specific audiences,” including “issuance of specialized alerts [to] assist in getting the correct mitigating actions done.” EX1005, [0064]; EX1003, ¶124.

As discussed above in Section VII.C and Element 1.12, it would have been obvious to combine Stokes and Crabtree’s reporting modules. EX1003, ¶125.

G. Claim 15

[15] The computer system of claim 1, wherein the computer system comprises a plurality of virtual computing machines.

Stokes describes that the “client devices” and “server” of the computer system can “be implemented using a variety of computing devices.” EX1004, [0022]. A POSITA would understand this would include virtual computing machines, which were known. EX1003, ¶126. Indeed, Crabtree discloses that its system can be “implemented in one or more virtualized computing environments (e.g., network computing clouds, virtual machines hosted on one or more physical computing machines, or other appropriate virtual environments).” EX1005, [0085]; EX1003, ¶126.

A POSITA would have been motivated to combine this teaching with Stokes because of the well-known benefits of using virtual computing machines. EX1003, ¶127. For example, virtual computing machines were known to be easier to administer than physical computing machines, since new computing resources could be spun up via software within a few minutes—whereas new physical machines could take days if not weeks to provision, and are more costly. *Id.* Moreover, in the context of securing the network, to which both Stokes and Crabtree are directed, virtual computing machines are advantageous as they can be isolated and shut down via software, which is useful and necessary if the computing resource is compromised. *Id.* Administrators can also recover earlier virtual machine snapshots to restore the machine to an uncompromised state. A POSITA would have had a reasonable expectation of success with combining Stokes and virtual machines, as the implementation of virtual machines was well-known and ubiquitous, and could be implemented using off-the-shelf software like VMWare or Windows. *Id.*

H. Claim 18

Although this petition does not challenge the validity of claim 18, Challenged Claims 19, 20, and 21 are dependent of Claim 18. Each limitation of Claim 18 is disclosed and/or rendered obvious by Stokes in view of Crabtree.

1. Elements 18.1-18.5, 18.7, and 18.9

Elements [18.1]-[18.5], [18.7], and [18.9] are substantively identical to elements [1.1]-[1.5], [1.6] and [1.8], respectively. EX1016, 1-2. Stokes-Crabtree satisfies those elements for the same reasons explained above for elements [1.1]-[1.5], [1.6], and [1.8] in Sections VII.D.1-VII.D.5, VII.D.6, and VII.D.8, *supra*. EX1003, ¶128.

2. Element 18.6

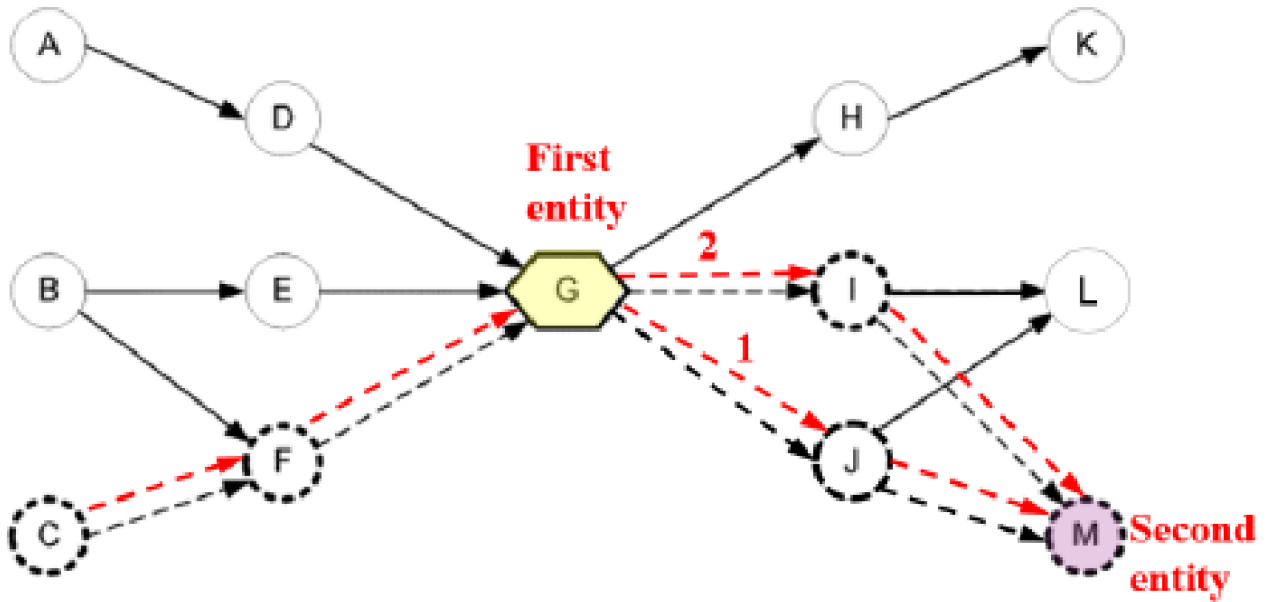
[18.6] identify, based on the representation of the first graph, a first plurality of attack paths comprising a first entity of the first plurality of entities, wherein each attack path of the first plurality of attack paths targets a second entity of the first plurality of entities,

As discussed above for element [1.10], Section VII.D.10 *supra*, Stokes teaches identifying plural paths through the graph that correspond to malicious lateral movement and thus correspond to the claimed “attack paths.” EX1004, [0003] (explaining that lateral movement “can be viewed as a set of malicious paths corresponding to the attacker’s activity”); *see also id.*, [0029], [0038]-[0039]. EX1003, ¶129.

Stokes provides an example of identifying such malicious paths in its Figure 3. Stokes explains that “in FIG. 3, a known compromised node (denoted as [node] G) is discovered and the task is to identify the unknown compromised computers and user accounts (dashed nodes C, F, I, L, and M) along the malicious access path (dashed lines).” EX1004, [0066]. As an illustrative example, a modified version of

Stokes's Figure 3 is provided below with annotations⁴ (red dashed lines), depicting an obvious use case of Stokes graph in a network setting. EX1003, ¶130. The modified Figure 3 shows a plurality of malicious paths identified using Stokes's techniques, including a first path (including nodes C, F, G, J, and M) and a second path (including nodes C, F, G, I, and M). In modified Figure 3, node G (representing a first entity included in the first and second paths) is highlighted in yellow, and node M (representing a second entity at the end of the first and second paths) is highlighted in purple. As shown, the first and second paths each include the edge between nodes C and F and the edge between nodes F and G; the paths then diverge downstream of node G, as indicated by the red numeric labels.

⁴ In this modified Stokes Figure 3, for ease of explanation, the I-L edge has been converted from a dashed line to a solid line; node L has been converted from a dashed node to a solid node; node J has been converted from a solid node to a dashed node; the G-J and J-M edges have been converted from solid lines to dashed lines; and the bracketed inbound and outbound path labels and time arrow have been omitted. EX1003, ¶130.



EX1003, ¶130, FIG. 3 (modified, with red annotations). As shown above, the first and second paths each comprise a first entity, represented by node G, and terminate at a second entity, represented by node M. Accordingly, node M corresponds to the claimed second entity targeted by the first plurality of attack paths. EX1003, ¶130.

Thus, Stokes teaches identifying, based on a representation of a first graph, a first plurality of attack paths that each comprise a first entity and target a second entity, where the first and second entities are among a plurality of entities represented by nodes in the graph. EX1003, ¶131.

3. Element 18.8

Element [18.8] differs from element [1.7] only in that the former recites a “third entity” where the latter recites a “first entity.” EX1016, 2. As explained for

element [1.7], *supra* Section VII.D.7, Stokes-Crabtree identifies a newly detected entity, not corresponding to any of the first plurality of nodes, in a first portion of streaming data. While such disclosure was mapped to a “first entity” for purposes of element [1.7], such disclosure equally satisfies the “third entity” for purposes of element [18.8] for the same reasons explained for element [1.7]. EX1003, ¶132.

4. Element 18.10

Element [18.10] differs from element [1.9] in that the former recites a “*third entity*,” a “*node corresponding to the third entity*,” and “*an edge corresponding to the first relationship*” where the latter recites a “*first entity*,” a “*first node corresponding to the first entity*,” and “*a first edge corresponding to the first relationship*.” EX1016, 2. As explained for element [1.9], *supra* Section VII.D.9, Stokes-Crabtree modifies a graph to include a new node corresponding to a newly detected entity and not of the first plurality of nodes, as well as updating the graph to include a new edge corresponding to a newly detected relationship and not of the first plurality of edges. While such disclosures were mapped to the “first entity,” “first node corresponding to the first entity,” and “a first edge corresponding to the first relationship” for purposes of element [1.9], such disclosures equally satisfy, respectively, the “third entity,” “node corresponding to the third entity,” and “an edge corresponding to the first relationship” for purposes of element [18.10] for the same reasons explained for element [1.9]. EX1003, ¶133.

5. Element 18.11

[18.11] identify, based on the modified representation of the first graph, a second plurality of attack paths comprising the first entity, wherein each attack path of the second plurality of attack paths targets the second entity, and wherein an attack path of the second plurality of attack paths comprises the third entity.

As discussed above for element [18.6], Section VII.H.2 *supra*, Stokes teaches identifying a first plurality of malicious lateral-movement paths through a network connection graph, i.e., attack paths, that comprise a first entity and target a second entity. EX1004, FIG. 3, [0003], [0029], [0038]-[0039], [0066]; EX1003, ¶134. As further discussed above for elements 18.8 and 18.10 (*supra* Sections VII.H.3-4), the Stokes-Crabtree combination teaches identifying a third entity from streaming data and modifying the graph to include a node corresponding to that third entity, where the third entity was not previously represented in the graph. EX1005, [0075]-[0081]; EX1003, ¶¶132-135.

A POSITA would have understood that, once the graph is modified to include the third entity, the same Stokes path-identification analysis would be applied to the modified graph to identify additional malicious lateral-movement paths supported by that updated graph structure. EX1004, [0038]-[0040], [0047], [0066]-[0071]; EX1003, ¶135. Because Stokes identifies attack paths by analyzing graph connectivity among nodes, the addition of a newly identified third entity/node

connected to the earlier-detected paths would lead to identification of a second plurality of attack paths that includes paths comprising the first entity and targeting the second entity, with at least one such path comprising the newly added third entity. EX1003, ¶135.

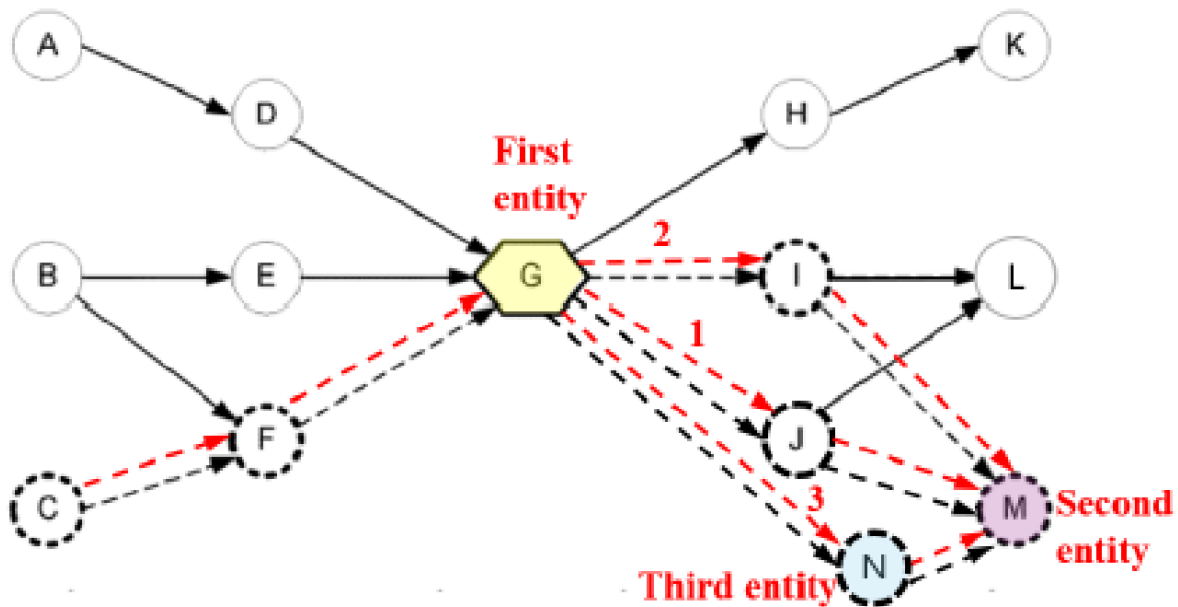
This follows directly from the combined teachings. Stokes teaches graph-based identification of malicious paths through connected nodes, while Crabtree teaches updating the graph as new entities and relationships are identified from incoming event data. EX1004, [0038]-[0040], [0047], [0066]-[0071]; EX1005, [0075]-[0081]. In the combined system, once the modified graph includes a node representing the third entity, a POSITA would have applied Stokes's path analysis to that modified graph to thereby identify a second plurality of attack paths that still comprises the first entity and targets the second entity, but now also includes at least one attack path that comprises the third entity. EX1003, ¶¶135-139.

As discussed above the Stokes-Crabtree combination teaches a modified graph representation that includes nodes corresponding to entities that were not represented by nodes in the graph representation before the modification, including the claimed third entity. *Supra* Sections VII.D.9 (elements 1.9) and VII.H.4 (18.10); EX1004, [0051]; EX1005, FIG. 1, [0050], [0081]; EX1003, ¶135. Where the newly added third entity/node corresponds to an unknown compromised computer or user account, the Stokes-Crabtree system would identify, based on the modified graph, a

malicious path that includes that third node/entity. EX1003, ¶135. Malicious lateral movements in a network involved a malicious actor exploring and accessing multiple resources and the Stokes-Crabtree graph would be updated to reflect such exploration and access. *Id.*, ¶135; EX1004, [0002] (“Next, the attacker begins to use lateral movement techniques to explore the network, i.e., the process an attacker uses to move from one computer to the next in a compromised network.”), [0031]; EX1017, 4:10-21. EX1003, ¶135.

The below modified Stokes Figure 3 illustrates one example of this type of attack path detection.⁵ Red dashed lines show a second plurality of attack paths including a first path (including nodes C, F, G, J, and M) and a second path (including nodes C, F, G, I, and M). Node G (a first entity) is yellow-highlighted and node M (a second entity) is purple-highlighted. Further, new node N (corresponding to a third, newly-identified entity) is highlighted in blue. In this modified example, Node N is in a third path of the second plurality of attack paths; the third path also includes nodes C, F, G, and M.

⁵ This modified Stokes Figure 3 includes the same modifications as above for element 18.6, and also adds new node N and new black dashed lines to represent the G-N edge and N-M edge, with bracketed inbound and outbound path labels and time arrows omitted. EX1003, ¶136.



EX1003, ¶136, FIG. 3 (modified, with red annotations). New node N represents the type of node that would be added to the graph upon identification of a new entity in the Stokes-Crabtree system. *Id.*, ¶136. In the above example, the first, second, and third paths each comprise the first entity (i.e., compromised node G) and target the second entity (node M). Furthermore, as shown, node N of the third path corresponds to a newly identified entity (i.e., the claimed third entity); thus, the third path shown in modified Figure 3 is an attack path of the second plurality of attack paths that comprises the third entity. EX1003, ¶135.

This modified Figure 3 illustrates one scenario in which an attacker, after unsuccessfully attempting to access the second entity (node M) through the entities corresponding to nodes I and J, tries to access a different entity in the network (for

example one having different access credentials), specifically the third entity (node N), in an attempt to reach the second entity via the third entity. EX1003, ¶137.

A POSITA would also recognize a malicious entity targeting the second entity (node M) through additional entities not represented by nodes in the second modified version of Figure 3. This would lead to the addition of nodes between nodes G and M that correspond to additional entities (i.e., in addition to node N), for use in identifying additional malicious paths including those entities. EX1003, ¶138.

Accordingly, Stokes-Crabtree teaches identifying, based on the modified representation of the first graph, a second plurality of attack paths comprising the first entity, wherein each attack path of the second plurality of attack paths targets the second entity, and wherein an attack path of the second plurality of attack paths comprises the third entity. EX1003, ¶139.

I. Claim 19

[19] The computer system of claim 18, wherein the first graph is a cyber-physical graph.
--

See Claim 9. EX1003, ¶140.

VIII. GROUND 2: CLAIMS 20 AND 21 ARE OBVIOUS OVER STOKES IN VIEW OF CRABTREE AND SEKAR

As explained below, Claims 20 and 21 are further obvious over Stokes in view of Crabtree and Sekar. EX1003, ¶141.

A. Sekar (EX1006)

Sekar is a U.S. patent application filed August 19, 2019, claims priority to a provisional application filed August 17, 2018, and thus qualifies as prior art under at least 35 U.S.C. § 102(a)(2). EX1006, Cover. Sekar is analogous art to the '627 patent because they are both in the same field of endeavor of graph-based detection of attack-related activity in computer networks. EX1001, Abstract, 2:52-4:13; EX1006, [0027]-[0034]. EX1003, ¶142.

Sekar discloses graph-based backward analysis for identifying the point of origin or entry point of an attack based on event flows or dependencies in a graph. In particular, Sekar describes processing suspect vertices and tracing backward to reachable source vertices to identify each entry point of an attack and further describes determining the attacker's initial point of entry through backward analysis before performing forward analysis to assess impact. EX1006, [0239], [0269]. EX1003, ¶143.

B. Combining Stokes, Crabtree, And Sekar

A POSITA would have found it obvious to implement a combination of Stokes and Crabtree, as discussed above at Section VII.C. A POSITA further would have been motivated, with a reasonable expectation of success, to combine the teachings of Stokes-Crabtree with Sekar's teachings regarding identifying a point of origin of an attack (hereinafter "Stokes-Crabtree-Sekar"). EX1003, ¶144.

First, Sekar, like Stokes and Crabtree, discloses graph-based cybersecurity analysis of attack-related activity. Stokes-Crabtree uses event-derived information to update and analyze graphs of entities and relationships to identify suspicious paths and other attack-related information. EX1004, [0037]-[0040], [0047]-[0048], [0064]-[0071]; EX1005, [0063]-[0064], [0072], [0076], [0080]-[0081]. Sekar likewise discloses graph-based attack analysis, including backward analysis to identify dependencies and each entry point of an attack and forward analysis to assess impact. EX1006, [0233]-[0239], [0269]. Sekar thus complements Stokes-Crabtree by providing explicit techniques for identifying a point of origin of an attack. EX1003, ¶145.

Second, a POSITA implementing Stokes-Crabtree would have recognized the need not only to identify suspicious paths and anomalous activity, but also to determine where that activity originated. Stokes already performs analysis of inbound paths in a graph to determine event flows. EX1004, [0048], [0064], [0070]. Sekar expressly teaches determining the initial point of entry of an attacker through backward analysis of event flows and dependencies, including source vertices from which suspect vertices are reachable. EX1006, [0233]-[0239], [0269]. A POSITA thus would have been motivated to incorporate Sekar's backward origin-identification techniques into Stokes-Crabtree to predictably improve identification of the point of origin for anomalous events and attack paths. A POSITA would

recognize the benefit of identifying the point of origin of an anomalous event because it would allow for remedial action against such source. *See* EX1004, [0061] (disclosing that “identified malicious computers and/or user accounts can be shut down (i.e., disabled)”). EX1003, ¶¶146-147.

Finally, a POSITA would have had a reasonable expectation of success in combining Stokes, Crabtree, and Sekar. EX1003, ¶148. The references operate in the same technical field of graph-based cybersecurity analysis and rely on compatible inputs, structures, and objectives. EX1004, [0037]-[0040], [0047]-[0048], [0064]-[0071]; EX1005, [0063]-[0064], [0072], [0076], [0080]-[0081]; EX1006, [0239], [0269]. Because Sekar’s point of origin identification techniques would have been applied to the same type of graph-based cybersecurity system disclosed in Stokes-Crabtree, incorporating Sekar’s teachings would involve the predictable use of known techniques in a closely related system, with no technological incompatibility or undue experimentation required. EX1003, ¶148.

C. Claim 20

1. Element 20.1

[20.1] The computer system of claim 18, wherein identifying the second plurality of attack paths comprises:

See Claim 18. EX1003, ¶149.

2. Element 20.2

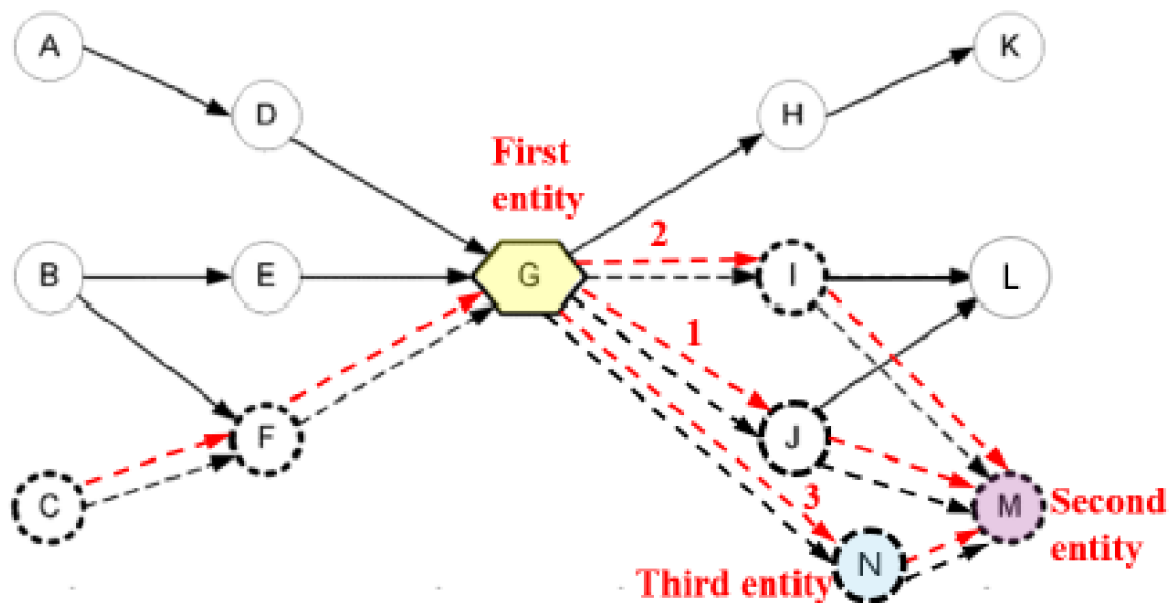
[20.2] identifying a first plurality of event flows that could involve the first entity, and

Stokes discloses “[g]raph-based detection systems and techniques ... to identify potential malicious lateral movement paths,” where “[a] series of lateral movements across computers can be seen as paths constructed by a sequence of edges in the same direction.” EX1004, [0020], [0025]. As edges in Stokes represent events (e.g., “computer connections and user logon events”), the disclosed “paths constructed by a sequence of edges in the same direction” satisfy the claimed “event flows.” *Id.*, [0006], [0025]; EX1003, ¶150. This is consistent with the ’627 patent’s description of “possible event flows” and “interaction pathways” through a behavior graph, which likewise treats event flows as directed pathways of events and interactions between affected nodes. EX1001, 21:54-22:42. EX1003, ¶150.

As explained for element [18.11], Section VII.H.5 *supra*, Stokes identifies malicious lateral movements involved in a potential attack in a graph, including inbound and outbound paths from a known compromised node, based on path-rate scores. *See* EX1004, [0067], [0069]. Stokes’s path-rate score module “generates all possible paths and calculates the corresponding path-rate score, *p*, for each path” (*Id.*, [0047]). This path rate score reflects “how rare (i.e., *anomalous*) a path is in a computer network” (*id.*, [0044] (emphasis added)), such that the “highest ranked paths... are the most rare and are more likely to correspond to malicious lateral

movement” (*id.*, [0047]). Accordingly, the identified malicious lateral movements through the first entity/node reflect an anomalous event flow involving the first entity. EX1003, ¶151.

With reference to the graph depicted in modified Figure 3 as discussed in element [18.11], reproduced below, Stokes describes that dashed lines indicate a malicious access path associated with known compromised node G (*first entity*) and explains that, “given a malicious node, possible lateral movement leading into or out of the node is identified” based on path-rate scores. EX1004, [0028], [0044], [0066]-[0067].



Id., FIG. 3. EX1003, ¶152. As shown above, modified Figure 3 depicts a plurality of event flows involving node G (i.e., a first entity/node), including a first event flow through nodes C-F-G-I-L and a second event flow through nodes C-F-G-I-M.

Accordingly, Stokes teaches identifying a plurality of event flows including an anomalous event associated with a first entity/node. EX1003, ¶153.

3. Element 20.3

[20.3] identifying a first plurality of dependencies associated with the first plurality of event flows.

As explained for elements [18.11] and [20.2], Sections VII.H.5 and VIII.C.2 *supra*, Stokes utilizes a graph of nodes (including a known compromised node) and edges and performs forensic analysis using the graph to “identify the unknown compromised computers and user accounts... along the malicious access path.” EX1004, [0066]-[0071]. And, Stokes explains that “[s]earching for potentially compromised nodes ... considers inbound paths,” meaning that Stokes identifies the upstream series of edges and nodes, along a malicious access path leading to the compromised node, which would result in identifying a point of origin for an anomalous event. *Id.*, [0066]. EX1003, ¶155.

As explained in Section VIII.A, *supra*, Sekar also discloses identifying a point of origin of an attack based on event flows. EX1006, [0239]. In particular, Sekar describes backward analysis in which the system aims to identify each entry point of an attack, where an entry point is a source vertex, and processes nodes to find source vertices from which a suspect vertex is reachable. *Id.* Sekar’s backward analysis is performed using an algorithm that assigns costs to edges based on their

“dependencies,” with the idea being to identify those edges which are likely to be part of the attack path. EX1006, [0233]-[0239]; *see also* [0078] (“Dependence graphs are used to trace back to the root causes of intrusions”). Edges are assigned a low cost if they “introduce a dependency from a node with unknown code ... to a node with benign code” or if they are between nodes with “unknown” code, and edges are assigned a high cost if they “introduce[e] a dependency from a node with benign code.” EX1006, [0234]-[0236]. Sekar explains that an edge between a benign node and untrusted node “represents the boundary between the malicious and benign portions of the graph” and so “must be included in the search” by giving it a low weight. EX1006, [0237]. By contrast, edges between benign nodes “are not part of the attack” and so are assigned a high cost. *Id.* Thus, Sekar teaches identifying dependencies between nodes. EX1003, ¶155.

D. Claim 21

1. Element 21.1

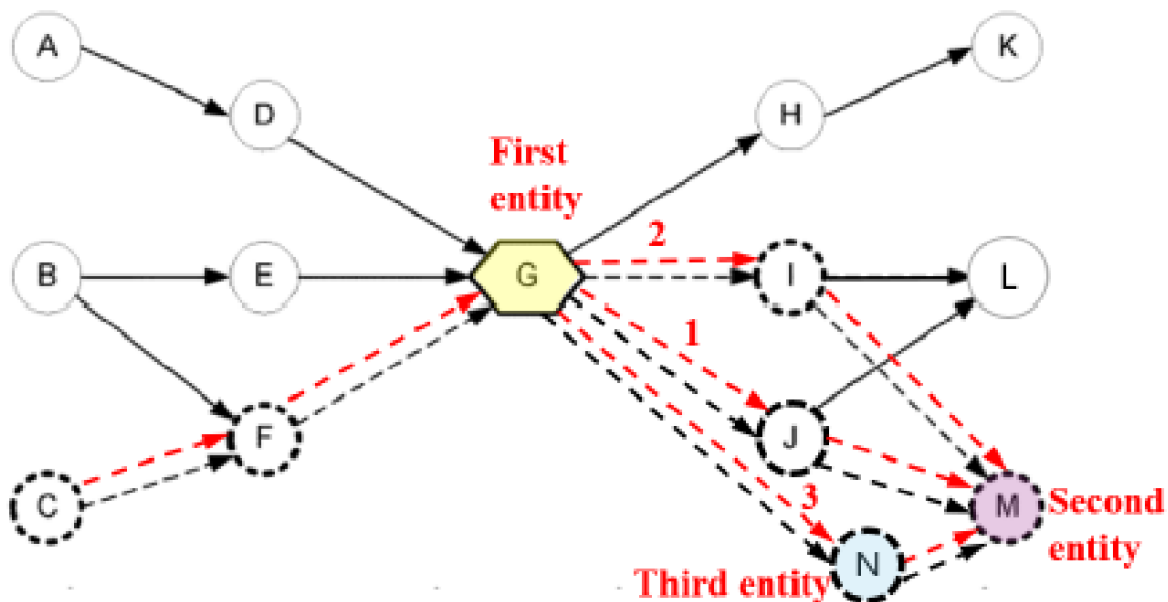
[21.1] The computer system of claim 20, further configured to execute software instructions stored on nontransitory machine-readable storage media comprising software instructions that:

See Claim 20 and Element [18.1]. EX1003, ¶156.

2. Element 21.2

[21.2] based on a third portion of the streaming data, identify an anomalous event,

As explained for element [20.2], Section VIII.C.2, *supra*, Stokes teaches identifying malicious lateral movements through nodes, which are ***anomalous events***. EX1004, [0044], [0047]. EX1003, ¶157. For example, for the below modified Figure 3, Stokes describes that dashed lines indicate a malicious access path associated with known compromised node G (*first entity*) and explains that, “given a malicious node, possible lateral movement leading into or out of the node is identified” based on path-rate scores. EX1004, [0028], [0044], [0066]-[0067]. The modified Figure 3 depicts a plurality of event flows involving node G (i.e., a first entity/node), including a first event flow through nodes C-F-G-I-L and a second event flow through nodes C-F-G-I-M. Accordingly, Stokes teaches identifying a plurality of event flows including an anomalous event associated with a first entity/node.



The data that is used to identify anomalous events is therefore the edges of the malicious access path, which are received from ongoing *streaming event data*, as discussed above in element [1.6], Section VII.D.6, *supra*. EX1003, ¶158.

The anomalous events associated with a first entity are identified by a *third portion of the streaming data*. As discussed above for elements [18.8] and [18.9], the first and second portions of streaming data do not correspond with the nodes and edges of the first graph, which includes the first entity and its edges. Accordingly, the anomalous events associated with the first entity are not identified by the first and second portion of the streaming data and therefore are identified by a *third portion*. EX1003, ¶159.

3. Element 21.3

[21.3] store in the hardware memory a dependency tree for the anomalous event, and

As discussed above for element [20.3], Section VIII.C.3 *supra*, Sekar discloses (which would be obvious to combine with Stokes-Crabtree) using backward analysis to identify the origin of an anomaly, by using an algorithm that assigns costs to edges based on their “dependencies,” with the idea being to identify those edges which are likely to be part of the attack path. EX1006, [0233]-[0239]; *see also* [0078] (“Dependence graphs are used to trace back to the root causes of intrusions”); EX1003, ¶160.

Sekar teaches that such dependencies are “store[d] ... in a graph data structure.” EX1006, [0084]. When analyzing the dependency graph to determine the origin of an anomalous event, a shortest path algorithm is used, where “each step of this algorithm adds a node to the shortest path *tree*,” *i.e.*, *a dependency tree*. EX1006, [0222]. EX1003, ¶161. As discussed above for element [20.3], the shortest path is determined from weights assigned to edges based on their dependencies, and therefore the shortest path tree is a *dependency tree for the anomalous event*. *Id.*, [0233]-[0238]; EX1003, ¶161.

Sekar teaches that the dependency graph is stored in “main memory,” *i.e.*, *hardware memory*. EX1006, [0084]. Accordingly, a POSITA would understand that the dependency tree derived from the dependency graph is likewise stored in “main memory.” EX1003, ¶162.

4. Element 21.4

[21.4] based on the dependency tree, determine a point of origin for the anomalous event.

Sekar teaches that the shortest path tree, *i.e.* *dependency tree*, is used to determine the node of origin of the anomalous event. Specifically, Sekar teaches that when performing backward analysis, each step of the algorithm “adds a node to the shortest path tree, which consists of the shortest paths computed thus far.” EX1006, [0222]; EX1003, ¶163. This “enables the search to stop as soon as an *entry point node* is added to the tree.” *Id.* The “entry point node” is the *point of origin*

for the anomalous event. Id., [0215] (“The general goal of backward analysis is to identify the entry points of an attack campaign.”).

IX. NO OBJECTIVE INDICIA OF NON-OBVIOUSNESS

Petitioner is unaware of any evidence of objective indicia of non-obviousness having a nexus to the Challenged Claims. EX1003, ¶¶164-68.

X. CONCLUSION

Petitioner requests institution for Claims 9, 10, 15, 19, 20, and 21 specified in this petition.

Date: July 20, 2026

Respectfully submitted,

By: /s/ James M. Glass

James M. Glass
QUINN EMANUEL URQUHART
& SULLIVAN, LLP
295 5th Ave., 9th Floor
New York, NY 10016
Tel: (212) 849-7000
Fax: (212) 849-7100

Counsel for Petitioner

**CERTIFICATE OF COMPLIANCE WITH
TYPE-VOLUME LIMITATION, TYPEFACE REQUIREMENTS,
AND TYPE STYLE REQUIREMENTS**

1. This Petition complies with the type-volume limitation of 14,000 words, comprising 11,301 words, as counted using the Microsoft Word software that was used to prepare this paper, excluding the parts exempted by 37 C.F.R. § 42.24(a).

2. This Petition complies with the general format requirements of 37 C.F.R. § 42.6(a) and has been prepared using Microsoft® Word 2016 in 14-point Times New Roman.

Date: July 20, 2026

By: /s/ James M. Glass
James M. Glass
QUINN EMANUEL URQUHART
& SULLIVAN, LLP
295 5th Ave., 9th Floor
New York, NY 10016
Tel: (212) 849-7000
Fax: (212) 849-7100

Counsel for Petitioner

CERTIFICATE OF SERVICE

The undersigned hereby certifies that on July 7, 2026, true and correct copies of the foregoing document and supporting materials were served in its entirety on the Patent Owner at the following address of record as listed on PAIR via First-Class Mail:

NK Patent Law
4101 Lake Boone Trail, Suite 218
Raleigh, NC 27607

Official Correspondence Address

Courtesy copies were also sent via electronic mail to Patent Owner's counsel of record in the related district court proceeding

aberge@irell.com
cwen@irell.com
dslusarczyk@irell.com
iwashburn@irell.com
claire@millerfairhenry.com

Date: July 20, 2026

By: /s/ James M. Glass
James Glass
QUINN EMANUEL URQUHART
& SULLIVAN, LLP
295 5th Ave., 9th Floor
New York, NY 10016
Tel: (212) 849-7000
Fax: (212) 849-7100

Counsel for Petitioner