



US010659324B2

(12) **United States Patent**  
**Pang et al.**

(10) **Patent No.:** **US 10,659,324 B2**  
(45) **Date of Patent:** **May 19, 2020**

(54) **APPLICATION MONITORING  
PRIORITIZATION**

(56) **References Cited**

(71) Applicant: **Cisco Technology, Inc.**, San Jose, CA (US)  
(72) Inventors: **Jackson Ngoc Ki Pang**, Sunnyvale, CA (US); **Navindra Yadav**, Cupertino, CA (US); **Anubhav Gupta**, Sunnyvale, CA (US); **Shashidhar Gandham**, Fremont, CA (US); **Supreeth Hosur Nagesh Rao**, Cupertino, CA (US); **Sunil Kumar Gupta**, Milpitas, CA (US)

U.S. PATENT DOCUMENTS  
5,086,385 A 2/1992 Launey et al.  
5,319,754 A 6/1994 Meinecke et al.  
(Continued)

FOREIGN PATENT DOCUMENTS  
CN 101093452 12/2007  
CN 101770551 7/2010  
(Continued)

(73) Assignee: **CISCO TECHNOLOGY, INC.**, San Jose, CA (US)

OTHER PUBLICATIONS

(\*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 80 days.

Sarah Heckman; On Establishing a Benchmark for Evaluating Static Analysis Alert Prioritization and Classification Techniques; IEEE 2008; p. 1-10.\*  
(Continued)

(21) Appl. No.: **15/173,477**

(22) Filed: **Jun. 3, 2016**

*Primary Examiner* — Monjur Rahim

(74) *Attorney, Agent, or Firm* — Polsinelli PC

(65) **Prior Publication Data**

US 2016/0359891 A1 Dec. 8, 2016

**Related U.S. Application Data**

(60) Provisional application No. 62/171,899, filed on Jun. 5, 2015.

(51) **Int. Cl.**  
**G08B 23/00** (2006.01)  
**G06F 12/16** (2006.01)  
(Continued)

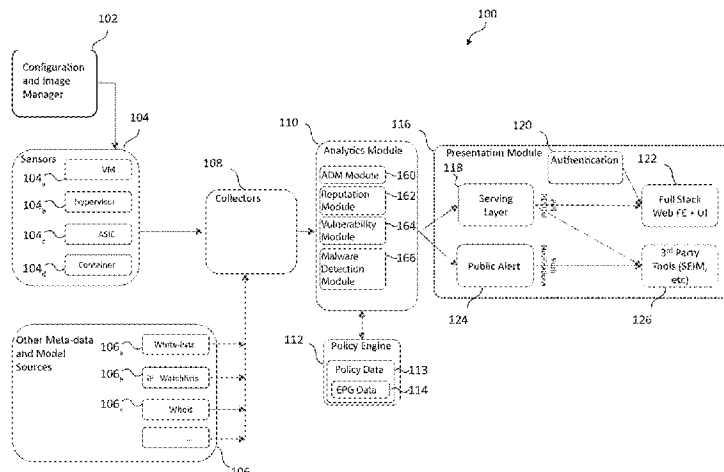
(52) **U.S. Cl.**  
CPC ..... **H04L 43/045** (2013.01); **G06F 3/0482** (2013.01); **G06F 3/04842** (2013.01);  
(Continued)

(58) **Field of Classification Search**  
CPC ..... H04L 43/045; H04L 41/0668; H04L 41/0816; H04L 67/22; G06F 3/0482;  
(Continued)

(57) **ABSTRACT**

An approach for establishing a priority ranking for endpoints in a network. This can be useful when triaging endpoints after an endpoint becomes compromised. Ensuring that the most critical and vulnerable endpoints are triaged first can help maintain network stability and mitigate damage to endpoints in the network after an endpoint is compromised. The present technology involves determining a criticality ranking and a secondary value for a first endpoint in a datacenter. The criticality ranking and secondary value can be combined to form priority ranking for the first endpoint which can then be compared to a priority ranking for a second endpoint to determine if the first endpoint or the second endpoint should be triaged first.

**20 Claims, 7 Drawing Sheets**



- (51) **Int. Cl.**  
*G06F 12/14* (2006.01); *G06F 11/00* (2006.01); *H04L 12/26* (2006.01); *H04L 29/06* (2006.01); *G06F 9/455* (2018.01); *G06N 20/00* (2019.01); *G06F 16/29* (2019.01); *G06F 16/248* (2019.01); *G06F 16/28* (2019.01); *G06F 16/9535* (2019.01); *G06F 16/2457* (2019.01); *G06F 21/55* (2013.01); *G06F 21/56* (2013.01); *H04L 12/851* (2013.01); *H04L 12/24* (2006.01); *H04W 84/18* (2009.01); *H04L 29/08* (2006.01); *G06F 21/53* (2013.01); *H04L 12/723* (2013.01); *G06F 3/0484* (2013.01); *H04L 1/24* (2006.01); *H04W 72/08* (2009.01); *H04L 9/08* (2006.01); *H04L 9/32* (2006.01); *H04J 3/06* (2006.01); *H04J 3/14* (2006.01); *H04L 29/12* (2006.01); *H04L 12/813* (2013.01); *H04L 12/823* (2013.01); *H04L 12/801* (2013.01); *H04L 12/741* (2013.01); *H04L 12/833* (2013.01); *H04L 12/721* (2013.01); *G06F 3/0482* (2013.01); *G06T 11/20* (2006.01); *H04L 12/841* (2013.01); *H04L 12/725* (2013.01); *H04L 12/715* (2013.01); *G06F 16/16* (2019.01); *G06F 16/17* (2019.01); *G06F 16/11* (2019.01); *G06F 16/13* (2019.01); *G06N 99/00* (2019.01); *G06F 16/174* (2019.01); *G06F 16/23* (2019.01); *9/3242* (2013.01); *H04L 41/046* (2013.01); *H04L 41/0668* (2013.01); *H04L 41/0803* (2013.01); *H04L 41/0806* (2013.01); *H04L 41/0816* (2013.01); *H04L 41/0893* (2013.01); *H04L 41/12* (2013.01); *H04L 41/16* (2013.01); *H04L 41/22* (2013.01); *H04L 43/02* (2013.01); *H04L 43/04* (2013.01); *H04L 43/062* (2013.01); *H04L 43/08* (2013.01); *H04L 43/0805* (2013.01); *H04L 43/0811* (2013.01); *H04L 43/0829* (2013.01); *H04L 43/0841* (2013.01); *H04L 43/0858* (2013.01); *H04L 43/0864* (2013.01); *H04L 43/0876* (2013.01); *H04L 43/0882* (2013.01); *H04L 43/0888* (2013.01); *H04L 43/10* (2013.01); *H04L 43/106* (2013.01); *H04L 43/12* (2013.01); *H04L 43/16* (2013.01); *H04L 45/306* (2013.01); *H04L 45/38* (2013.01); *H04L 45/46* (2013.01); *H04L 45/507* (2013.01); *H04L 45/66* (2013.01); *H04L 45/74* (2013.01); *H04L 47/11* (2013.01); *H04L 47/20* (2013.01); *H04L 47/2441* (2013.01); *H04L 47/2483* (2013.01); *H04L 47/28* (2013.01); *H04L 47/31* (2013.01); *H04L 47/32* (2013.01); *H04L 61/2007* (2013.01); *H04L 63/0227* (2013.01); *H04L 63/0263* (2013.01); *H04L 63/06* (2013.01); *H04L 63/0876* (2013.01); *H04L 63/145* (2013.01); *H04L 63/1408* (2013.01); *H04L 63/1416* (2013.01); *H04L 63/1425* (2013.01); *H04L 63/1433* (2013.01); *H04L 63/1441* (2013.01); *H04L 63/1458* (2013.01); *H04L 63/1466* (2013.01); *H04L 63/16* (2013.01); *H04L 63/20* (2013.01); *H04L 67/10* (2013.01); *H04L 67/1002* (2013.01); *H04L 67/12* (2013.01); *H04L 67/16* (2013.01); *H04L 67/36* (2013.01); *H04L 67/42* (2013.01); *H04L 69/16* (2013.01); *H04L 69/22* (2013.01); *H04W 72/08* (2013.01); *H04W 84/18* (2013.01); *G06F 2009/4557* (2013.01); *G06F 2009/45587* (2013.01); *G06F 2009/45591* (2013.01); *G06F 2009/45595* (2013.01); *G06F 2221/033* (2013.01); *G06F 2221/2101* (2013.01); *G06F 2221/2105* (2013.01); *G06F 2221/2111* (2013.01); *G06F 2221/2115* (2013.01); *G06F 2221/2145* (2013.01); *H04L 67/22* (2013.01)
- (52) **U.S. Cl.**  
 CPC ..... *G06F 3/04847* (2013.01); *G06F 9/45558* (2013.01); *G06F 16/122* (2019.01); *G06F 16/137* (2019.01); *G06F 16/162* (2019.01); *G06F 16/17* (2019.01); *G06F 16/173* (2019.01); *G06F 16/174* (2019.01); *G06F 16/1744* (2019.01); *G06F 16/1748* (2019.01); *G06F 16/235* (2019.01); *G06F 16/2322* (2019.01); *G06F 16/2365* (2019.01); *G06F 16/248* (2019.01); *G06F 16/24578* (2019.01); *G06F 16/285* (2019.01); *G06F 16/288* (2019.01); *G06F 16/29* (2019.01); *G06F 16/9535* (2019.01); *G06F 21/53* (2013.01); *G06F 21/552* (2013.01); *G06F 21/566* (2013.01); *G06N 20/00* (2019.01); *G06N 99/00* (2013.01); *G06T 11/206* (2013.01); *H04J 3/0661* (2013.01); *H04J 3/14* (2013.01); *H04L 1/242* (2013.01); *H04L 9/0866* (2013.01); *H04L 9/3239* (2013.01); *H04L*
- (58) **Field of Classification Search**  
 CPC ..... *G06F 3/04842*; *G06F 3/04847*; *G06F 9/45558*; *G06F 17/30241*; *G06F 17/30604*; *G06F 21/53*; *H04W 72/08*; *H04J 3/14*; *G06T 11/206*  
 USPC ..... 726/23  
 See application file for complete search history.
- (56) **References Cited**  
 U.S. PATENT DOCUMENTS  
 5,400,246 A 3/1995 Wilson et al.  
 5,436,909 A 7/1995 Dev et al.  
 5,555,416 A 9/1996 Owens et al.  
 5,726,644 A 3/1998 Jednacz et al.  
 5,742,829 A 4/1998 Davis et al.  
 5,822,731 A 10/1998 Schultz  
 5,831,848 A 11/1998 Rielly et al.  
 5,903,545 A 5/1999 Sabourin et al.  
 6,012,096 A 1/2000 Link et al.  
 6,141,595 A 10/2000 Gloudeman et al.  
 6,144,962 A 11/2000 Weinberg et al.

(56)

References Cited

U.S. PATENT DOCUMENTS

6,239,699	B1	5/2001	Ronnen	8,040,232	B2	10/2011	Oh et al.
6,247,058	B1	6/2001	Miller et al.	8,040,822	B2	10/2011	Proulx et al.
6,249,241	B1	6/2001	Jordan et al.	8,115,617	B2	2/2012	Thubert et al.
6,330,562	B1	12/2001	Boden et al.	8,135,657	B2	3/2012	Kapoor et al.
6,353,775	B1	3/2002	Nichols	8,156,430	B2	4/2012	Newman
6,525,658	B2	2/2003	Streetman et al.	8,160,063	B2	4/2012	Maltz et al.
6,546,420	B1	4/2003	Lemler et al.	8,179,809	B1	5/2012	Eppstein et al.
6,611,896	B1	8/2003	Mason, Jr. et al.	8,185,824	B1	5/2012	Mitchell et al.
6,654,750	B1	11/2003	Adams et al.	8,239,365	B2	8/2012	Salman
6,728,779	B1	4/2004	Griffin et al.	8,239,915	B1	8/2012	Satish et al.
6,801,878	B1	10/2004	Hintz et al.	8,250,657	B1	8/2012	Nachenberg et al.
6,816,461	B1	11/2004	Scrandis et al.	8,255,972	B2	8/2012	Azagury et al.
6,847,993	B1	1/2005	Novaes et al.	8,266,697	B2	9/2012	Coffinan
6,848,106	B1	1/2005	Hipp	8,272,875	B1	9/2012	Jurmain
6,925,490	B1	8/2005	Novaes et al.	8,281,397	B2	10/2012	Vaidyanathan et al.
6,958,998	B2	10/2005	Shorey	8,291,495	B1*	10/2012	Burns ..... H04L 63/0254 455/410
6,983,323	B2	1/2006	Cantrell et al.	8,296,847	B2	10/2012	Mendonca et al.
6,996,817	B2	2/2006	Birum et al.	8,311,973	B1	11/2012	Zadeh
6,999,452	B1	2/2006	Drummond-Murray et al.	8,370,407	B1	2/2013	Devarajan et al.
7,002,464	B2	2/2006	Bruemmer et al.	8,381,289	B1	2/2013	Pereira et al.
7,024,468	B1	4/2006	Meyer et al.	8,391,270	B2	3/2013	Van Der Stok et al.
7,111,055	B2	9/2006	Falkner	8,407,164	B2	3/2013	Malik et al.
7,120,934	B2	10/2006	Ishikawa	8,407,798	B1	3/2013	Lotem et al.
7,133,923	B2	11/2006	MeLampy et al.	8,442,073	B2	5/2013	Skubacz et al.
7,181,769	B1	2/2007	Keanini et al.	8,451,731	B1	5/2013	Lee et al.
7,185,103	B1	2/2007	Jain	8,462,212	B1	6/2013	Kundu et al.
7,203,740	B1	4/2007	Putzolu et al.	8,489,765	B2	7/2013	Vasseur et al.
7,302,487	B2	11/2007	Ylonen et al.	8,499,348	B1	7/2013	Rubin
7,349,761	B1	3/2008	Cruse	8,516,590	B1	8/2013	Ranadive et al.
7,353,511	B1	4/2008	Ziese	8,527,977	B1	9/2013	Cheng et al.
7,356,679	B1	4/2008	Le et al.	8,570,861	B1	10/2013	Brandwine et al.
7,360,072	B1	4/2008	Soltis et al.	8,572,600	B2	10/2013	Chung et al.
7,370,092	B2	5/2008	Aderton et al.	8,572,734	B2	10/2013	McConnell et al.
7,395,195	B2	7/2008	Suenbuel et al.	8,572,735	B2	10/2013	Ghosh et al.
7,444,404	B2	10/2008	Wetherall et al.	8,588,081	B2	11/2013	Salam et al.
7,466,681	B2	12/2008	Ashwood-Smith et al.	8,600,726	B1	12/2013	Varshney et al.
7,467,205	B1	12/2008	Dempster et al.	8,615,803	B2	12/2013	Dacier et al.
7,496,040	B2	2/2009	Seo	8,630,316	B2	1/2014	Haba
7,496,575	B2	2/2009	Buccella et al.	8,640,086	B2	1/2014	Bonev et al.
7,530,105	B2	5/2009	Gilbert et al.	8,656,493	B2	2/2014	Capalik
7,539,770	B2	5/2009	Meier	8,661,544	B2	2/2014	Yen et al.
7,568,107	B1	7/2009	Rathi et al.	8,677,487	B2	3/2014	Balupari et al.
7,610,330	B1	10/2009	Quinn et al.	8,683,389	B1	3/2014	Bar-Yam et al.
7,633,942	B2	12/2009	Bearden et al.	8,706,914	B2	4/2014	Duchesneau
7,644,438	B1	1/2010	Dash et al.	8,719,452	B1	5/2014	Ding et al.
7,676,570	B2	3/2010	Levy et al.	8,719,835	B2	5/2014	Kanso et al.
7,681,131	B1	3/2010	Quarterman et al.	8,750,287	B2	6/2014	Bui et al.
7,693,947	B2	4/2010	Judge et al.	8,752,042	B2	6/2014	Ratica
7,743,242	B2	6/2010	Oberhaus et al.	8,755,396	B2	6/2014	Sindhu et al.
7,752,307	B2	7/2010	Takara	8,762,951	B1	6/2014	Kosche et al.
7,774,498	B1*	8/2010	Kraemer ..... H04L 63/20 709/224	8,769,084	B2	7/2014	Westerfeld et al.
7,783,457	B2	8/2010	Cunningham	8,775,577	B1	7/2014	Alford et al.
7,787,480	B1	8/2010	Mehta et al.	8,776,180	B2	7/2014	Kumar et al.
7,788,477	B1	8/2010	Huang et al.	8,812,448	B1	8/2014	Anderson et al.
7,808,897	B1	10/2010	Mehta et al.	8,812,725	B2	8/2014	Kulkarni
7,813,822	B1	10/2010	Hoffberg	8,813,236	B1*	8/2014	Saha ..... H04L 63/1408 726/22
7,844,696	B2	11/2010	Labovitz et al.	8,825,848	B1	9/2014	Dotan et al.
7,844,744	B2	11/2010	Abercrombie et al.	8,832,013	B1*	9/2014	Adams ..... G06N 5/02 706/46
7,864,707	B2	1/2011	Dimitropoulos et al.	8,832,461	B2	9/2014	Sarouj et al.
7,873,025	B2	1/2011	Patel et al.	8,849,926	B2	9/2014	Marzencki et al.
7,873,074	B1	1/2011	Boland	8,881,258	B2	11/2014	Paul et al.
7,874,001	B2	1/2011	Beck et al.	8,887,238	B2	11/2014	Howard et al.
7,885,197	B2	2/2011	Metzler	8,904,520	B1	12/2014	Nachenberg et al.
7,895,649	B1*	2/2011	Brook ..... H04L 43/026 709/224	8,908,685	B2	12/2014	Patel et al.
7,904,420	B2	3/2011	Ianni	8,914,497	B1	12/2014	Xiao et al.
7,930,752	B2	4/2011	Hertzog et al.	8,931,043	B2	1/2015	Cooper et al.
7,934,248	B1	4/2011	Yehuda et al.	8,954,610	B2	2/2015	Berke et al.
7,957,934	B2	6/2011	Greifeneder	8,966,021	B1	2/2015	Allen
7,961,637	B2	6/2011	McBeath	8,973,147	B2*	3/2015	Pearcy ..... H04L 41/0893 726/25
7,970,946	B1	6/2011	Djabarov et al.	8,990,386	B2	3/2015	He et al.
7,975,035	B2	7/2011	Popescu et al.	8,996,695	B2	3/2015	Anderson et al.
8,001,610	B1	8/2011	Chickering et al.	8,997,227	B1	3/2015	Mhatre et al.
8,005,935	B2	8/2011	Pradhan et al.	9,014,047	B2	4/2015	Alcala et al.
				9,015,716	B2	4/2015	Fletcher et al.
				9,071,575	B2	6/2015	Lemaster et al.

(56)

References Cited

U.S. PATENT DOCUMENTS

9,088,598 B1	7/2015	Zhang et al.	2005/0198247 A1	9/2005	Perry et al.	
9,110,905 B2	8/2015	Polley et al.	2005/0198371 A1	9/2005	Smith et al.	
9,130,836 B2	9/2015	Kapadia et al.	2005/0198629 A1	9/2005	Vishwanath	
9,160,764 B2	10/2015	Stiansen et al.	2005/0207376 A1	9/2005	Ashwood-Smith et al.	
9,170,917 B2	10/2015	Kumar et al.	2005/0257244 A1	11/2005	Joly et al.	
9,178,906 B1	11/2015	Chen et al.	2005/0289244 A1	12/2005	Sahu et al.	
9,191,400 B1	11/2015	Ptasinski et al.	2006/0048218 A1	3/2006	Lingafelt et al.	
9,197,654 B2	11/2015	Ben-Shalom et al.	2006/0077909 A1	4/2006	Saleh et al.	
9,225,793 B2	12/2015	Dutta et al.	2006/0080733 A1	4/2006	Khosmood et al.	
9,237,111 B2	1/2016	Banavalikar et al.	2006/0089985 A1	4/2006	Poletto	
9,246,702 B1	1/2016	Sharma et al.	2006/0095968 A1	5/2006	Portolani et al.	
9,246,773 B2	1/2016	Degioanni	2006/0143432 A1	6/2006	Rothman et al.	
9,253,042 B2	2/2016	Lumezanu et al.	2006/0156408 A1*	7/2006	Himberger .....	G06F 21/552 726/25
9,253,206 B1	2/2016	Fleischman	2006/0159032 A1	7/2006	Ukrainetz et al.	
9,258,217 B2	2/2016	Duffield et al.	2006/0173912 A1	8/2006	Lindvall et al.	
9,281,940 B2	3/2016	Matsuda et al.	2006/0195448 A1	8/2006	Newport	
9,286,047 B1	3/2016	Avramov et al.	2006/0272018 A1	11/2006	Fouant	
9,317,574 B1	4/2016	Brisebois et al.	2006/0274659 A1	12/2006	Ouderkirk	
9,319,384 B2	4/2016	Yan et al.	2006/0280179 A1	12/2006	Meier	
9,369,435 B2	6/2016	Short et al.	2006/0294219 A1	12/2006	Ogawa et al.	
9,378,068 B2	6/2016	Anantharam et al.	2007/0014275 A1	1/2007	Bettink et al.	
9,405,903 B1	8/2016	Xie et al.	2007/0025306 A1	2/2007	Cox et al.	
9,417,985 B2	8/2016	Baars et al.	2007/0044147 A1	2/2007	Choi et al.	
9,418,222 B1	8/2016	Rivera et al.	2007/0097976 A1	5/2007	Wood et al.	
9,426,068 B2	8/2016	Dunbar et al.	2007/0118654 A1	5/2007	Jamkhedkar et al.	
9,454,324 B1	9/2016	Madhavapeddi	2007/0127491 A1	6/2007	Verzipp et al.	
9,462,013 B1	10/2016	Boss et al.	2007/0162420 A1	7/2007	Ou et al.	
9,465,696 B2	10/2016	McNeil et al.	2007/0169179 A1	7/2007	Narad	
9,501,744 B1	11/2016	Brisebois et al.	2007/0195729 A1	8/2007	Li et al.	
9,531,589 B2	12/2016	Clemm et al.	2007/0195794 A1	8/2007	Fujita et al.	
9,563,517 B1	2/2017	Natanzon et al.	2007/0201474 A1	8/2007	Isobe	
9,621,413 B1	4/2017	Lee	2007/0211637 A1	9/2007	Mitchell	
9,634,915 B2	4/2017	Bley	2007/0214348 A1	9/2007	Danielsen	
9,645,892 B1	5/2017	Patwardhan	2007/0230415 A1	10/2007	Malik	
9,684,453 B2	6/2017	Holt et al.	2007/0232265 A1	10/2007	Park et al.	
9,697,033 B2	7/2017	Koponen et al.	2007/0250930 A1	10/2007	Aziz et al.	
9,733,973 B2	8/2017	Prasad et al.	2007/0300061 A1	12/2007	Kim et al.	
9,749,145 B2	8/2017	Banavalikar et al.	2008/0002697 A1	1/2008	Anantharamaiah et al.	
9,800,608 B2	10/2017	Korsunsky et al.	2008/0022385 A1	1/2008	Crowell et al.	
9,904,584 B2	2/2018	Konig et al.	2008/0028389 A1	1/2008	Genty et al.	
9,916,538 B2	3/2018	Zadeh et al.	2008/0040088 A1*	2/2008	Vankov .....	H04L 41/0681 703/13
9,935,851 B2	4/2018	Gandham et al.	2008/0046708 A1	2/2008	Fitzgerald et al.	
10,009,240 B2	6/2018	Rao et al.	2008/0049633 A1	2/2008	Edwards et al.	
2001/0028646 A1	10/2001	Arts et al.	2008/0082662 A1	4/2008	Dandliker et al.	
2002/0053033 A1	5/2002	Cooper et al.	2008/0101234 A1	5/2008	Nakil et al.	
2002/0097687 A1	7/2002	Meiri et al.	2008/0120350 A1	5/2008	Grabowski et al.	
2002/0103793 A1	8/2002	Koller et al.	2008/0126534 A1	5/2008	Mueller et al.	
2002/0107857 A1	8/2002	Teraslinna	2008/0141246 A1	6/2008	Kuck et al.	
2002/0141343 A1	10/2002	Bays	2008/0155245 A1	6/2008	Lipscombe et al.	
2002/0184393 A1	12/2002	Leddy et al.	2008/0250122 A1	10/2008	Zsigmond et al.	
2003/0023601 A1	1/2003	Fortier, Jr. et al.	2008/0270199 A1	10/2008	Chess et al.	
2003/0065986 A1	4/2003	Fraenkel et al.	2008/0282347 A1	11/2008	Dadhia et al.	
2003/0097439 A1	5/2003	Strayer et al.	2008/0295163 A1	11/2008	Kang	
2003/0126242 A1	7/2003	Chang	2008/0301765 A1	12/2008	Nicol et al.	
2003/0145232 A1	7/2003	Poletto et al.	2009/0059934 A1	3/2009	Aggarwal et al.	
2003/0151513 A1	8/2003	Herrmann et al.	2009/0064332 A1	3/2009	Porras et al.	
2003/0154399 A1	8/2003	Zuk et al.	2009/0077666 A1*	3/2009	Chen .....	G06F 21/577 726/25
2003/0177208 A1	9/2003	Harvey, IV	2009/0109849 A1	4/2009	Wood et al.	
2004/0019676 A1	1/2004	Iwatsuki et al.	2009/0138590 A1	5/2009	Lee et al.	
2004/0030776 A1	2/2004	Cantrell et al.	2009/0180393 A1	7/2009	Nakamura	
2004/0213221 A1	10/2004	Civanlar et al.	2009/0241170 A1	9/2009	Kumar et al.	
2004/0220984 A1	11/2004	Dudfield et al.	2009/0292795 A1	11/2009	Ford et al.	
2004/0243533 A1	12/2004	Dempster et al.	2009/0296593 A1	12/2009	Prescott	
2004/0255050 A1	12/2004	Takehiro et al.	2009/0300180 A1	12/2009	Dehaan et al.	
2004/0268149 A1	12/2004	Aaron	2009/0307753 A1	12/2009	Dupont et al.	
2005/0028154 A1	2/2005	Smith et al.	2009/0313373 A1	12/2009	Hanna et al.	
2005/0039104 A1	2/2005	Shah et al.	2009/0313698 A1	12/2009	Wahl	
2005/0063377 A1	3/2005	Bryant et al.	2009/0319912 A1	12/2009	Serr et al.	
2005/0083933 A1	4/2005	Fine et al.	2009/0328219 A1*	12/2009	Narayanawamy .....	H04L 63/1425 726/23
2005/0108331 A1	5/2005	Osterman	2010/0005288 A1	1/2010	Rao et al.	
2005/0122325 A1	6/2005	Twait	2010/0049839 A1	2/2010	Parker et al.	
2005/0138157 A1	6/2005	Jung et al.	2010/0054241 A1	3/2010	Shah et al.	
2005/0166066 A1	7/2005	Ahuja et al.	2010/0077445 A1	3/2010	Schneider et al.	
2005/0177829 A1	8/2005	Vishwanath	2010/0095293 A1	4/2010	O'Neill et al.	
2005/0182681 A1	8/2005	Bruskotter et al.				

(56)

References Cited

U.S. PATENT DOCUMENTS

2010/0095367	A1	4/2010	Narayanaswamy	2012/0254109	A1	10/2012	Shukla et al.
2010/0095377	A1	4/2010	Krywaniuk	2012/0260227	A1	10/2012	Shukla et al.
2010/0138526	A1	6/2010	DeHaan et al.	2012/0278021	A1	11/2012	Lin et al.
2010/0138810	A1	6/2010	Komatsu et al.	2012/0281700	A1	11/2012	Koganti et al.
2010/0148940	A1*	6/2010	Gelvin ..... H04L 67/12 340/286.02	2012/0300628	A1	11/2012	Prescott et al.
2010/0153316	A1	6/2010	Duffield et al.	2013/0003538	A1	1/2013	Greenberg et al.
2010/0153696	A1	6/2010	Beachem et al.	2013/0003733	A1	1/2013	Venkatesan et al.
2010/0180016	A1	7/2010	Bugwadia et al.	2013/0006935	A1	1/2013	Grisby
2010/0194741	A1	8/2010	Finocchio	2013/0007435	A1	1/2013	Bayani
2010/0220584	A1	9/2010	DeHaan et al.	2013/0038358	A1	2/2013	Cook et al.
2010/0235514	A1	9/2010	Beachem	2013/0041934	A1	2/2013	Annamalaisami et al.
2010/0235879	A1	9/2010	Burnside et al.	2013/0054682	A1	2/2013	Malik et al.
2010/0235915	A1	9/2010	Memon et al.	2013/0085889	A1	4/2013	Fitting et al.
2010/0287266	A1	11/2010	Asati et al.	2013/0086272	A1	4/2013	Chen et al.
2010/0303240	A1	12/2010	Beachem	2013/0103827	A1	4/2013	Dunlap et al.
2010/0306180	A1	12/2010	Johnson et al.	2013/0107709	A1	5/2013	Campbell et al.
2010/0317420	A1	12/2010	Hoffberg	2013/0124807	A1	5/2013	Nielsen et al.
2010/0319060	A1	12/2010	Aiken et al.	2013/0125107	A1	5/2013	Bandakka et al.
2011/0004935	A1	1/2011	Moffie et al.	2013/0145099	A1	6/2013	Liu et al.
2011/0010585	A1	1/2011	Bugenhagen et al.	2013/0148663	A1	6/2013	Xiong
2011/0022641	A1	1/2011	Werth et al.	2013/0159999	A1	6/2013	Chiueh et al.
2011/0055381	A1	3/2011	Narasimhan et al.	2013/0173784	A1	7/2013	Wang et al.
2011/0055388	A1	3/2011	Yumerefendi et al.	2013/0174256	A1	7/2013	Powers
2011/0066719	A1	3/2011	Mirymanov et al.	2013/0179487	A1	7/2013	Lubetzky et al.
2011/0069685	A1	3/2011	Tofighbakhsh	2013/0179879	A1	7/2013	Zhang et al.
2011/0072119	A1	3/2011	Bronstein et al.	2013/0198517	A1	8/2013	Mazzarella
2011/0083125	A1	4/2011	Komatsu et al.	2013/0198839	A1	8/2013	Wei et al.
2011/0085556	A1	4/2011	Breslin et al.	2013/0201986	A1	8/2013	Sajassi et al.
2011/0103259	A1	5/2011	Aybay et al.	2013/0205293	A1	8/2013	Levijarvi et al.
2011/0107074	A1	5/2011	Chan et al.	2013/0219161	A1	8/2013	Fontignie et al.
2011/0107331	A1	5/2011	Evans et al.	2013/0219500	A1	8/2013	Lukas et al.
2011/0126136	A1	5/2011	Abella et al.	2013/0232498	A1	9/2013	Mangtani et al.
2011/0126275	A1	5/2011	Anderson et al.	2013/0242999	A1	9/2013	Kamble et al.
2011/0145885	A1	6/2011	Rivers et al.	2013/0246925	A1	9/2013	Ahuja et al.
2011/0153039	A1	6/2011	Gvelesiani et al.	2013/0247201	A1	9/2013	Alperovitch et al.
2011/0153811	A1	6/2011	Jeong et al.	2013/0254879	A1	9/2013	Chesla et al.
2011/0158088	A1	6/2011	Lofstrand et al.	2013/0268994	A1	10/2013	Cooper et al.
2011/0170860	A1	7/2011	Smith et al.	2013/0275579	A1	10/2013	Hernandez et al.
2011/0173490	A1	7/2011	Narayanaswamy et al.	2013/0283374	A1	10/2013	Zisapel et al.
2011/0185423	A1	7/2011	Sallam	2013/0290521	A1	10/2013	Labovitz
2011/0196957	A1	8/2011	Ayachitula et al.	2013/0297771	A1	11/2013	Osterloh et al.
2011/0202655	A1	8/2011	Sharma et al.	2013/0301472	A1	11/2013	Allan
2011/0214174	A1	9/2011	Herzog et al.	2013/0304900	A1	11/2013	Trabelsi et al.
2011/0225207	A1	9/2011	Subramanian et al.	2013/0305369	A1	11/2013	Karta et al.
2011/0228696	A1	9/2011	Agarwal et al.	2013/0318357	A1	11/2013	Abraham et al.
2011/0238793	A1	9/2011	Bedare et al.	2013/0326623	A1	12/2013	Kruglick
2011/0246663	A1	10/2011	Melsen et al.	2013/0333029	A1	12/2013	Chesla et al.
2011/0277034	A1	11/2011	Hanson	2013/0336164	A1	12/2013	Yang et al.
2011/0283277	A1	11/2011	Castillo et al.	2013/0346736	A1	12/2013	Cook et al.
2011/0302652	A1	12/2011	Westerfeld	2013/0347103	A1	12/2013	Veteikis et al.
2011/0314148	A1	12/2011	Petersen et al.	2014/0006610	A1	1/2014	Formby et al.
2011/0317982	A1	12/2011	Xu et al.	2014/0006871	A1	1/2014	Lakshmanan et al.
2012/0005542	A1	1/2012	Petersen et al.	2014/0012814	A1	1/2014	Bercovici et al.
2012/0079592	A1	3/2012	Pandurangi	2014/0019972	A1	1/2014	Yahalom et al.
2012/0089664	A1	4/2012	Igelka	2014/0031005	A1	1/2014	Sumcad et al.
2012/0102361	A1	4/2012	Sass et al.	2014/0033193	A1	1/2014	Palaniappan
2012/0102543	A1	4/2012	Kohli et al.	2014/0036688	A1	2/2014	Stassinopoulos et al.
2012/0110188	A1	5/2012	Van Biljon et al.	2014/0040343	A1	2/2014	Nickolov Peter et al.
2012/0117226	A1	5/2012	Tanaka et al.	2014/0047185	A1	2/2014	Peterson et al.
2012/0117642	A1	5/2012	Lin et al.	2014/0047372	A1	2/2014	Gnezdov et al.
2012/0136996	A1	5/2012	Seo et al.	2014/0056318	A1	2/2014	Hansson et al.
2012/0137278	A1	5/2012	Draper et al.	2014/0059200	A1	2/2014	Nguyen et al.
2012/0137361	A1	5/2012	Yi et al.	2014/0074946	A1	3/2014	Dirstine et al.
2012/0140626	A1	6/2012	Anand et al.	2014/0089494	A1	3/2014	Dasari et al.
2012/0195198	A1	8/2012	Regan	2014/0092884	A1	4/2014	Murphy et al.
2012/0197856	A1	8/2012	Banka et al.	2014/0096058	A1	4/2014	Molesky et al.
2012/0198541	A1	8/2012	Reeves	2014/0105029	A1	4/2014	Jain et al.
2012/0216271	A1	8/2012	Cooper et al.	2014/0115219	A1	4/2014	Ajanovic et al.
2012/0218989	A1	8/2012	Tanabe et al.	2014/0129942	A1	5/2014	Rathod
2012/0219004	A1	8/2012	Balus et al.	2014/0137109	A1	5/2014	Sharma et al.
2012/0233348	A1	9/2012	Winters	2014/0140244	A1	5/2014	Kapadia et al.
2012/0233473	A1	9/2012	Vasseur et al.	2014/0143825	A1	5/2014	Behrendt et al.
2012/0240232	A1	9/2012	Azuma	2014/0149490	A1	5/2014	Luxenberg et al.
2012/0246303	A1	9/2012	Petersen et al.	2014/0156814	A1	6/2014	Barabash et al.
				2014/0156861	A1	6/2014	Cruz-Aguilar et al.
				2014/0164607	A1	6/2014	Bai et al.
				2014/0165200	A1	6/2014	Singla
				2014/0173623	A1	6/2014	Chang et al.
				2014/0192639	A1	7/2014	Smirnov

(56)

References Cited

U.S. PATENT DOCUMENTS

2014/0201717 A1	7/2014	Mascaro et al.	2016/0162308 A1	6/2016	Chen et al.
2014/0215573 A1	7/2014	Cepuran	2016/0162312 A1	6/2016	Doherty et al.
2014/0215621 A1	7/2014	Xaypanya et al.	2016/0173446 A1	6/2016	Nantel
2014/0224784 A1	8/2014	Kohler	2016/0173535 A1	6/2016	Barabash et al.
2014/0225603 A1	8/2014	Auguste et al.	2016/0183093 A1	6/2016	Vaughn et al.
2014/0233387 A1	8/2014	Zheng et al.	2016/0205002 A1	7/2016	Rieke et al.
2014/0269777 A1	9/2014	Rothstein et al.	2016/0216994 A1	7/2016	Sefidcon et al.
2014/0280499 A1	9/2014	Basavaiah et al.	2016/0217022 A1	7/2016	Velipasaoglu et al.
2014/0281030 A1	9/2014	Cui et al.	2016/0248794 A1*	8/2016	Cam ..... H04L 63/1433
2014/0286354 A1	9/2014	Van De Poel et al.	2016/0255082 A1	9/2016	Rathod
2014/0289854 A1	9/2014	Mahvi	2016/0269424 A1	9/2016	Chandola et al.
2014/0298461 A1	10/2014	Hohndel et al.	2016/0269442 A1	9/2016	Shieh
2014/0307686 A1	10/2014	Su et al.	2016/0269482 A1	9/2016	Jamjoom et al.
2014/0317278 A1	10/2014	Kersch et al.	2016/0294691 A1	10/2016	Joshi
2014/0317737 A1	10/2014	Shin et al.	2016/0308908 A1	10/2016	Kirby et al.
2014/0330616 A1	11/2014	Lyras	2016/0337204 A1	11/2016	Dubey et al.
2014/0331048 A1	11/2014	Casas-Sanchez et al.	2016/0357424 A1	12/2016	Pang et al.
2014/0331276 A1	11/2014	Frascadore et al.	2016/0357546 A1	12/2016	Chang et al.
2014/0331280 A1	11/2014	Porras et al.	2016/0357587 A1	12/2016	Yadav et al.
2014/0331304 A1	11/2014	Wong	2016/0357957 A1	12/2016	Deen et al.
2014/0348182 A1	11/2014	Chandra et al.	2016/0359592 A1	12/2016	Kulshreshtha et al.
2014/0351203 A1	11/2014	Kunnatur et al.	2016/0359628 A1	12/2016	Singh et al.
2014/0351415 A1	11/2014	Harrigan et al.	2016/0359658 A1	12/2016	Yadav et al.
2014/0359695 A1	12/2014	Chari et al.	2016/0359673 A1	12/2016	Gupta et al.
2015/0006689 A1	1/2015	Szilagy et al.	2016/0359677 A1	12/2016	Kulshreshtha et al.
2015/0006714 A1	1/2015	Jain	2016/0359678 A1	12/2016	Madani et al.
2015/0009840 A1	1/2015	Pruthi et al.	2016/0359679 A1	12/2016	Parandehgheibi et al.
2015/0033305 A1	1/2015	Shear et al.	2016/0359680 A1	12/2016	Parandehgheibi et al.
2015/0036480 A1	2/2015	Huang et al.	2016/0359686 A1	12/2016	Parandehgheibi et al.
2015/0036533 A1	2/2015	Sodhi et al.	2016/0359695 A1	12/2016	Yadav et al.
2015/0039751 A1	2/2015	Harrigan et al.	2016/0359696 A1	12/2016	Yadav et al.
2015/0046882 A1	2/2015	Menyhart et al.	2016/0359697 A1	12/2016	Scheib et al.
2015/0052441 A1	2/2015	Degioanni	2016/0359698 A1	12/2016	Deen et al.
2015/0058976 A1	2/2015	Carney et al.	2016/0359699 A1	12/2016	Gandham et al.
2015/0067143 A1	3/2015	Babakhan et al.	2016/0359700 A1	12/2016	Pang et al.
2015/0067786 A1	3/2015	Fiske	2016/0359701 A1	12/2016	Pang et al.
2015/0082151 A1	3/2015	Liang et al.	2016/0359703 A1	12/2016	Gandham et al.
2015/0085665 A1	3/2015	Kompella et al.	2016/0359704 A1	12/2016	Gandham et al.
2015/0095332 A1	4/2015	Beisiegel et al.	2016/0359705 A1	12/2016	Parandehgheibi et al.
2015/0112933 A1	4/2015	Satapathy	2016/0359708 A1	12/2016	Gandham et al.
2015/0113133 A1	4/2015	Srinivas et al.	2016/0359709 A1	12/2016	Deen et al.
2015/0124608 A1	5/2015	Agarwal et al.	2016/0359711 A1	12/2016	Deen et al.
2015/0124652 A1	5/2015	Dharmapurikar et al.	2016/0359712 A1	12/2016	Alizadeh Attar et al.
2015/0128205 A1	5/2015	Mahaffey et al.	2016/0359740 A1	12/2016	Parandehgheibi et al.
2015/0138993 A1	5/2015	Forster et al.	2016/0359759 A1	12/2016	Singh et al.
2015/0142962 A1	5/2015	Srinivas et al.	2016/0359872 A1	12/2016	Yadav et al.
2015/0195291 A1	7/2015	Zuk et al.	2016/0359877 A1	12/2016	Kulshreshtha et al.
2015/0222939 A1	8/2015	Gallant et al.	2016/0359878 A1	12/2016	Prasad et al.
2015/0249622 A1	9/2015	Phillips et al.	2016/0359879 A1	12/2016	Deen et al.
2015/0256555 A1	9/2015	Choi et al.	2016/0359880 A1	12/2016	Pang et al.
2015/0261842 A1	9/2015	Huang et al.	2016/0359881 A1	12/2016	Yadav et al.
2015/0261886 A1	9/2015	Wu et al.	2016/0359888 A1	12/2016	Gupta et al.
2015/0271008 A1	9/2015	Jain et al.	2016/0359889 A1	12/2016	Yadav et al.
2015/0271255 A1	9/2015	Mackay et al.	2016/0359890 A1	12/2016	Deen et al.
2015/0295945 A1	10/2015	Canzanese, Jr. et al.	2016/0359897 A1	12/2016	Yadav et al.
2015/0312233 A1	10/2015	Graham, III et al.	2016/0359905 A1	12/2016	Touboul et al.
2015/0347554 A1	12/2015	Vasantham et al.	2016/0359912 A1	12/2016	Gupta et al.
2015/0358352 A1	12/2015	Chasin et al.	2016/0359913 A1	12/2016	Gupta et al.
2016/0006753 A1	1/2016	McDaid et al.	2016/0359914 A1	12/2016	Deen et al.
2016/0019030 A1	1/2016	Shukla et al.	2016/0359915 A1	12/2016	Gupta et al.
2016/0020959 A1	1/2016	Rahaman	2016/0359917 A1	12/2016	Rao et al.
2016/0021131 A1	1/2016	Heilig	2016/0373481 A1	12/2016	Sultan et al.
2016/0026552 A1	1/2016	Holden et al.	2016/0380865 A1	12/2016	Dubai et al.
2016/0036636 A1	2/2016	Erickson et al.	2017/0006141 A1	1/2017	Bhadra
2016/0036837 A1	2/2016	Jain et al.	2017/0024453 A1	1/2017	Raja et al.
2016/0050132 A1	2/2016	Zhang et al.	2017/0032310 A1	2/2017	Mimnaugh
2016/0072815 A1	3/2016	Rieke et al.	2017/0034018 A1	2/2017	Parandehgheibi et al.
2016/0080414 A1	3/2016	Kolton et al.	2017/0048121 A1	2/2017	Hobbs et al.
2016/0087861 A1	3/2016	Kuan et al.	2017/0070582 A1	3/2017	Desai et al.
2016/0094394 A1	3/2016	Sharma et al.	2017/0085483 A1	3/2017	Mihaly et al.
2016/0103692 A1	4/2016	Guntaka et al.	2017/0208487 A1	7/2017	Ratakonda et al.
2016/0105350 A1	4/2016	Greifeneder et al.	2017/0250880 A1	8/2017	Akens et al.
2016/0112284 A1	4/2016	Pon et al.	2017/0250951 A1	8/2017	Wang et al.
2016/0119234 A1	4/2016	Valencia Lopez et al.	2017/0289067 A1	10/2017	Lu et al.
2016/0127395 A1	5/2016	Underwood et al.	2017/0295141 A1	10/2017	Thubert et al.
			2017/0302691 A1	10/2017	Singh et al.
			2017/0331747 A1	11/2017	Singh et al.
			2017/0346736 A1	11/2017	Chander et al.
			2017/0364380 A1	12/2017	Frye, Jr. et al.

(56)

## References Cited

## U.S. PATENT DOCUMENTS

2018/0006911	A1	1/2018	Dickey
2018/0007115	A1	1/2018	Nedelchev et al.
2018/0013670	A1	1/2018	Kapadia et al.
2018/0145906	A1	5/2018	Yadav et al.

## FOREIGN PATENT DOCUMENTS

CN	102521537	6/2012
CN	103023970	4/2013
CN	103716137	4/2014
CN	104065518	9/2014
CN	107196807	9/2017
EP	0811942	12/1997
EP	1076848	7/2002
EP	1 383 261	1/2004
EP	1450511	8/2004
EP	2045974	4/2008
EP	2043320	4/2009
EP	2860912	4/2015
EP	2887595	6/2015
JP	2009-016906	1/2009
KR	1394338	5/2014
WO	WO 2007/014314	2/2007
WO	WO 2007/070711	6/2007
WO	WO 2008/069439	6/2008
WO	2013030830	3/2013
WO	WO 2015/042171	3/2015
WO	WO 2016/004075	1/2016
WO	WO 2016/019523	2/2016

## OTHER PUBLICATIONS

Bosch, Greg, "Virtualization," 2010, 33 pages.

Breen, Christopher, "Mac 911, How to dismiss Mac App Store Notifications," Macworld.com, Mar. 24, 2014, 3 pages.

Chou, C.W., et al., "Optical Clocks and Relativity," *Science* vol. 329, Sep. 24, 2010, pp. 1630-1633.

Huang, Hing-Jie, et al., "Clock Skew Based Node Identification in Wireless Sensor Networks," *IEEE*, 2008, 5 pages.

Ives, Herbert, E., et al., "An Experimental Study of the Rate of a Moving Atomic Clock," *Journal of the Optical Society of America*, vol. 28, No. 7, Jul. 1938, pp. 215-226.

Witze, Alexandra, "Special relativity aces time trial, 'Time dilation' predicted by Einstein confirmed by lithium ion experiment," *Nature*, Sep. 19, 2014, 3 pages.

Zatrochova, Zuzana, "Analysis and Testing of Distributed NoSQL Datastore Riak," Spring, 2015, 76 pages.

Australian Government Department of Defence, Intelligence and Security, "Top 4 Strategies to Mitigate Targeted Cyber Intrusions," Cyber Security Operations Centre Jul. 2013, <http://www.asd.gov.au/infosec/top-mitigations/top-4-strategies-explained.htm>.

Author Unknown, "Blacklists & Dynamic Reputation: Understanding Why the Evolving Threat Eludes Blacklists," [www.dambala.com](http://www.dambala.com), 9 pages, Dambala, Atlanta, GA, USA.

Aydin, Galip, et al., "Architecture and Implementation of a Scalable Sensor Data Storage and Analysis Using Cloud Computing and Big Data Technologies," *Journal of Sensors*, vol. 2015, Article ID 834217, Feb. 2015, 11 pages.

Backes, Michael, et al., "Data Lineage in Malicious Environments," *IEEE* 2015, pp. 1-13.

Bayati, Mohsen, et al., "Message-Passing Algorithms for Sparse Network Alignment," Mar. 2013, 31 pages.

Berezinski, Przemyslaw, et al., "An Entropy-Based Network Anomaly Detection Method," *Entropy*, 2015, vol. 17, [www.mdpi.com/journal/entropy](http://www.mdpi.com/journal/entropy), pp. 2367-2408.

Berthier, Robin, et al., "Nfsight: Netflow-based Network Awareness Tool," 2010, 16 pages.

Bhuyan, Dhiraj, "Fighting Bots and Botnets," 2006, pp. 23-28.

Blair, Dana, et al., U.S. Appl. No. 62/106,006, filed Jan. 21, 2015, entitled "Monitoring Network Policy Compliance."

Chandran, Midhun, et al., "Monitoring in a Virtualized Environment," *GSTF International Journal on Computing*, vol. 1, No. 1, Aug. 2010.

Chari, Suresh, et al., "Ensuring continuous compliance through reconciling policy with usage," *Proceedings of the 18<sup>th</sup> ACM symposium on Access control models and technologies (SACMAT '13)*. ACM, New York, NY, USA, 49-60.

Chen, Xu, et al., "Automating network application dependency discovery: experiences, limitations, and new solutions," 8th USENIX conference on Operating systems design and implementation (OSDI'08), USENIX Association, Berkeley, CA, USA, 117-130.

Cisco Systems, "Cisco Network Analysis Modules (NAM) Tutorial," Cisco Systems, Inc., Version 3.5.

Cisco Systems, Inc., "Addressing Compliance from One Infrastructure: Cisco Unified Compliance Solution Framework," 2014.

Cisco Systems, Inc., "Cisco Application Dependency Mapping Service," 2009.

Cisco Systems, Inc., "White Paper—New Cisco Technologies Help Customers Achieve Regulatory Compliance," 1992-2008.

Cisco Systems, Inc., "A Cisco Guide to Defending Against Distributed Denial of Service Attacks," May 3, 2016, 34 pages.

Cisco Technology, Inc., "Cisco Lock-and-Key:Dynamic Access Lists," <http://www.cisco.com/c/en/us/support/docs/security/vpn/lock-key/7604-13.html>; Updated Jul. 12, 2006, 16 pages.

Di Lorenzo, Guisy, et al., "EXSED: An Intelligent Tool for Exploration of Social Events Dynamics from Augmented Trajectories," *Mobile Data Management (MDM)*, pp. 323-330, Jun. 3-6, 2013.

Feinstein, Laura, et al., "Statistical Approaches to DDoS Attack Detection and Response," *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX '03)*, Apr. 2003, 12 pages.

George, Ashley, et al., "NetPal: A Dynamic Network Administration Knowledge Base," 2008, pp. 1-14.

Goldstein, Abigail, et al., "A Tool for Monitoring and Maintaining System Trustworthiness at Run Time," *REFSQ* (2015), pp. 142-147.

Hamadi, S., et al., "Fast Path Acceleration for Open vSwitch in Overlay Networks," *Global Information Infrastructure and Networking Symposium (GIIS)*, Montreal, QC, pp. 1-5, Sep. 15-19, 2014.

Hewlett-Packard, "Effective use of reputation intelligence in a security operations center," Jul. 2013, 6 pages.

Hideshima, Yusuke, et al., "Starmine: A Visualization System for Cyber Attacks," <https://www.researchgate.net/publication/221536306>, Feb. 2006, 9 pages.

InternetPerils, Inc., "Control Your Internet Business Risk," 2003-2015, <http://www.internetperils.com>.

Janoff, Christian, et al., "Cisco Compliance Solution for HIPAA Security Rule Design and Implementation Guide," Cisco Systems, Inc., Updated Nov. 14, 2015, part 1 of 2, 350 pages.

Janoff, Christian, et al., "Cisco Compliance Solution for HIPAA Security Rule Design and Implementation Guide," Cisco Systems, Inc., Updated Nov. 14, 2015, part 2 of 2, 588 pages.

Kerrison, Adam, et al., "Four Steps to Faster, Better Application Dependency Mapping—Laying the Foundation for Effective Business Service Models," BMCSoftware, 2011.

Kraemer, Brian, "Get to know your data center with CMDB," *TechTarget*, Apr. 5, 2006, <http://searchdatacenter.techtarget.com/news/118820/Get-to-know-your-data-center-with-CMDB>.

Lab Sku, "VMware Hands-on Labs—HOL-SDC-1301" Version: 20140321-160709, 2013; [http://docs.hol.vmware.com/HOL-2013/hoisdc-1301\\_html\\_en/](http://docs.hol.vmware.com/HOL-2013/hoisdc-1301_html_en/) (part 1 of 2).

Lab Sku, "VMware Hands-on Labs—HOL-SDC-1301" Version: 20140321-160709, 2013; [http://docs.hol.vmware.com/HOL-2013/hoisdc-1301\\_html\\_en/](http://docs.hol.vmware.com/HOL-2013/hoisdc-1301_html_en/) (part 2 of 2).

Lachance, Michael, "Dirty Little Secrets of Application Dependency Mapping," Dec. 26, 2007.

Landman, Yoav, et al., "Dependency Analyzer," Feb. 14, 2008, <http://ifrog.com/confluence/display/DA/Home>.

Lee, Sihyung, "Reducing Complexity of Large-Scale Network Configuration Management," Ph.D. Dissertation, Carnegie Mellon University, 2010.

(56) **References Cited**

## OTHER PUBLICATIONS

- Li, Ang, et al., "Fast Anomaly Detection for Large Data Centers," Global Telecommunications Conference (GLOBECOM 2010, Dec. 2010, 6 pages.
- Li, Bingbong, et al., "A Supervised Machine Learning Approach to Classify Host Roles on Line Using sFlow," in Proceedings of the first edition workshop on High performance and programmable networking, 2013, ACM, New York, NY, USA, 53-60.
- Liu, Ting, et al., "Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems," In Proceedings of the Ninth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '03), ACM, New York, NY, USA, Jun. 11-13, 2003, pp. 107-118.
- Lu, Zhonghai, et al., "Cluster-based Simulated Annealing for Mapping Cores onto 2D Mesh Networks on Chip," Design and Diagnostics of Electronic Circuits and Systems, pp. 1, 6, 16-18, Apr. 2008.
- Matteson, Ryan, "Depmap: Dependency Mapping of Applications Using Operating System Events: a Thesis," Master's Thesis, California Polytechnic State University, Dec. 2010.
- Natarajan, Arun, et al., "NSDMiner: Automated Discovery of Network Service Dependencies," Institute of Electrical and Electronics Engineers INFOCOM, Feb. 2012, 9 pages.
- Navaz, A.S. Syed, et al., "Entropy based Anomaly Detection System to Prevent DDoS Attacks in Cloud," International Journal of computer Applications (0975-8887), vol. 62, No. 15, Jan. 2013, pp. 42-47.
- Neverfail, "Neverfail IT Continuity Architect," 2015, <https://web.archive.org/web/20150908090456/http://www.neverfailgroup.com/products/it-continuity-architect>.
- Nilsson, Dennis K., et al., "Key Management and Secure Software Updates in Wireless Process Control Environments," In Proceedings of the First ACM Conference on Wireless Network Security (WiSec '08), ACM, New York, NY, USA, Mar. 31-Apr. 2, 2008, pp. 100-108.
- Nunnally, Troy, et al., "P3D: A Parallel 3D Coordinate Visualization for Advanced Network Scans," IEEE 2013, Jun. 9-13, 2013, 6 pages.
- O'Donnell, Glenn, et al., "The CMDB Imperative: How to Realize the Dream and Avoid the Nightmares," Prentice Hall, Feb. 19, 2009.
- Ohta, Kohei, et al., "Detection, Defense, and Tracking of Internet-Wide Illegal Access in a Distributed Manner," 2000, pp. 1-16.
- Pathway Systems International Inc., "How Blueprints does Integration," Apr. 15, 2014, 9 pages, <http://pathwayssystems.com/company-blog/>.
- Pathway Systems International Inc., "What is Blueprints?" 2010-2016, <http://pathwayssystems.com/blueprints-about/>.
- Popa, Lucian, et al., "Macroscopic: End-Point Approach to Networked Application Dependency Discovery," CoNEXT'09, Dec. 1-4, 2009, Rome, Italy, 12 pages.
- Prasad, K. Munivara, et al., "An Efficient Detection of Flooding Attacks to Internet Threat Monitors (ITM) using Entropy Variations under Low Traffic," Computing Communication & Networking Technologies (ICCCNT '12), Jul. 26-28, 2012, 11 pages.
- Sachan, Mrinmaya, et al., "Solving Electrical Networks to incorporate Supervision in Random Walks," May 13-17, 2013, pp. 109-110.
- Sammarco, Matteo, et al., "Trace Selection for Improved WLAN Monitoring," Aug. 16, 2013, pp. 9-14.
- Shneiderman, Ben, et al., "Network Visualization by Semantic Substrates," Visualization and Computer Graphics, vol. 12, No. 5, pp. 733,740, Sep.-Oct. 2006.
- Wang, Ru, et al., "Learning directed acyclic graphs via bootstrap aggregating," 2014, 47 pages, <http://arxiv.org/abs/1406.2098>.
- Wang, Yongjun, et al., "A Network Gene-Based Framework for Detecting Advanced Persistent Threats," Nov. 2014, 7 pages.
- Woodberg, Brad, "Snippet from Juniper SRX Series" Jun. 17, 2013, 1 page, O'Reilly Media, Inc.
- Zhang, Yue, et al., "Cantina: A Content-Based Approach to Detecting Phishing Web Sites," May 8-12, 2007, pp. 639-648.
- Bauch, Petr, "Reader's Report of Master's Thesis, Analysis and Testing of Distributed NoSQL Datastore Riak," May 28, 2015, Brno. 2 pages.
- Kim, Myung-Sup, et al. "A Flow-based Method for Abnormal Network Traffic Detection," IEEE, 2004, pp. 599-612.
- Thomas, R., "Bogon Dotted Decimal List," Version 7.0, Team Cymru NOC, Apr. 27, 2012, 5 pages.
- Al-Fuqaha, Ala, et al., "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Communication Surveys & Tutorials. vol. 17, No. 4, Nov. 18, 2015, pp. 2347-2376.
- Brocade Communications Systems, Inc., "Chapter 5—Configuring Virtual LANs (VLANs)," Jun. 2009, 38 pages.
- Cisco Systems, Inc. "Cisco, Nexus 3000 Series NX-OS Release Notes, Release 5.0(3)U3(1)," Feb. 29, 2012, Part No. OL-26631-01, 16 pages.
- Cisco Systems, Inc., "Cisco—VPN Client User Guide for Windows," Release 4.6, Aug. 2004, 148 pages.
- Cisco Systems, Inc., "Cisco 4710 Application Control Engine Appliance Hardware Installation Guide," Nov. 2007, 66 pages.
- Cisco Systems, Inc., "Cisco Data Center Network Architecture and Solutions Overview," Feb. 2006, 19 pages.
- Cisco Systems, Inc., "Cisco IOS Configuration Fundamentals Configuration Guide: *Using Autoinstall and Setup*," Release 12.2, first published Apr. 2001, last updated Sep. 2003, 32 pages.
- Cisco Systems, Inc., "Cisco VN-Link: Virtualization-Aware Networking," White Paper, Mar. 2009, 10 pages.
- Cisco Systems, Inc., "Cisco, Nexus 5000 Series and Cisco Nexus 2000 Series Release Notes, Cisco NX-OS Release 5.1(3)N2(1b), NX-OS Release 5.1(3)N2(1a) and NX-OS Release 5.1(3)N2(1)," Sep. 5, 2012, Part No. OL-26652-03 CO, 24 pages.
- Cisco Systems, Inc., "Nexus 3000 Series NX-OS Fundamentals Configuration Guide, Release 5.0(3)U3(1): *Using PowerOn Auto Provisioning*," Feb. 29, 2012, Part No. OL-26544-01, 10 pages.
- Cisco Systems, Inc., "Quick Start Guide, Cisco ACE 4700 Series Application Control Engine Appliance," Software Ve740rson A5(1.0), Sep. 2011, 138 pages.
- Cisco Systems, Inc., "Routing and Bridging Guide, Cisco ACE Application Control Engine," Software Version A5(1.0), Sep. 2011, 248 pages.
- Cisco Systems, Inc., "VMWare and Cisco Virtualization Solution: Scale Virtual Machine Networking," Jul. 2009, 4 pages.
- Cisco Systems, Inc., "Cisco Remote Integrated Service Engine for Citrix NetScaler Appliances and Cisco Nexus 7000 Series Switches Configuration Guide," Last modified Apr. 29, 2014, 78 pages.
- Cisco Technology, Inc., "Cisco IOS Software Release 12.4T Features and Hardware Support," Feb. 2009, 174 pages.
- Cisco Systems, Inc., "Cisco Application Control Engine (ACE) Troubleshooting Guide—Understanding the ACE Module Architecture and Traffic Flow," Mar. 11, 2011, 6 pages.
- Costa, Raul, et al., "An Intelligent Alarm Management System for Large-Scale Telecommunication Companies," in Portuguese Conference on Artificial Intelligence, Oct. 2009, 14 pages.
- De Carvalho, Tiago Filipe Rodrigues, "Root Cause Analysis in Large and Complex Networks," Dec. 2008, Repositorio.ul.pt, pp. 1-55.
- Foundation for Intelligent Physical Agents, "FIPA Agent Message Transport Service Specification," Dec. 3, 2002, <http://www.fipa.org/>; 15 pages.
- Gia, Tuan Nguyen, et al., "Fog Computing in Healthcare Internet of Things: A Case Study on ECG Feature Extraction," 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Oct. 26, 2015, pp. 356-363.
- Joseph, Dilip, et al., "Modeling Middleboxes," IEEE Network, Sep./Oct. 2008, pp. 20-25.
- Kent, S., et al. "Security Architecture for the Internet Protocol," Network Working Group, Nov. 1998, 67 pages.
- Online Collins English Dictionary, 1 page. (Year: 2018).
- Voris, Jonathan, et al., "Bait and Snitch: Defending Computer Systems with Decoys," Columbia University Libraries, Department of Computer Science, 2013, pp. 1-25.

(56)

**References Cited**

OTHER PUBLICATIONS

Baek, Kwang-Hyun, et al., "Preventing Theft of Quality of Service on Open Platforms," 2005 Workshop of the 1<sup>st</sup> International Conference on Security and Privacy for Emerging Areas in Communication Networks, 2005, 12 pages.

Theodorakopoulos, George, et al., "On Trust Models and Trust Evaluation Metrics for Ad Hoc Networks," IEEE Journal on Selected Areas in Communications. vol. 24, Issue 2, Feb. 2006, pp. 318-328.

Zeng, Sai, et al., "Managing Risk in Multi-node Automation of Endpoint Management," 2014 IEEE Network Operations and Management Symposium (NOMS), 2014, 6 pages.

\* cited by examiner

FIG. 1

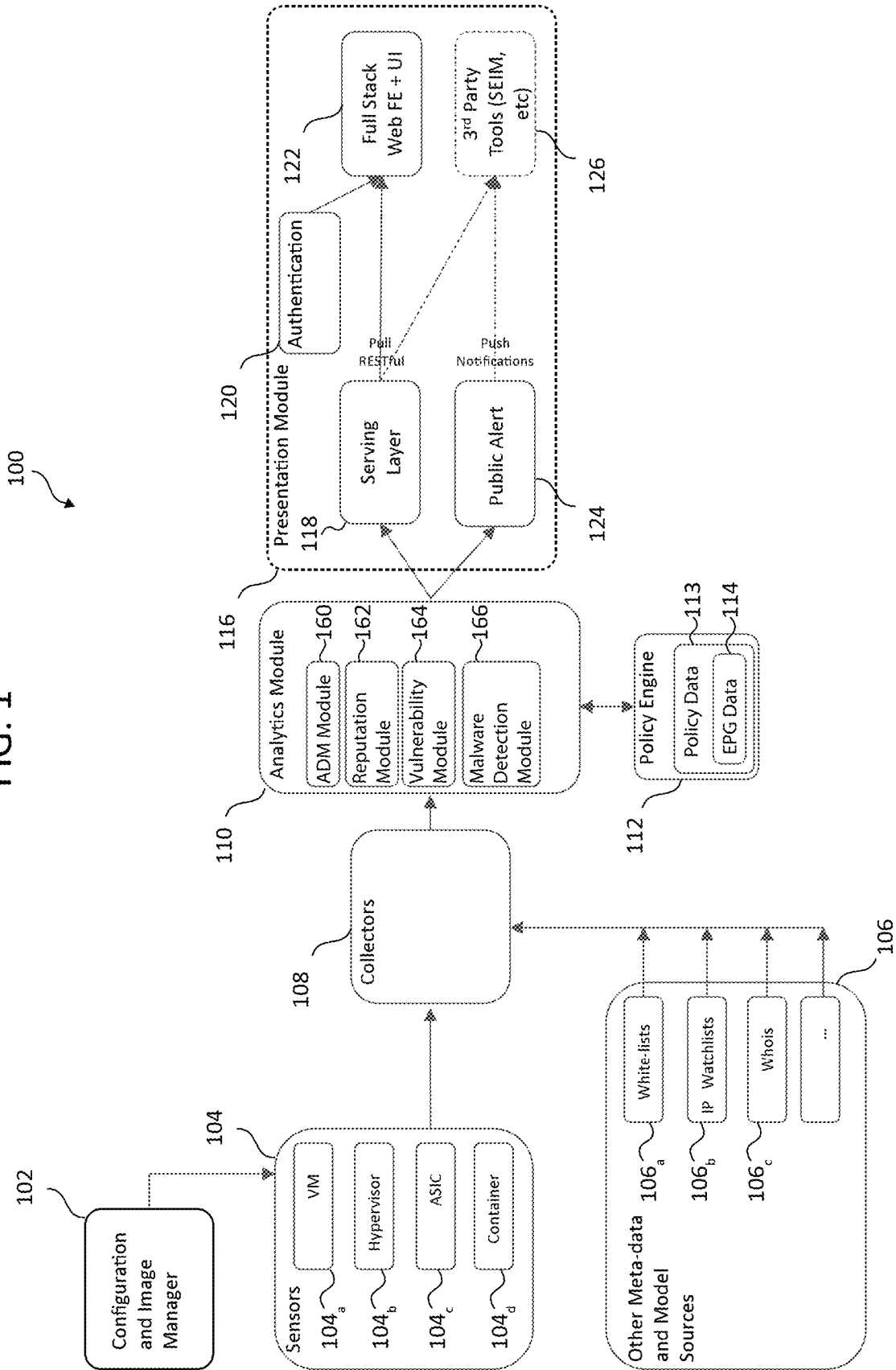
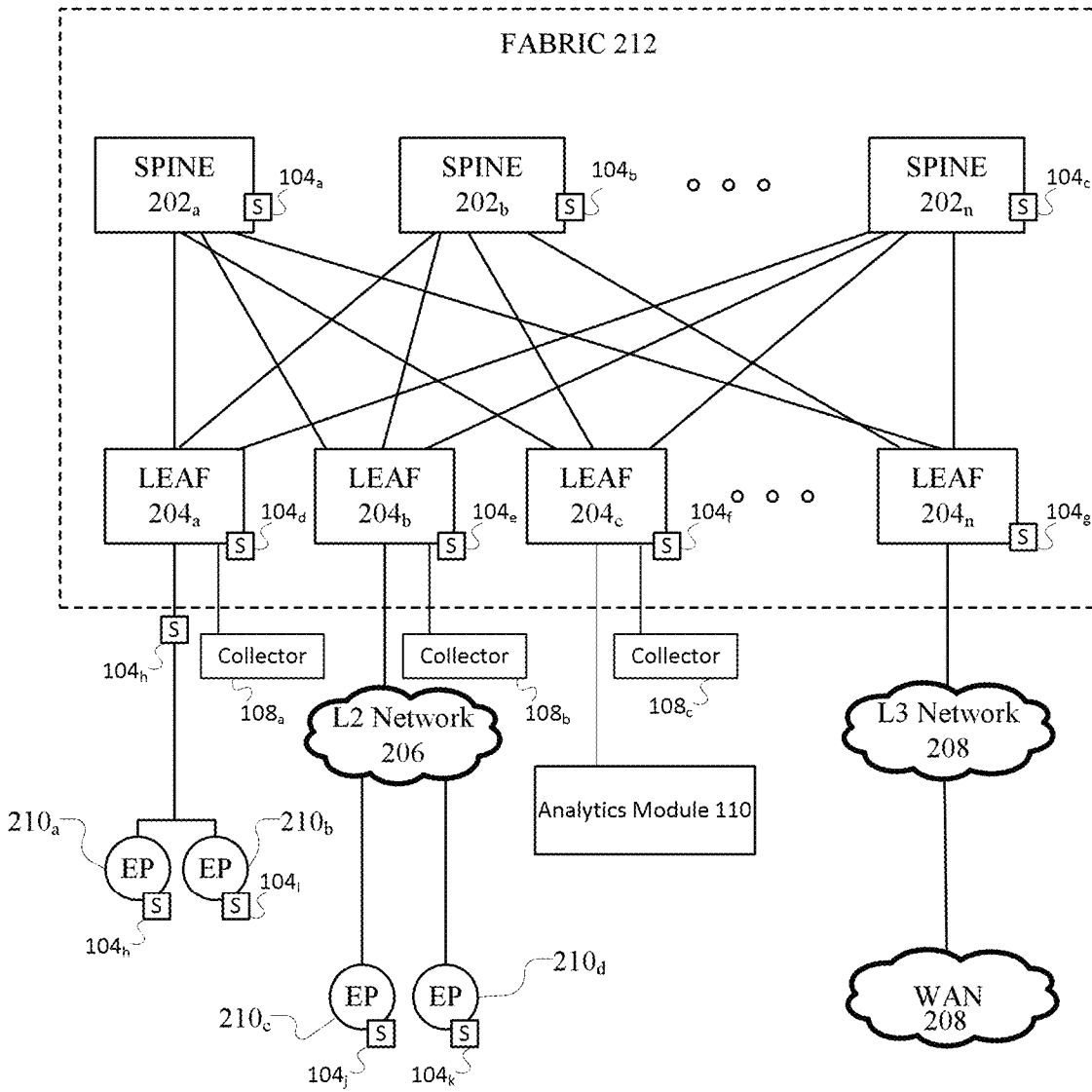


FIG. 2

200



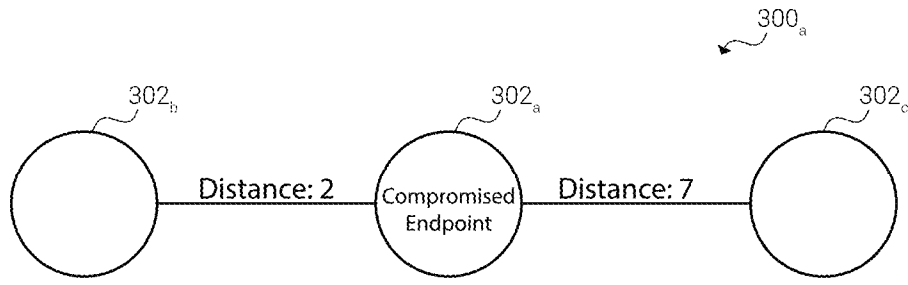


FIG. 3A

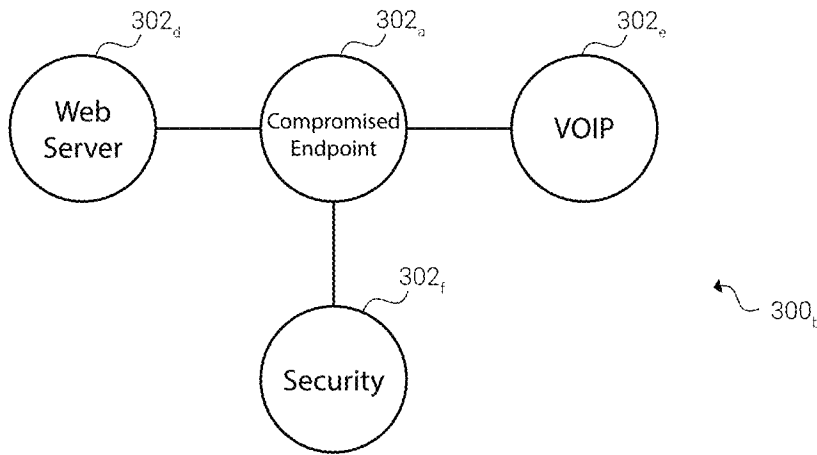


FIG. 3B

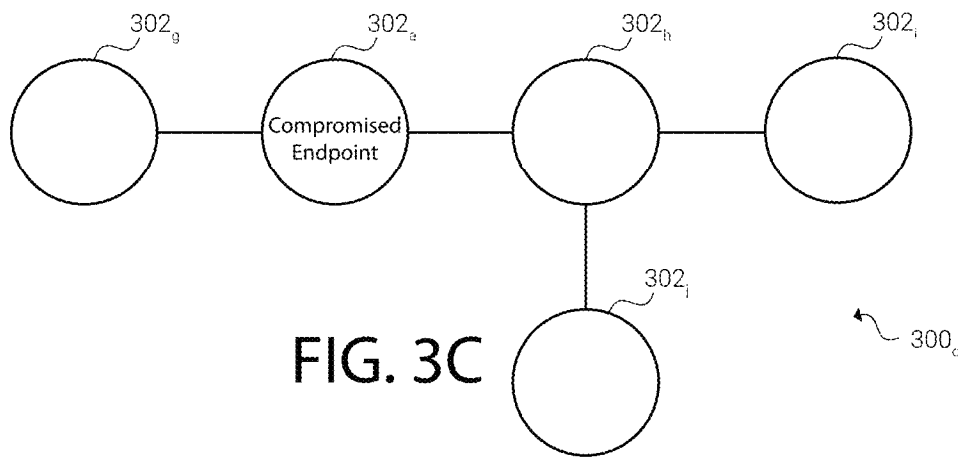


FIG. 3C

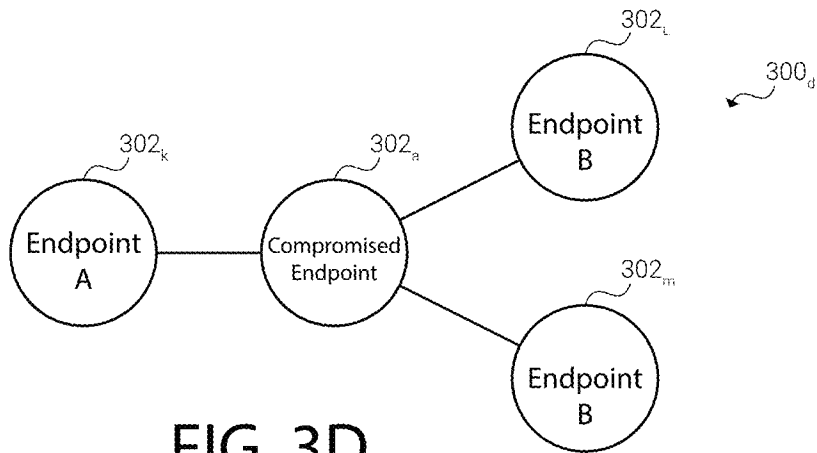


FIG. 3D

400

Application	Business Criticality Ranking	Secondary Value	Priority Ranking
Finance	1	1	1
Security	1	2	2
Human Resources	2		3

FIG. 4

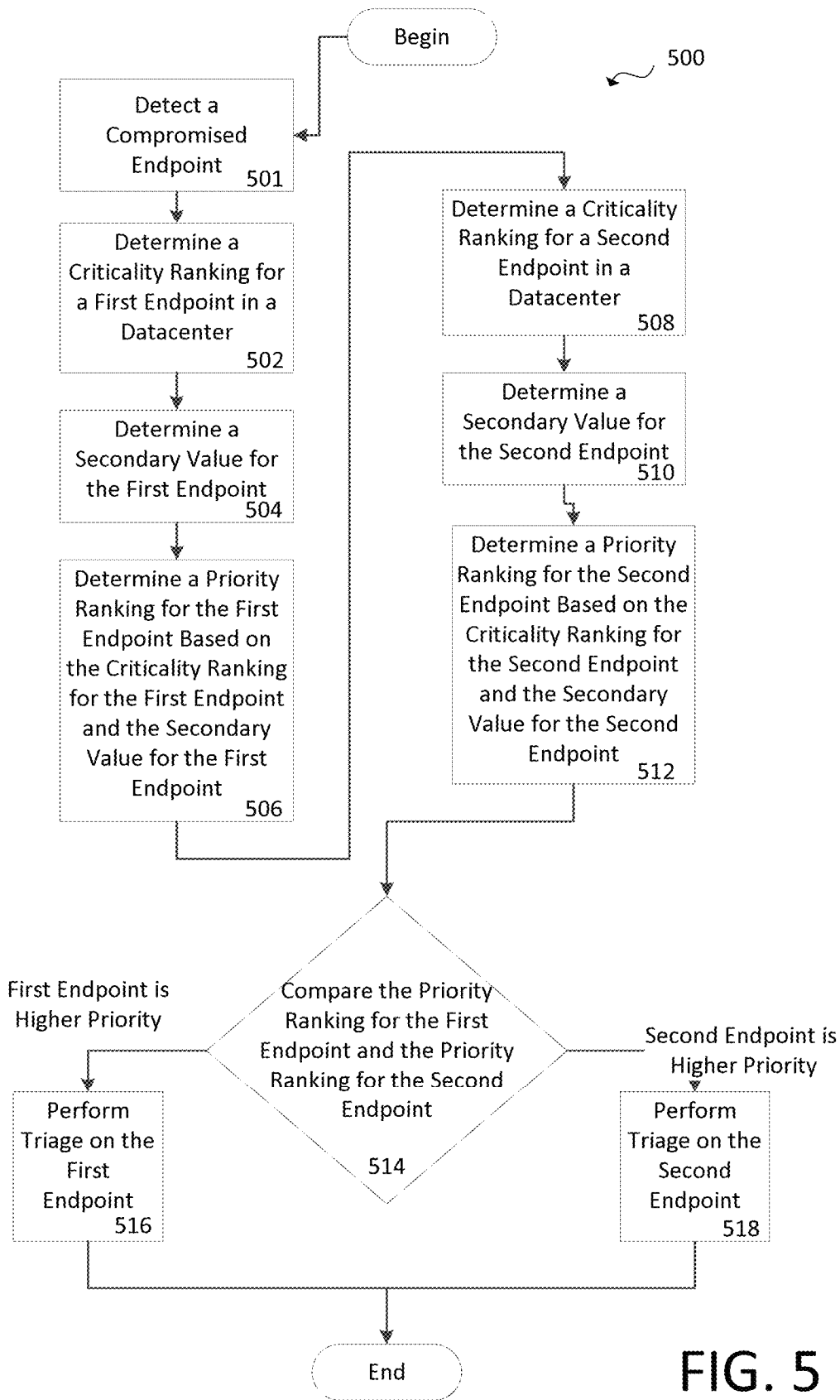


FIG. 5

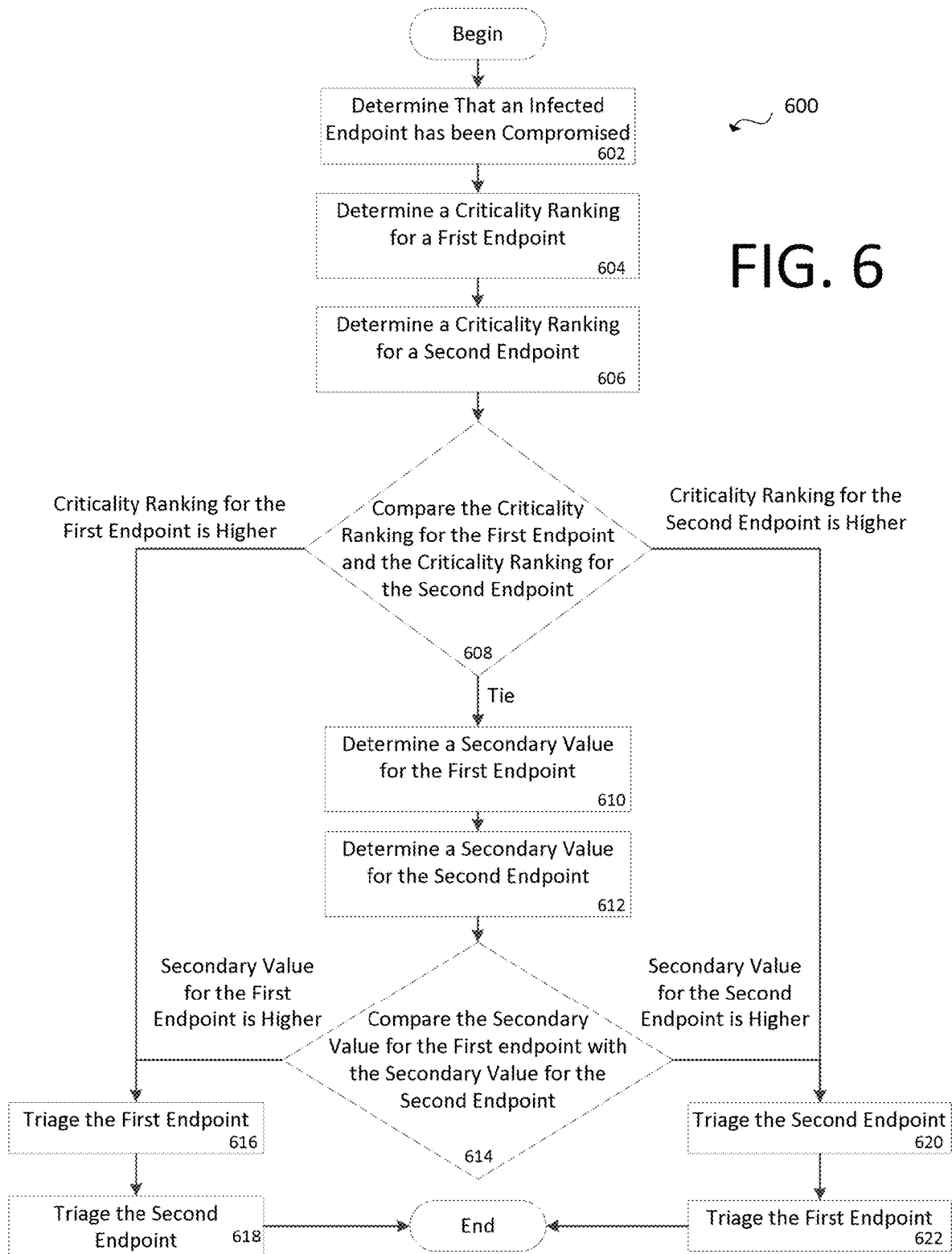


FIG. 6

FIG. 7A

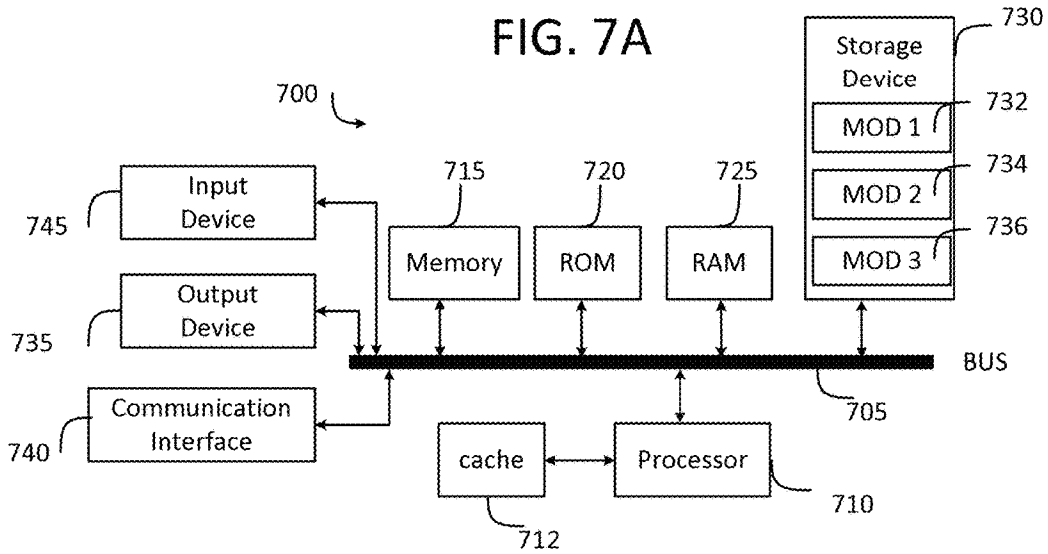
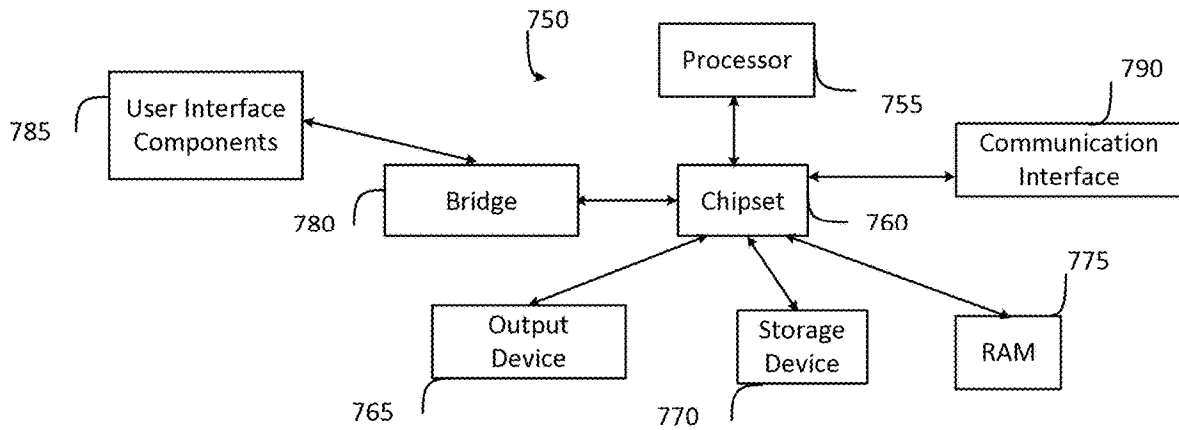


FIG. 7B



1

## APPLICATION MONITORING PRIORITIZATION

### RELATED APPLICATIONS

This application claims priority to U.S. Provisional Application No. 62/171,899, entitled "SYSTEM FOR MONITORING AND MANAGING DATACENTERS", filed Jun. 5, 2015, which is incorporated herein by reference in its entirety.

### TECHNICAL FIELD

The present technology pertains to network security and more specifically establishing a priority ranking for an endpoint.

### BACKGROUND

When an endpoint is compromised in a network, other endpoints may become compromised as well. It can be important to triage the other nodes and determine if they are also compromised or if they are at risk of being compromised in the future. In a datacenter, there can be a large number of endpoints and triaging each one can take a large amount of time. An endpoint at the tail end of the triage queue might become compromised while awaiting triage.

### BRIEF DESCRIPTION OF THE FIGURES

In order to describe the manner in which the above-recited and other advantages and features of the disclosure can be obtained, a more particular description of the principles briefly described above will be rendered by reference to specific embodiments that are illustrated in the appended drawings. Understanding that these drawings depict only example embodiments of the disclosure and are not therefore to be considered to be limiting of its scope, the principles herein are described and explained with additional specificity and detail through the use of the accompanying drawings in which:

FIG. 1 illustrates an example network traffic monitoring system according to some example embodiments;

FIG. 2 illustrates an example network environment according to some example embodiments;

FIG. 3A, FIG. 3B, FIG. 3C, and FIG. 3D illustrate example network configurations;

FIG. 4 illustrates an example table depicting example business criticality rankings, secondary values, and priority rankings of various example applications;

FIG. 5 illustrates an example method according to some embodiments;

FIG. 6 illustrates an example method according to some embodiments; and

FIGS. 7A and 7B illustrate example system embodiments.

### DESCRIPTION OF EXAMPLE EMBODIMENTS

#### Overview

An approach for establishing a priority ranking for endpoints in a network. This can be useful when triaging endpoints after an endpoint becomes compromised. Ensuring that the most critical and vulnerable endpoints are triaged first can help maintain network stability and mitigate damage to endpoints in the network after an endpoint is compromised. The present technology involves determining a criticality ranking and a secondary value for a first end-

2

point in a datacenter. The criticality ranking and secondary value can be combined to form priority ranking for the first endpoint which can then be compared to a priority ranking for a second endpoint to determine if the first endpoint or the second endpoint should be triaged first.

#### Detailed Description

Various embodiments of the disclosure are discussed in detail below. While specific implementations are discussed, it should be understood that this is done for illustration purposes only. A person skilled in the relevant art will recognize that other components and configurations may be used without parting from the spirit and scope of the disclosure.

The disclosed technology addresses the need in the art for determining a priority ranking for endpoints in a network.

FIG. 1 shows an example network traffic monitoring system **100** according to some example embodiments. Network traffic monitoring system **100** can include configuration and image manager **102**, sensors **104**, external data sources **106**, collectors **108**, analytics module **110**, policy engine **112**, and presentation module **116**. These modules may be implemented as hardware and/or software components. Although FIG. 1 illustrates an example configuration of the various components of network traffic monitoring system **100**, those of skill in the art will understand that the components of network traffic monitoring system **100** or any system described herein can be configured in a number of different ways and can include any other type and number of components. For example, sensors **104** and collectors **108** can belong to one hardware and/or software module or multiple separate modules. Other modules can also be combined into fewer components and/or further divided into more components.

Configuration and image manager **102** can provision and maintain sensors **104**. In some example embodiments, sensors **104** can reside within virtual machine images, and configuration and image manager **102** can be the component that also provisions virtual machine images.

Configuration and image manager **102** can configure and manage sensors **104**. When a new virtual machine (VM) is instantiated or when an existing VM is migrated, configuration and image manager **102** can provision and configure a new sensor on the physical server hosting the VM. In some example embodiments configuration and image manager **102** can monitor the health of sensors **104**. For instance, configuration and image manager **102** may request status updates or initiate tests. In some example embodiments, configuration and image manager **102** can also manage and provision the virtual machines themselves.

In some example embodiments, configuration and image manager **102** can verify and validate sensors **104**. For example, sensors **104** can be provisioned a unique ID that is created using a one-way hash function of its basic input/output system (BIOS) universally unique identifier (UUID) and a secret key stored on configuration and image manager **102**. This UUID can be a large number that is difficult for an imposter sensor to guess. In some example embodiments, configuration and image manager **102** can keep sensors **104** up to date by installing new versions of their software and applying patches. Configuration and image manager **102** can obtain these updates automatically from a local source or the Internet.

Sensors **104** can reside on nodes of a data center network (e.g., virtual partition, hypervisor, physical server, switch, router, gateway, other network device, other electronic device, etc.). In general, a virtual partition may be an instance of a virtual machine (VM) (e.g., VM **104a**), sand-

box, container (e.g., container **104c**), or any other isolated environment that can have software operating within it. The software may include an operating system and application software. For software running within a virtual partition, the virtual partition may appear to be a distinct physical server. In some example embodiments, a hypervisor (e.g., hypervisor **104b**) may be a native or “bare metal” hypervisor that runs directly on hardware, but that may alternatively run under host software executing on hardware. Sensors **104** can monitor communications to and from the nodes and report on environmental data related to the nodes (e.g., node IDs, statuses, etc.). Sensors **104** can send their records over a high-speed connection to collectors **108** for storage. Sensors **104** can comprise a piece of software (e.g., running on a VM, container, virtual switch, hypervisor, physical server, or other device), an application-specific integrated circuit (ASIC) (e.g., a component of a switch, gateway, router, standalone packet monitor, or other network device including a packet capture (PCAP) module or similar technology), or an independent unit (e.g., a device connected to a network device’s monitoring port or a device connected in series along a main trunk of a datacenter). It should be understood that various software and hardware configurations can be used as sensors **104**. Sensors **104** can be lightweight, thereby minimally impeding normal traffic and compute resources in a datacenter. Sensors **104** can “sniff” packets being sent over its host network interface card (NIC) or individual processes can be configured to report traffic to sensors **104**. This sensor structure allows for robust capture of granular (i.e., specific) network traffic data from each hop of data transmission.

As sensors **104** capture communications, they can continuously send network traffic and associated data to collectors **108**. The network traffic data can relate to a packet, a collection of packets, a flow, a group of flows, etc. The associated data can include details such as the VM BIOS ID, sensor ID, associated process ID, associated process name, process user name, sensor private key, geo-location of a sensor, environmental details, etc. The network traffic data can include information describing the communication on all layers of the Open Systems Interconnection (OSI) model. For example, the network traffic data can include signal strength (if applicable), source/destination media access control (MAC) address, source/destination internet protocol (IP) address, protocol, port number, encryption data, requesting process, a sample packet, etc.

In some example embodiments, sensors **104** can preprocess network traffic data before sending to collectors **108**. For example, sensors **104** can remove extraneous or duplicative data or they can create a summary of the data (e.g., latency, packets and bytes sent per flow, flagged abnormal activity, etc.). In some example embodiments, sensors **104** can be configured to only capture certain types of connection information and disregard the rest. Because it can be overwhelming for a system to capture every packet in a network, in some example embodiments, sensors **104** can be configured to capture only a representative sample of packets (e.g., every 1,000th packet or other suitable sample rate).

Sensors **104** can send network traffic data to one or multiple collectors **108**. In some example embodiments, sensors **104** can be assigned to a primary collector and a secondary collector. In other example embodiments, sensors **104** are not assigned a collector, but can determine an optimal collector through a discovery process. Sensors **104** can change where they send their network traffic data if their environments change, such as if a certain collector experiences failure or if a sensor is migrated to a new location and becomes closer to a different collector. In some example

embodiments, sensors **104** can send different types of network traffic data to different collectors. For example, sensors **104** can send network traffic data related to one type of process to one collector and network traffic data related to another type of process to another collector.

Collectors **108** can serve as a repository for the data recorded by sensors **104**. In some example embodiments, collectors **108** can be directly connected to a top of rack switch. In other example embodiments, collectors **108** can be located near an end of row switch. Collectors **108** can be located on or off premises. It will be appreciated that the placement of collectors **108** can be optimized according to various priorities such as network capacity, cost, and system responsiveness. In some example embodiments, data storage of collectors **108** is located in an in-memory database, such as dashDB by International Business Machines. This approach benefits from rapid random access speeds that typically are required for analytics software. Alternatively, collectors **108** can utilize solid state drives, disk drives, magnetic tape drives, or a combination of the foregoing according to cost, responsiveness, and size requirements. Collectors **108** can utilize various database structures such as a normalized relational database or NoSQL database.

In some example embodiments, collectors **108** may only serve as network storage for network traffic monitoring system **100**. In other example embodiments, collectors **108** can organize, summarize, and preprocess data. For example, collectors **108** can tabulate how often packets of certain sizes or types are transmitted from different nodes of a data center. Collectors **108** can also characterize the traffic flows going to and from various nodes. In some example embodiments, collectors **108** can match packets based on sequence numbers, thus identifying traffic flows and connection links. In some example embodiments, collectors **108** can flag anomalous data. Because it would be inefficient to retain all data indefinitely, in some example embodiments, collectors **108** can periodically replace detailed network traffic flow data and associated data (host data, process data, user data, etc.) with consolidated summaries. In this manner, collectors **108** can retain a complete dataset describing one period (e.g., the past minute or other suitable period of time), with a smaller dataset of another period (e.g., the previous 2-10 minutes or other suitable period of time), and progressively consolidate network traffic flow data and associated data of other periods of time (e.g., day, week, month, year, etc.). By organizing, summarizing, and preprocessing the network traffic flow data and associated data, collectors **108** can help network traffic monitoring system **100** scale efficiently. Although collectors **108** are generally referred to herein in the plurality, it will be appreciated that collectors **108** can be implemented using a single machine, especially for smaller datacenters.

In some example embodiments, collectors **108** can receive data from external data sources **106**, such as security reports, white-lists (**106a**), IP watchlists (**106b**), whois data (**106c**), or out-of-band data, such as power status, temperature readings, etc.

In some example embodiments, network traffic monitoring system **100** can include a wide bandwidth connection between collectors **108** and analytics module **110**. Analytics module **110** can include application dependency (ADM) module **160**, reputation module **162**, vulnerability module **164**, malware detection module **166**, etc., to accomplish various tasks with respect to the flow data and associated data collected by sensors **104** and stored in collectors **108**. In some example embodiments, network traffic monitoring system **100** can automatically determine network topology.

Using network traffic flow data and associated data captured by sensors **104**, network traffic monitoring system **100** can determine the type of devices existing in the network (e.g., brand and model of switches, gateways, machines, etc.), physical locations (e.g., latitude and longitude, building, datacenter, room, row, rack, machine, etc.), interconnection type (e.g., 10 Gb Ethernet, fiber-optic, etc.), and network characteristics (e.g., bandwidth, latency, etc.). Automatically determining the network topology can assist with integration of network traffic monitoring system **100** within an already established datacenter. Furthermore, analytics module **110** can detect changes of network topology without the need of further configuration.

Analytics module **110** can determine dependencies of components within the network using ADM module **160**. For example, if component A routinely sends data to component B but component B never sends data to component A, then analytics module **110** can determine that component B is dependent on component A, but A is likely not dependent on component B. If, however, component B also sends data to component A, then they are likely interdependent. These components can be processes, virtual machines, hypervisors, virtual local area networks (VLANs), etc. Once analytics module **110** has determined component dependencies, it can then form a component (“application”) dependency map. This map can be instructive when analytics module **110** attempts to determine a root cause of a failure (because failure of one component can cascade and cause failure of its dependent components). This map can also assist analytics module **110** when attempting to predict what will happen if a component is taken offline. Additionally, analytics module **110** can associate edges of an application dependency map with expected latency, bandwidth, etc. for that individual edge.

Analytics module **110** can establish patterns and norms for component behavior. For example, it can determine that certain processes (when functioning normally) will only send a certain amount of traffic to a certain VM using a small set of ports. Analytics module can establish these norms by analyzing individual components or by analyzing data coming from similar components (e.g., VMs with similar configurations). Similarly, analytics module **110** can determine expectations for network operations. For example, it can determine the expected latency between two components, the expected throughput of a component, response times of a component, typical packet sizes, traffic flow signatures, etc. In some example embodiments, analytics module **110** can combine its dependency map with pattern analysis to create reaction expectations. For example, if traffic increases with one component, other components may predictably increase traffic in response (or latency, compute time, etc.).

In some example embodiments, analytics module **110** can use machine learning techniques to identify security threats to a network using malware detection module **166**. For example, malware detection module **166** can be provided with examples of network states corresponding to an attack and network states corresponding to normal operation. Malware detection module **166** can then analyze network traffic flow data and associated data to recognize when the network is under attack. In some example embodiments, the network can operate within a trusted environment for a time so that analytics module **110** can establish baseline normalcy. In some example embodiments, analytics module **110** can contain a database of norms and expectations for various components. This database can incorporate data from sources external to the network (e.g., external sources **106**). Analytics module **110** can then create access policies for

how components can interact using policy engine **112**. In some example embodiments, policies can be established external to network traffic monitoring system **100** and policy engine **112** can detect the policies and incorporate them into analytics module **110**. A network administrator can manually tweak the policies. Policies can dynamically change and be conditional on events. These policies can be enforced by the components depending on a network control scheme implemented by a network. Policy engine **112** can maintain these policies and receive user input to change the policies.

Policy engine **112** can configure analytics module **110** to establish or maintain network policies. For example, policy engine **112** may specify that certain machines should not intercommunicate or that certain ports are restricted. A network and security policy controller (not shown) can set the parameters of policy engine **112**. In some example embodiments, policy engine **112** can be accessible via presentation module **116**. In some example embodiments, policy engine **112** can include policy data **112**. In some example embodiments, policy data **112** can include endpoint group (EPG) data **114**, which can include the mapping of EPGs to IP addresses and/or MAC addresses. In some example embodiments, policy data **112** can include policies for handling data packets.

In some example embodiments, analytics module **110** can simulate changes in the network. For example, analytics module **110** can simulate what may result if a machine is taken offline, if a connection is severed, or if a new policy is implemented. This type of simulation can provide a network administrator with greater information on what policies to implement. In some example embodiments, the simulation may serve as a feedback loop for policies. For example, there can be a policy that if certain policies would affect certain services (as predicted by the simulation) those policies should not be implemented. Analytics module **110** can use simulations to discover vulnerabilities in the datacenter. In some example embodiments, analytics module **110** can determine which services and components will be affected by a change in policy. Analytics module **110** can then take necessary actions to prepare those services and components for the change. For example, it can send a notification to administrators of those services and components, it can initiate a migration of the components, it can shut the components down, etc.

In some example embodiments, analytics module **110** can supplement its analysis by initiating synthetic traffic flows and synthetic attacks on the datacenter. These artificial actions can assist analytics module **110** in gathering data to enhance its model. In some example embodiments, these synthetic flows and synthetic attacks are used to verify the integrity of sensors **104**, collectors **108**, and analytics module **110**. Over time, components may occasionally exhibit anomalous behavior. Analytics module **110** can analyze the frequency and severity of the anomalous behavior to determine a reputation score for the component using reputation module **162**. Analytics module **110** can use the reputation score of a component to selectively enforce policies. For example, if a component has a high reputation score, the component may be assigned a more permissive policy or more permissive policies; while if the component frequently violates (or attempts to violate) its relevant policy or policies, its reputation score may be lowered and the component may be subject to a stricter policy or stricter policies. Reputation module **162** can correlate observed reputation score with characteristics of a component. For example, a particular virtual machine with a particular configuration may be more prone to misconfiguration and receive a lower

reputation score. When a new component is placed in the network, analytics module **110** can assign a starting reputation score similar to the scores of similarly configured components. The expected reputation score for a given component configuration can be sourced outside of the datacenter. A network administrator can be presented with expected reputation scores for various components before installation, thus assisting the network administrator in choosing components and configurations that will result in high reputation scores.

Some anomalous behavior can be indicative of a misconfigured component or a malicious attack. Certain attacks may be easy to detect if they originate outside of the datacenter, but can prove difficult to detect and isolate if they originate from within the datacenter. One such attack could be a distributed denial of service (DDOS) where a component or group of components attempt to overwhelm another component with spurious transmissions and requests. Detecting an attack or other anomalous network traffic can be accomplished by comparing the expected network conditions with actual network conditions. For example, if a traffic flow varies from its historical signature (packet size, transport control protocol header options, etc.) it may be an attack.

In some cases, a traffic flow and associated data may be expected to be reported by a sensor, but the sensor may fail to report it. This situation could be an indication that the sensor has failed or become compromised. By comparing the network traffic flow data and associated data from multiple sensors **104** spread throughout the datacenter, analytics module **110** can determine if a certain sensor is failing to report a particular traffic flow.

Presentation module **116** can include serving layer **118**, authentication module **120**, web front end **122**, public alert module **124**, and third party tools **126**. In some example embodiments, presentation module **116** can provide an external interface for network monitoring system **100**. Using presentation module **116**, a network administrator, external software, etc. can receive data pertaining to network monitoring system **100** via a webpage, application programming interface (API), audiovisual queues, etc. In some example embodiments, presentation module **116** can preprocess and/or summarize data for external presentation. In some example embodiments, presentation module **116** can generate a webpage. As analytics module **110** processes network traffic flow data and associated data and generates analytic data, the analytic data may not be in a human-readable form or it may be too large for an administrator to navigate. Presentation module **116** can take the analytic data generated by analytics module **110** and further summarize, filter, and organize the analytic data as well as create intuitive presentations of the analytic data.

Serving layer **118** can be the interface between presentation module **116** and analytics module **110**. As analytics module **110** generates reports, predictions, and conclusions, serving layer **118** can summarize, filter, and organize the information that comes from analytics module **110**. In some example embodiments, serving layer **118** can also request raw data from a sensor or collector.

Web frontend **122** can connect with serving layer **118** to present the data from serving layer **118** in a webpage. For example, web frontend **122** can present the data in bar charts, core charts, tree maps, acyclic dependency maps, line graphs, tables, etc. Web frontend **122** can be configured to allow a user to “drill down” on information sets to get a filtered data representation specific to the item the user wishes to drill down to. For example, individual traffic

flows, components, etc. Web frontend **122** can also be configured to allow a user to filter by search. This search filter can use natural language processing to analyze the user’s input. There can be options to view data relative to the current second, minute, hour, day, etc. Web frontend **122** can allow a network administrator to view traffic flows, application dependency maps, network topology, etc.

In some example embodiments, web frontend **122** may be solely configured to present information. In other example embodiments, web frontend **122** can receive inputs from a network administrator to configure network traffic monitoring system **100** or components of the datacenter. These instructions can be passed through serving layer **118** to be sent to configuration and image manager **102** or policy engine **112**. Authentication module **120** can verify the identity and privileges of users. In some example embodiments, authentication module **120** can grant network administrators different rights from other users according to established policies.

Public alert module **124** can identify network conditions that satisfy specified criteria and push alerts to third party tools **126**. Public alert module **124** can use analytic data generated or accessible through analytics module **110**. One example of third party tools **126** is a security information and event management system (SIEM). Third party tools **126** may retrieve information from serving layer **118** through an API and present the information according to the SIEM’s user interfaces.

FIG. 2 illustrates an example network environment **200** according to some example embodiments. It should be understood that, for the network environment **100** and any environment discussed herein, there can be additional or fewer nodes, devices, links, networks, or components in similar or alternative configurations. Example embodiments with different numbers and/or types of clients, networks, nodes, cloud components, servers, software components, devices, virtual or physical resources, configurations, topologies, services, appliances, deployments, or network devices are also contemplated herein. Further, network environment **200** can include any number or type of resources, which can be accessed and utilized by clients or tenants. The illustrations and examples provided herein are for clarity and simplicity.

Network environment **200** can include network fabric **212**, layer 2 (L2) network **206**, layer 3 (L3) network **208**, endpoints **210a**, **210b**, . . . , and **210d** (collectively, “**204**”). Network fabric **212** can include spine switches **202a**, **202b**, . . . , **202n** (collectively, “**202**”) connected to leaf switches **204a**, **204b**, **204c**, . . . , **204n** (collectively, “**204**”). Spine switches **202** can connect to leaf switches **204** in network fabric **212**. Leaf switches **204** can include access ports (or non-fabric ports) and fabric ports. Fabric ports can provide uplinks to spine switches **202**, while access ports can provide connectivity for devices, hosts, endpoints, VMs, or other electronic devices (e.g., endpoints **204**), internal networks (e.g., L2 network **206**), or external networks (e.g., L3 network **208**).

Leaf switches **204** can reside at the edge of network fabric **212**, and can thus represent the physical network edge. In some cases, leaf switches **204** can be top-of-rack switches configured according to a top-of-rack architecture. In other cases, leaf switches **204** can be aggregation switches in any particular topology, such as end-of-row or middle-of-row topologies. Leaf switches **204** can also represent aggregation switches, for example.

Network connectivity in network fabric **212** can flow through leaf switches **204**. Here, leaf switches **204** can

provide servers, resources, VMs, or other electronic devices (e.g., endpoints **210**), internal networks (e.g., L2 network **206**), or external networks (e.g., L3 network **208**), access to network fabric **212**, and can connect leaf switches **204** to each other. In some example embodiments, leaf switches **204** can connect endpoint groups (EPGs) to network fabric **212**, internal networks (e.g., L2 network **206**), and/or any external networks (e.g., L3 network **208**). EPGs can be used in network environment **200** for mapping applications to the network. In particular, EPGs can use a grouping of application endpoints in the network to apply connectivity and policy to the group of applications. EPGs can act as a container for buckets or collections of applications, or application components, and tiers for implementing forwarding and policy logic. EPGs also allow separation of network policy, security, and forwarding from addressing by instead using logical application boundaries. For example, each EPG can connect to network fabric **212** via leaf switches **204**.

Endpoints **210** can connect to network fabric **212** via leaf switches **204**. For example, endpoints **210a** and **210b** can connect directly to leaf switch **204a**, which can connect endpoints **210a** and **210b** to network fabric **212** and/or any other one of leaf switches **204**. Endpoints **210c** and **210d** can connect to leaf switch **204b** via L2 network **206**. Endpoints **210c** and **210d** and L2 network **206** are examples of LANs. LANs can connect nodes over dedicated private communications links located in the same general physical location, such as a building or campus.

Wide area network (WAN) **212** can connect to leaf switches **204c** or **204d** via L3 network **208**. WANs can connect geographically dispersed nodes over long-distance communications links, such as common carrier telephone lines, optical lightpaths, synchronous optical networks (SONET), or synchronous digital hierarchy (SDH) links. LANs and WANs can include layer 2 (L2) and/or layer 3 (L3) networks and endpoints.

The Internet is an example of a WAN that connects disparate networks throughout the world, providing global communication between nodes on various networks. The nodes typically communicate over the network by exchanging discrete frames or packets of data according to predefined protocols, such as the Transmission Control Protocol/Internet Protocol (TCP/IP). In this context, a protocol can refer to a set of rules defining how the nodes interact with each other. Computer networks may be further interconnected by an intermediate network node, such as a router, to extend the effective size of each network. Endpoints **210** can include any communication device or component, such as a computer, server, hypervisor, virtual machine, container, process (e.g., running on a virtual machine), switch, router, gateway, host, device, external network, etc. In some example embodiments, endpoints **210** can include a server, hypervisor, process, or switch configured with virtual tunnel endpoint (VTEP) functionality which connects an overlay network with network fabric **212**. The overlay network may allow virtual networks to be created and layered over a physical network infrastructure. Overlay network protocols, such as Virtual Extensible LAN (VXLAN), Network Virtualization using Generic Routing Encapsulation (NVGRE), Network Virtualization Overlays (NVO3), and Stateless Transport Tunneling (STT), can provide a traffic encapsulation scheme which allows network traffic to be carried across L2 and L3 networks over a logical tunnel. Such logical tunnels can be originated and terminated through VTEPs. The overlay network can host physical devices, such as servers, applications, endpoint groups, virtual seg-

ments, virtual workloads, etc. In addition, endpoints **210** can host virtual workload(s), clusters, and applications or services, which can connect with network fabric **212** or any other device or network, including an internal or external network. For example, endpoints **210** can host, or connect to, a cluster of load balancers or an EPG of various applications.

Network environment **200** can also integrate a network traffic monitoring system, such as the one shown in FIG. 1. For example, as shown in FIG. 2, the network traffic monitoring system can include sensors **104a**, **104b**, . . . , **104n** (collectively, “**104**”), collectors **108a**, **108b**, . . . **108n** (collectively, “**108**”), and analytics module **110**. In some example embodiments, spine switches **202** do not have sensors **104**. Analytics module **110** can receive and process network traffic and associated data collected by collectors **108** and detected by sensors **104** placed on nodes located throughout network environment **200**. In some example embodiments, analytics module **110** can be implemented in an active-standby model to ensure high availability, with a first analytics module functioning in a primary role and a second analytics module functioning in a secondary role. If the first analytics module fails, the second analytics module can take over control. Although analytics module **110** is shown to be a standalone network appliance in FIG. 2, it will be appreciated that analytics module **110** can also be implemented as a VM image that can be distributed onto a VM, a cluster of VMs, a software as a service (SaaS), or other suitable distribution model in various other example embodiments. In some example embodiments, sensors **104** can run on endpoints **210**, leaf switches **204**, spine switches **202**, in-between network elements (e.g., sensor **104h**), etc. In some example embodiments, leaf switches **204** can each have an associated collector **108**. For example, if leaf switch **204** is a top of rack switch then each rack can contain an assigned collector **108**.

Although network fabric **212** is illustrated and described herein as an example leaf-spine architecture, one of ordinary skill in the art will readily recognize that the subject technology can be implemented based on any network topology, including any data center or cloud network fabric. Indeed, other architectures, designs, infrastructures, and variations are contemplated herein. For example, the principles disclosed herein are applicable to topologies including three-tier (including core, aggregation, and access levels), fat tree, mesh, bus, hub and spoke, etc. It should be understood that sensors and collectors can be placed throughout the network as appropriate according to various architectures.

FIG. 3A, FIG. 3B, FIG. 3C, and FIG. 3D represent example network configurations of network environment **200**. Various endpoints **302<sub>a</sub>-302<sub>m</sub>** (collectively or individually, “endpoint **302**”) can run services within the network. Endpoint **302** can be similar to endpoint **210**. Endpoint **302** can be associated with an application (e.g., mail server, web server, security application, voice over IP, storage host, etc.). Endpoint **302** can be a network switch, router, firewall, etc. Endpoint **302** can comprise a virtual machine, bare metal hardware, container, etc. Endpoint **302** can run on a virtual machine, bare metal hardware, container, etc.

In FIGS. 3A-3D, compromised endpoint **302<sub>a</sub>** can represent an endpoint **302** that has been compromised or misconfigured. For example, a sensor **104** associated with endpoint **302<sub>a</sub>** might have stopped reporting, reported irregular traffic or behavior, or otherwise indicated that endpoint **302<sub>a</sub>** is compromised. Compromised endpoint **302<sub>a</sub>** might be running a virus, worm, other unauthorized programs, misconfigured programs, etc. Network monitoring

system 100 can identify compromised endpoint 302 using the principles herein disclosed.

When an endpoint 302 becomes compromised, there is a risk that it was compromised from another endpoint 302 on network 300, that it has or will compromise other endpoints 302, or the entity which compromised it might compromise other endpoints 302. It can be useful to perform triage with other endpoints 302 to determine their risk to the same causes that compromised endpoint 302<sub>a</sub> as well as implement defensive and remedial procedures. Triage, as used herein can entail ascertaining the present state of the endpoint 302, determining actions to be performed, and performing those actions. For example, network monitoring system 100 can block communications to another endpoint 302 that has a likelihood of becoming compromised, including blocking communications from compromised endpoint 302<sub>a</sub> to a vulnerable endpoint 302. Because not all endpoints 302 can be triaged at once, a prioritization scheme can be used to queue up analyzing and protecting endpoints 302. For example, endpoints 302 can be assigned a priority ranking and can be analyzed in according to their priority ranking. The priority ranking can be established using a variety of criteria such as distance, critically (e.g., business criticality), network connectivity, redundancy, vulnerability, similarity to compromised endpoint 302<sub>a</sub>, etc. A combination of criteria can also be utilized. It should be understood that the connections depicted in FIGS. 3A-3D can represent direct connections or indirect connections (e.g., those that require an intermediary).

In FIG. 3A, endpoint 302<sub>b</sub> is a distance of 2 away from compromised endpoint 302<sub>a</sub> while endpoint 302<sub>c</sub> is a distance of 7 away from compromised endpoint 302<sub>a</sub>. Distance can refer to the latency between two endpoints 302, the bandwidth between two endpoints 302, the number of hops in a path connecting two endpoints 302, the geographical distance between two endpoints 302, the redundancy in the connections between the two endpoints 302, etc. Distance can be calculated using any combination of the foregoing.

Network monitoring system 100 can determine the distances between two endpoints 302. For example, sensor 104 on an endpoint 302 can ping another endpoint 302 and, based on the response, can determine the latency between the two endpoints 302. Other techniques are contemplated for determining distances between endpoints 302. In some embodiments, a lower distance can result in a higher priority ranking for an endpoint 302. An endpoint 302 with a higher priority ranking can be triaged before an endpoint 302 with a lower priority ranking. It should be understood that “higher” and “lower” rankings and values as used herein can mean of greater importance or lesser priority as appropriate.

In FIG. 3B, compromised endpoint 302<sub>a</sub> is connected to web server endpoint 302<sub>d</sub>, voice over internet protocol (VOIP) endpoint 302<sub>e</sub>, and security endpoint 302<sub>f</sub>. Other endpoints associated with other applications are contemplated that can be run on an endpoint 302. For example, an endpoint can run an application for data storage, telecommunications, closed circuit television, data processing, finance, point-of-sale terminals, tech-support, video on demand, etc. In some embodiments, certain applications are especially critical to the business that owns or uses them. Endpoints 302 that are business critical can be those that would cause serious damage to the particular business should they have any problems. For example, a social network may consider web server endpoint 302<sub>d</sub> to be critical for the business whereas a telecommunications provider may consider VOIP endpoint 302<sub>e</sub> to be more critical to their business than web server endpoint 302<sub>d</sub>.

In some embodiments, business criticality can be provided by an administrator. For example, an administrator can indicate a business criticality ranking for a variety of endpoint classifications. Endpoints can be classified similar to the foregoing (e.g., “telecommunications”, “data storage”, etc.). Additionally or alternatively, business criticality can be determined based on an analysis of network 300 provided by network monitoring system 100. For example, network monitoring system 100 can determine that many communications and interactions depend on a classification of endpoint 302. Similarly, network monitoring system 100 can create an application dependency map which can inform criticality rankings.

An endpoint 302 with a higher business criticality ranking can be prioritized over other endpoints 302. For example, the business criticality ranking can inform a priority ranking.

In FIG. 3C, compromised endpoint 302<sub>a</sub> is connected directly to endpoint 302<sub>g</sub> and 302<sub>h</sub>. Endpoint 302<sub>h</sub> is connected to endpoint 302<sub>i</sub> and endpoint 302<sub>j</sub>. If an endpoint 302 serves as a hub for interconnecting multiple endpoints 302 it can have a higher priority ranking. For example, because endpoint 302<sub>h</sub> is connected to endpoints 302<sub>a</sub>, 302<sub>i</sub>, and 302<sub>j</sub>, it can have a higher priority ranking in comparison to endpoint 302<sub>g</sub> which is only connected to endpoint 302<sub>a</sub>. The more endpoints 302 that an endpoint 302 is connected to, the greater its priority ranking can be. In some embodiments, the priority ranking of an endpoint 302 is only increased based on the number of directly connected endpoints 302 it has; alternatively, the priority ranking can increase based on the number of indirectly connected endpoints 302. The priority ranking can be higher based on a distance-weighting of the number of connected endpoints 302, the distance being calculated as discussed above. The priority ranking can be higher based on the number of endpoints 302 that an endpoint 302 “protects” from compromised endpoint 302<sub>a</sub>. Protecting an endpoint can mean that communications from a protected endpoint 302 must go through this endpoint if they are to reach compromised endpoint 302<sub>a</sub>.

In FIG. 3D, compromised endpoint 402<sub>a</sub> is connected to Endpoint A 302<sub>k</sub>, Endpoint B 302<sub>L</sub>, and Endpoint B 302<sub>m</sub>. Endpoint B can be redundantly provided on endpoints 302<sub>L</sub> and 302<sub>m</sub>. Redundancy can contribute to a decrease in the priority ranking of an endpoint 302 while a lack of redundancy can contribute to an increase in the priority ranking of an endpoint 302. Thus, in example network 300<sub>a</sub>, Endpoint A 302<sub>k</sub> can have a higher priority ranking. Redundancy can mean simultaneous operation where both redundant endpoints 302 are active. Redundancy can mean where one endpoint 302 is a backup of another endpoint 302 in case one endpoint 302 suffers a failure. Redundancy can mean how recent a backup has been made of an endpoint 302. For example, an endpoint 302 that was recently backed up can have a higher redundancy than an endpoint 302 that was backed up a long time ago.

FIG. 4 shows example table 400 showing example business criticality rankings, secondary values, and priority rankings of various example applications (e.g., endpoints 302). A system (e.g., networking monitoring system 100) can use a table, database, or any other data structure similar to table 400 in order to determine priority rankings for endpoints. For example, the priority ranking can be a combination (such as a summation, weighted summation, average, maximum, etc.) of the business criticality ranking and the secondary value. The secondary value can be another factor (e.g., distance, redundancy, vulnerability, etc.) or a combination of factors. For example, a ranking or value

for distance can be averaged with a ranking or value for redundancy for the application for the secondary value.

In some embodiments, a system doing triage on a network can attempt to identify and mitigate vulnerabilities. It can begin by assigning business criticality rankings to endpoints **302**. In some embodiments, this might result in two endpoints **302** having the same criticality ranking. Arbitrarily deciding which endpoint **302** to triage first is possible; however applying an extra calculation to determine an ordering of the “tied” endpoints can be better. The system can then look to secondary values to tie-break. In table **400** for example, both “finance” and “security” have the same business criticality ranking, but “finance” can be considered of greater priority after considering secondary values (a value of 1 whereas “security” has a secondary value of 2).

In some embodiments, secondary values includes business criticality ranking and, instead of business criticality ranking being the primary consideration, another metric can be utilize (e.g., distance). For example, a system can determine the distance of an endpoint **302** from compromised endpoint **302<sub>a</sub>** and use business criticality ranking as a tie-breaking secondary value.

FIG. **5** shows an example method **500** according to some embodiments. A system (e.g., network monitoring system **100**) performing example method **500** can begin and detect a compromised endpoint (step **501**). Compromised endpoint **302<sub>a</sub>** can be an endpoint **302** that is running unauthorized code (e.g., a virus, trojan, worm, script, etc.), an endpoint **302** that is misconfigured, an endpoint **302** that is not authorized to be on the network, an endpoint **302** that is associated with a malicious entity (e.g., a user that has been labelled as malicious), an endpoint **302** that has been disconnected, or an endpoint **302** that otherwise is not performing optimally. Detecting compromised endpoint **302<sub>a</sub>** can include analyzing flow data from various sensors **104** including a sensor **104** associated with compromised endpoint **302<sub>a</sub>**.

After compromised endpoint **302<sub>a</sub>** is detected, other endpoints **302** can be triaged to determine whether they have problems or might soon have problems associated with compromised endpoint **302<sub>a</sub>**. For example, a virus that is installed on compromised endpoint **302<sub>a</sub>** might spread to connected endpoints **302**. Another example is that a vulnerability on compromised endpoint **302<sub>a</sub>** (that caused it to be compromised) might be present on other endpoints **302**. Timeliness in triage can be important because systems that rely on compromised endpoint **302<sub>a</sub>** might crash or have problems as a result of compromised endpoint **302<sub>a</sub>** not behaving regularly. Timeliness can also be important because the unwanted software installed on compromised endpoint **302<sub>a</sub>** might quickly spread throughout the datacenter, it can be important to determine if other endpoints **302** are compromised as well.

The system can continue and determine a criticality ranking for a first endpoint in a datacenter (step **502**). Step **502** can include a network administrator labelling the first endpoint with a criticality ranking. A label can be assigned to the first endpoint (e.g., “telecommunications”) and a ranking can be derived from that label. In some embodiments, this includes referring to a prioritization list of labels. Criticality can be specific to the business of the datacenter. For example, an internet provider can have telecommunications endpoints receive a higher criticality ranking. As should be evident in this description, the term “ranking” does not necessarily require exclusivity; i.e., multiple endpoints **302** can receive identical rankings.

The system can then determine a secondary value for the first endpoint (step **504**). This can include determining values (or rankings) associated with various criteria (e.g., distance, redundancy, vulnerability, etc.) and then combining multiple values (if there are multiple). Combining can include creating an average, a weighted average, a summation, etc.

The system can then determine a priority ranking for the first endpoint based on the criticality ranking for the first endpoint and the secondary value for the first endpoint (step **506**). This can be generated by combining the criticality ranking and secondary value. For example, the criticality ranking can be an integer component while and the secondary value can be a decimal component of the priority ranking. The priority ranking can be a value (e.g., 9.5), a position in a queue, a relative ordering of endpoints (e.g., the first endpoint has a higher priority than a second endpoint), a group of endpoints (e.g., a first group of endpoints can be triaged first, followed by a second group), etc. In some embodiments, the priority ranking is designed to be exclusive or nearly exclusive to avoid “ties” where two endpoints would have the same priority ranking.

In some embodiments, machine learning can be utilized to inform any of the criticality ranking, the secondary value, and the priority ranking. For example, a system can monitor scenarios where compromised node **302<sub>a</sub>** becomes compromised and then test various priority rankings and attempt to minimize problems for other endpoints **302** by varying the priority ranking. In some embodiments, the system can run simulations of compromised endpoint **302<sub>a</sub>** becoming compromised and have the machine learning program learn based on the simulations.

The system can then determine a criticality ranking for a second endpoint in a datacenter (step **508**). It can then determine a secondary value for the second endpoint (step **510**). It can then determine a priority ranking for the second endpoint based on the criticality ranking for the second endpoint and the secondary value for the second endpoint (step **512**). Steps **508**, **510**, and **512** can be similar to steps **502**, **504**, and **506**, respectively but for the second endpoint **302**.

The system can then compare the priority ranking for the first endpoint and the priority ranking for the second endpoint (step **514**). In some embodiments, the endpoint **302** with the higher priority ranking is triaged first.

In some embodiments, the system performing example method **500** can determine a criticality ranking for the first endpoint (step **502**) and the second endpoint (step **508**). After determining the respective criticality rankings, the system can determine that they are identical. In order to “break the tie”, the system can then analyze the respective secondary values (steps **504** and **508**). For example, it can determine the respective distances, and perform triage on the endpoint **302** that is fewer hops away from compromised node **302<sub>a</sub>**. Depending on how the secondary values are calculated, it is contemplated that there can be a tie even after comparing secondary values. The system can then determine tertiary values of the respective endpoints. Tertiary values can be calculated using some of the criteria not analyzed when calculating secondary values.

If the first endpoint has a higher priority, the system can perform triage on the first endpoint (step **516**). If the second endpoint has a higher priority, the system can perform triage on the second endpoint (step **518**). The endpoint **302** that is not triaged at first (in steps **516** or **518**) can be triaged later.

For example, the system can perform triage on the second endpoint in step 518 and then perform triage on the first endpoint.

Triage can mean creating a backup of the endpoint 302, adding redundancy to the endpoint 302 (e.g., duplicating the endpoint 302), scanning the endpoint 302 for viruses, having an administrator review the endpoint 302, applying stricter security settings for the endpoint 302, limiting the traffic to the endpoint 302, retrieving data from endpoint 302, retrieving data from a sensor 104 associated with the endpoint 302, analyzing data from a sensor 104 associated with the endpoint 302, analyzing the endpoint 302 for vulnerabilities (especially the vulnerabilities that compromised node 302<sub>a</sub>), changing an associated endpoint group for the endpoint 302, shutting down the endpoint 302, moving the endpoint 302 (e.g., if the endpoint 302 is a virtual machine or container, migrating it to another machine), etc.

FIG. 6 represents an example method 600 according to some embodiments. The example method can be performed by a system such as traffic monitoring system 100. The system can begin and determine that an infected endpoint has been compromised (step 602). For example, it can determine that an endpoint is misconfigured, hacked, insecure, running malicious code, etc. The system can then determine a criticality ranking for a first endpoint (step 604). It can then determine a criticality ranking for a second endpoint (step 606). The criticality rankings can be according to business criticality, that is, how much the business that runs the endpoints is dependent on these endpoints.

The system can then compare the criticality ranking for the first endpoint and the criticality ranking for the second endpoint (step 608). If there is a tie, meaning that the criticality rankings are the same or substantially the same. The system can determine a secondary value for the first endpoint (step 610). The system can then determine a secondary value for the second endpoint (step 612). The secondary value can be any combination of: an endpoint's distance to the compromised endpoint, the endpoint's similarity to the compromised endpoint (including vulnerability similarities), the endpoint's redundancy, etc.

The system can then compare the secondary value for the first endpoint with the secondary value for the second endpoint (step 614). In some embodiments, the secondary values are calculated in order to prevent a tie. If there is a tie, tertiary values (based on possible secondary criteria that were not used to determine the secondary criteria) can be determined.

If the criticality ranking for the first endpoint is higher (at step 608) or if the secondary value for the first endpoint is higher (at step 614), the system can triage the first endpoint (step 616). The system can then triage the second endpoint (step 618).

If the criticality ranking for the second endpoint is higher (at step 608) or if the secondary value for the second endpoint is higher (at step 614), the system can triage the second endpoint (step 620). The system can then triage the first endpoint (step 622).

Any of the steps in example method 600 can be accomplished with the assistance of sensors installed within the related datacenter, including the infected endpoint, the first endpoint, and the second endpoint.

FIG. 7A and FIG. 7B illustrate example system embodiments. The more appropriate embodiment will be apparent to those of ordinary skill in the art when practicing the present technology. Persons of ordinary skill in the art will also readily appreciate that other system embodiments are possible.

FIG. 7A illustrates a conventional system bus computing system architecture 700 wherein the components of the system are in electrical communication with each other using a bus 705. Example system 700 includes a processing unit (CPU or processor) 710 and a system bus 705 that couples various system components including the system memory 715, such as read only memory (ROM) 770 and random access memory (RAM) 775, to the processor 710. The system 700 can include a cache of high-speed memory connected directly with, in close proximity to, or integrated as part of the processor 710. The system 700 can copy data from the memory 715 and/or the storage device 730 to the cache 712 for quick access by the processor 710. In this way, the cache can provide a performance boost that avoids processor 710 delays while waiting for data. These and other modules can control or be configured to control the processor 710 to perform various actions. Other system memory 715 may be available for use as well. The memory 715 can include multiple different types of memory with different performance characteristics. The processor 710 can include any general purpose processor and a hardware module or software module, such as module 1 737, module 7 734, and module 3 736 stored in storage device 730, configured to control the processor 910 as well as a special-purpose processor where software instructions are incorporated into the actual processor design. The processor 710 may essentially be a completely self-contained computing system, containing multiple cores or processors, a bus, memory controller, cache, etc. A multi-core processor may be symmetric or asymmetric.

To enable user interaction with the computing device 700, an input device 745 can represent any number of input mechanisms, such as a microphone for speech, a touch-sensitive screen for gesture or graphical input, keyboard, mouse, motion input, speech and so forth. An output device 735 can also be one or more of a number of output mechanisms known to those of skill in the art. In some instances, multimodal systems can enable a user to provide multiple types of input to communicate with the computing device 700. The communications interface 740 can generally govern and manage the user input and system output. There is no restriction on operating on any particular hardware arrangement and therefore the basic features here may easily be substituted for improved hardware or firmware arrangements as they are developed.

Storage device 730 is a non-volatile memory and can be a hard disk or other types of computer readable media which can store data that are accessible by a computer, such as magnetic cassettes, flash memory cards, solid state memory devices, digital versatile disks, cartridges, random access memories (RAMs) 775, read only memory (ROM) 770, and hybrids thereof.

The storage device 730 can include software modules 737, 734, 736 for controlling the processor 710. Other hardware or software modules are contemplated. The storage device 730 can be connected to the system bus 705. In one aspect, a hardware module that performs a particular function can include the software component stored in a computer-readable medium in connection with the necessary hardware components, such as the processor 710, bus 705, display 735, and so forth, to carry out the function.

FIG. 7B illustrates an example computer system 750 having a chipset architecture that can be used in executing the described method and generating and displaying a graphical user interface (GUI). Computer system 750 is an example of computer hardware, software, and firmware that can be used to implement the disclosed technology. System

750 can include a processor 755, representative of any number of physically and/or logically distinct resources capable of executing software, firmware, and hardware configured to perform identified computations. Processor 755 can communicate with a chipset 760 that can control input to and output from processor 755. In this example, chipset 760 outputs information to output 765, such as a display, and can read and write information to storage device 770, which can include magnetic media, and solid state media, for example. Chipset 760 can also read data from and write data to RAM 775. A bridge 780 for interfacing with a variety of user interface components 785 can be provided for interfacing with chipset 760. Such user interface components 785 can include a keyboard, a microphone, touch detection and processing circuitry, a pointing device, such as a mouse, and so on. In general, inputs to system 750 can come from any of a variety of sources, machine generated and/or human generated.

Chipset 760 can also interface with one or more communication interfaces 790 that can have different physical interfaces. Such communication interfaces can include interfaces for wired and wireless local area networks, for broadband wireless networks, as well as personal area networks. Some applications of the methods for generating, displaying, and using the GUI disclosed herein can include receiving ordered datasets over the physical interface or be generated by the machine itself by processor 755 analyzing data stored in storage 770 or 775. Further, the machine can receive inputs from a user via user interface components 785 and execute appropriate functions, such as browsing functions by interpreting these inputs using processor 755.

It can be appreciated that example systems 700 and 750 can have more than one processor 710 or be part of a group or cluster of computing devices networked together to provide greater processing capability.

For clarity of explanation, in some instances the present technology may be presented as including individual functional blocks including functional blocks comprising devices, device components, steps or routines in a method embodied in software, or combinations of hardware and software.

In some embodiments the computer-readable storage devices, mediums, and memories can include a cable or wireless signal containing a bit stream and the like. However, when mentioned, non-transitory computer-readable storage media expressly exclude media such as energy, carrier signals, electromagnetic waves, and signals per se.

Methods according to the above-described examples can be implemented using computer-executable instructions that are stored or otherwise available from computer readable media. Such instructions can comprise, for example, instructions and data which cause or otherwise configure a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. Portions of computer resources used can be accessible over a network. The computer executable instructions may be, for example, binaries, intermediate format instructions such as assembly language, firmware, or source code. Examples of computer-readable media that may be used to store instructions, information used, and/or information created during methods according to described examples include magnetic or optical disks, flash memory, USB devices provided with non-volatile memory, networked storage devices, and so on.

Devices implementing methods according to these disclosures can comprise hardware, firmware and/or software, and can take any of a variety of form factors. Typical

examples of such form factors include laptops, smart phones, small form factor personal computers, personal digital assistants, rackmount devices, standalone devices, and so on. Functionality described herein also can be embodied in peripherals or add-in cards. Such functionality can also be implemented on a circuit board among different chips or different processes executing in a single device, by way of further example.

The instructions, media for conveying such instructions, computing resources for executing them, and other structures for supporting such computing resources are means for providing the functions described in these disclosures.

Although a variety of examples and other information was used to explain aspects within the scope of the appended claims, no limitation of the claims should be implied based on particular features or arrangements in such examples, as one of ordinary skill would be able to use these examples to derive a wide variety of implementations. Further and although some subject matter may have been described in language specific to examples of structural features and/or method steps, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to these described features or acts. For example, such functionality can be distributed differently or performed in components other than those identified herein. Rather, the described features and steps are disclosed as examples of components of systems and methods within the scope of the appended claims. Moreover, claim language reciting "at least one of" a set indicates that one member of the set or multiple members of the set satisfy the claim.

The invention claimed is:

1. A computer-implemented method comprising:

determining a first criticality ranking for a first endpoint in a datacenter;

determining a second criticality ranking for a second endpoint; and

when the first criticality ranking and the second criticality ranking are determined to be a same criticality ranking, executing a tie-breaker process by:

determining a first secondary value for the first endpoint;

determining a second secondary value for the second endpoint;

determining, based on the first criticality ranking, the first secondary value, the second criticality ranking, and the second secondary value, that one of the first endpoint and the second endpoint is a higher priority endpoint; and

triaging the higher priority endpoint before the other of the first endpoint and the second endpoint to mitigate endpoint damage to the datacenter if the datacenter becomes compromised.

2. The computer-implemented method of claim 1, further comprising:

determining a third endpoint is compromised, the first endpoint being the higher priority endpoint; and

determining a distance between the third endpoint and the first endpoint,

wherein,

the secondary value for the first endpoint is based on the distance.

3. The computer-implemented method of claim 1, wherein the determining of the first secondary value for the first endpoint further comprises:

determining a third endpoint is compromised, the first endpoint being the higher priority endpoint;

19

determining a similarity between the first endpoint and the third endpoint; and

determining the similarity likely caused the third endpoint to become compromised.

4. The computer-implemented method of claim 1, wherein determining the criticality ranking for the first endpoint comprises:

detecting, using a sensor installed on a computer hosting the first endpoint, a network flow associated with the first endpoint;

classifying the network flow to yield a classification; and determining a criticality of the classification.

5. The computer-implemented method of claim 1, wherein the determining of the second secondary value for the second endpoint is based on a redundancy of the first endpoint.

6. The computer-implemented method of claim 1, wherein,

the first endpoint is the higher priority endpoint, and the second endpoint is triaged after the first endpoint.

7. A non-transitory computer-readable medium having computer readable instructions that, when executed by a processor of a computer, cause the computer to:

determine a first criticality ranking for a first endpoint in a datacenter;

determine a second criticality ranking for a second endpoint; and

when the first criticality ranking and the second criticality ranking are determined to be a same criticality ranking, execute a tie-breaker process by:

determining a first secondary value for the first endpoint;

determining a second secondary value for the second endpoint;

determining, based on the first criticality ranking, the first secondary value, the first criticality ranking, and the second secondary value, that one of the first endpoint and the second endpoint is a higher priority endpoint; and

triaging the higher priority endpoint before the other of the first endpoint and the second endpoint to mitigate endpoint damage to the datacenter if the datacenter becomes compromised.

8. The non-transitory computer-readable medium of claim 7, wherein the instructions further cause the computer to:

determine a third endpoint is compromised, the first endpoint being the higher priority endpoint; and determine a distance between the third endpoint and the first endpoint,

wherein, the secondary value for the first endpoint is based on the distance.

9. The non-transitory computer-readable medium of claim 7, wherein the instructions that cause the computer to determine the first secondary value for the first endpoint further cause the computer to:

determine a third endpoint is compromised, the first endpoint being the higher priority endpoint;

determine a similarity between the first endpoint and the third endpoint; and

determine the similarity likely caused the third endpoint to become compromised.

10. The non-transitory computer-readable medium of claim 7, wherein the instructions that cause the computer to determine the criticality ranking for the first endpoint further cause the computer to:

20

detect, using a sensor, a network flow associated with the first endpoint;

classify the network flow to yield a classification; and determine a criticality of the classification.

11. The non-transitory computer-readable medium of claim 7, wherein the second secondary value for the second endpoint is determined based on a redundancy of the first endpoint.

12. The non-transitory computer-readable medium of claim 7,

wherein,

the first endpoint is the higher priority endpoint, and the second endpoint is triaged after the first endpoint.

13. A system comprising:

a processor;

memory including instructions that when executed by the processor, cause the system to:

determine a first criticality ranking for a first endpoint in a datacenter;

determine a second criticality ranking for a second endpoint; and

when the first criticality ranking and the second criticality ranking are determined to be a same criticality ranking, execute a tie-breaker process by:

determining a first secondary value for the first endpoint;

determining a second secondary value for the second endpoint;

determining, based on the first criticality ranking, the first secondary value, the second criticality ranking, and the second secondary value, that one of the first endpoint and the second endpoint is a higher priority endpoint; and

triaging the higher priority endpoint before the other of the first endpoint and the second endpoint to mitigate endpoint damage to the datacenter if the datacenter becomes compromised.

14. The system of claim 13, wherein the instructions further cause the system to:

determine a third endpoint is compromised, the first endpoint being the higher priority endpoint; and

determine a distance between the third endpoint and the first endpoint,

wherein,

the secondary value for the first endpoint is based on the distance.

15. The system of claim 13, wherein the instructions that cause the system to determine the first secondary value for the first endpoint further cause the system to:

determine a third endpoint is compromised, the first endpoint being the higher priority endpoint;

determine a similarity between the first endpoint and the third endpoint; and

determine the similarity likely caused the third endpoint to become compromised.

16. The system of claim 13, wherein the instructions that cause the system to determine the criticality ranking for the first endpoint further cause the system to:

detect, using a sensor installed on a computer hosting the first endpoint, a network flow associated with the first endpoint;

classify the network flow to yield a classification; and determine a criticality of the classification.

17. The system of claim 13, wherein the second secondary value for the second endpoint is determined based on a redundancy of the first endpoint.

18. The computer-implemented method of claim 1,  
wherein,

the first endpoint is determined to be the higher priority  
endpoint when the first criticality ranking equals the  
second criticality ranking and a first value associated  
with the first secondary value has a higher ranking  
than a second value associated with the second  
secondary value.

19. The non-transitory computer-readable medium of  
claim 7,  
wherein,

the first endpoint is determined to be the higher priority  
endpoint when the first criticality ranking equals the  
second criticality ranking and a first value associated  
with the first secondary value has a higher ranking  
than a second value associated with the second  
secondary value.

20. The system of claim 13,  
wherein,

the first endpoint is determined to be the higher priority  
endpoint when the first criticality ranking equals the  
second criticality ranking and a first value associated  
with the first secondary value has a higher ranking  
than a second value associated with the second  
secondary value.

\* \* \* \* \*