US Patent & Trademark Office
Patent Public Search | Text View

United States Patent	12388865
Kind Code	B2
Date of Patent	August 12, 2025
Inventor(s)	Sopan; Awalin Nabila

System and method for surfacing cyber-security threats with a self-learning recommendation engine

Abstract

Techniques for performing cyber-security alert analysis and prioritization according to machine learning employing a predictive model to implement a self-learning feedback loop. The system implements a method generating the predictive model associated with alert classifications and/or actions which automatically generated, or manually selected by cyber-security analysts. The predictive model is used to determine a priority for display to the cyber-security analyst and to obtain the input of the cyber-security analyst to improve the predictive model. Thereby the method implements a self-learning feedback loop to receive cyber-security alerts and mitigate the cyberthreats represented in the cybersecurity alerts.

Inventors:	**Sopan; Awalin Nabila (Reston, VA)**
Applicant:	**Google LLC** (Mountain View, CA)
Family ID:	**1000008749236**
Assignee:	**GOOGLE LLC (Mountain View, CA)**
Appl. No.:	**18/305898**
Filed:	**April 24, 2023**

Prior Publication Data

Document Identifier	Publication Date
US 20230336586 A1	Oct. 19, 2023

Related U.S. Application Data

continuation parent-doc US 16588967 20190930 US 11637862 child-doc US 18305898

Publication Classification

Int. Cl.: **H04L9/40** (20220101); **G06F11/32** (20060101); **G06F18/24** (20230101); **G06N20/00** (20190101)

U.S. Cl.:

CPC **H04L63/1466** (20130101); **G06F11/327** (20130101); **G06F18/24** (20230101); **G06N20/00** (20190101);

Field of Classification Search

CPC: H04L (63/1466); H04L (63/1416); H04L (63/1425); H04L (63/1433); G06N (20/00); G06N (5/025); G06F (18/24); G06F (11/327)

References Cited

U.S. PATENT DOCUMENTS

Patent No.	**Issued Date**	**Patentee Name**	**U.S. Cl.**	**CPC**
4292580	12/1980	Ott et al.	N/A	N/A
5175732	12/1991	Hendel et al.	N/A	N/A
5319776	12/1993	Hile et al.	N/A	N/A
5440723	12/1994	Arnold et al.	N/A	N/A
5490249	12/1995	Miller	N/A	N/A
5657473	12/1996	Killean et al.	N/A	N/A
5802277	12/1997	Cowlard	N/A	N/A
5842002	12/1997	Schnurer et al.	N/A	N/A
5960170	12/1998	Chen et al.	N/A	N/A
5978917	12/1998	Chi	N/A	N/A
5983348	12/1998	Ji	N/A	N/A
6088803	12/1999	Tso et al.	N/A	N/A
6092194	12/1999	Touboul	N/A	N/A
6094677	12/1999	Capek et al.	N/A	N/A
6108799	12/1999	Boulay et al.	N/A	N/A
6154844	12/1999	Touboul et al.	N/A	N/A
6269330	12/2000	Cidon et al.	N/A	N/A
6272641	12/2000	Ji	N/A	N/A
6279113	12/2000	Vaidya	N/A	N/A
6298445	12/2000	Shostack et al.	N/A	N/A
6357008	12/2001	Nachenberg	N/A	N/A
6424627	12/2001	Sorhaug et al.	N/A	N/A
6442696	12/2001	Wray et al.	N/A	N/A
6484315	12/2001	Ziese	N/A	N/A
6487666	12/2001	Shanklin et al.	N/A	N/A
6493756	12/2001	O'Brien et al.	N/A	N/A
6550012	12/2002	Villa et al.	N/A	N/A
6775657	12/2003	Baker	N/A	N/A
6831893	12/2003	Ben Nun et al.	N/A	N/A
6832367	12/2003	Choi et al.	N/A	N/A

6895550	12/2004	Kanchirayappa et al.	N/A	N/A
6898632	12/2004	Gordy et al.	N/A	N/A
6907396	12/2004	Muttik et al.	N/A	N/A
6941348	12/2004	Petry et al.	N/A	N/A
6971097	12/2004	Wallman	N/A	N/A
6981279	12/2004	Arnold et al.	N/A	N/A
7007107	12/2005	Ivchenko et al.	N/A	N/A
7028179	12/2005	Anderson et al.	N/A	N/A
7043757	12/2005	Hoefelmeyer et al.	N/A	N/A
7058822	12/2005	Edery et al.	N/A	N/A
7069316	12/2005	Gryaznov	N/A	N/A
7080407	12/2005	Zhao et al.	N/A	N/A
7080408	12/2005	Pak et al.	N/A	N/A
7093002	12/2005	Wolff et al.	N/A	N/A
7093239	12/2005	van der Made	N/A	N/A
7096498	12/2005	Judge	N/A	N/A
7100201	12/2005	Izatt	N/A	N/A
7107617	12/2005	Hursey et al.	N/A	N/A
7159149	12/2006	Spiegel et al.	N/A	N/A
7213260	12/2006	Judge	N/A	N/A
7231667	12/2006	Jordan	N/A	N/A
7240364	12/2006	Branscomb et al.	N/A	N/A
7240368	12/2006	Roesch et al.	N/A	N/A
7243371	12/2006	Kasper et al.	N/A	N/A
7249175	12/2006	Donaldson	N/A	N/A
7287278	12/2006	Liang	N/A	N/A
7308716	12/2006	Danford et al.	N/A	N/A
7328453	12/2007	Merkle, Jr. et al.	N/A	N/A
7346486	12/2007	Ivancic et al.	N/A	N/A
7356736	12/2007	Natvig	N/A	N/A
7386888	12/2007	Liang et al.	N/A	N/A
7392542	12/2007	Bucher	N/A	N/A
7418729	12/2007	Szor	N/A	N/A
7428300	12/2007	Drew et al.	N/A	N/A
7441272	12/2007	Durham et al.	N/A	N/A
7448084	12/2007	Apap et al.	N/A	N/A
7458098	12/2007	Judge et al.	N/A	N/A
7464404	12/2007	Carpenter et al.	N/A	N/A
7464407	12/2007	Nakae et al.	N/A	N/A
7467408	12/2007	O'Toole, Jr.	N/A	N/A
7478428	12/2008	Thomlinson	N/A	N/A
7480773	12/2008	Reed	N/A	N/A
7487543	12/2008	Arnold et al.	N/A	N/A
7496960	12/2008	Chen et al.	N/A	N/A
7496961	12/2008	Zimmer et al.	N/A	N/A
7519990	12/2008	Xie	N/A	N/A
7523493	12/2008	Liang et al.	N/A	N/A
7530104	12/2008	Thrower et al.	N/A	N/A
7540025	12/2008	Tzadikario	N/A	N/A

7546638	12/2008	Anderson et al.	N/A	N/A
7565550	12/2008	Liang et al.	N/A	N/A
7568233	12/2008	Szor et al.	N/A	N/A
7584455	12/2008	Ball	N/A	N/A
7603715	12/2008	Costa et al.	N/A	N/A
7607171	12/2008	Marsden et al.	N/A	N/A
7639714	12/2008	Stolfo et al.	N/A	N/A
7644441	12/2009	Schmid et al.	N/A	N/A
7657419	12/2009	van der Made	N/A	N/A
7676841	12/2009	Sobchuk et al.	N/A	N/A
7698548	12/2009	Shelest et al.	N/A	N/A
7707633	12/2009	Danford et al.	N/A	N/A
7712136	12/2009	Sprosts et al.	N/A	N/A
7730011	12/2009	Deninger et al.	N/A	N/A
7739740	12/2009	Nachenberg et al.	N/A	N/A
7779463	12/2009	Stolfo et al.	N/A	N/A
7784097	12/2009	Stolfo et al.	N/A	N/A
7832008	12/2009	Kraemer	N/A	N/A
7836502	12/2009	Zhao et al.	N/A	N/A
7849506	12/2009	Dansey et al.	N/A	N/A
7854007	12/2009	Sprosts et al.	N/A	N/A
7869073	12/2010	Oshima	N/A	N/A
7877803	12/2010	Enstone et al.	N/A	N/A
7904959	12/2010	Sidiroglou et al.	N/A	N/A
7908660	12/2010	Babl	N/A	N/A
7930738	12/2010	Petersen	N/A	N/A
7937387	12/2010	Frazier et al.	N/A	N/A
7937761	12/2010	Bennett	N/A	N/A
7949849	12/2010	Lowe et al.	N/A	N/A
7996556	12/2010	Raghavan et al.	N/A	N/A
7996836	12/2010	McCorkendale et al.	N/A	N/A
7996904	12/2010	Chiueh et al.	N/A	N/A
7996905	12/2010	Arnold et al.	N/A	N/A
8006305	12/2010	Aziz	N/A	N/A
8010667	12/2010	Zhang et al.	N/A	N/A
8020206	12/2010	Hubbard et al.	N/A	N/A
8028338	12/2010	Schneider et al.	N/A	N/A
8042184	12/2010	Batenin	N/A	N/A
8045094	12/2010	Teragawa	N/A	N/A
8045458	12/2010	Alperovitch et al.	N/A	N/A
8069484	12/2010	McMillan et al.	N/A	N/A
8087086	12/2010	Lai et al.	N/A	N/A
8171553	12/2011	Aziz et al.	N/A	N/A
8176049	12/2011	Deninger et al.	N/A	N/A
8176480	12/2011	Spertus	N/A	N/A
8201246	12/2011	Wu et al.	N/A	N/A
8204984	12/2011	Aziz et al.	N/A	N/A
8214905	12/2011	Doukhvalov et al.	N/A	N/A
8220055	12/2011	Kennedy	N/A	N/A
8225288	12/2011	Miller et al.	N/A	N/A

8225373	12/2011	Kraemer	N/A	N/A
8233882	12/2011	Rogel	N/A	N/A
8234640	12/2011	Fitzgerald et al.	N/A	N/A
8234709	12/2011	Viljoen et al.	N/A	N/A
8239944	12/2011	Nachenberg et al.	N/A	N/A
8260914	12/2011	Ranjan	N/A	N/A
8266091	12/2011	Gubin et al.	N/A	N/A
8286251	12/2011	Eker et al.	N/A	N/A
8291499	12/2011	Aziz et al.	N/A	N/A
8307435	12/2011	Mann et al.	N/A	N/A
8307443	12/2011	Wang et al.	N/A	N/A
8312545	12/2011	Tuvell et al.	N/A	N/A
8321936	12/2011	Green et al.	N/A	N/A
8321941	12/2011	Tuvell et al.	N/A	N/A
8332571	12/2011	Edwards, Sr.	N/A	N/A
8365286	12/2012	Poston	N/A	N/A
8365297	12/2012	Parshin et al.	N/A	N/A
8370938	12/2012	Daswani et al.	N/A	N/A
8370939	12/2012	Zaitsev et al.	N/A	N/A
8375444	12/2012	Aziz et al.	N/A	N/A
8381299	12/2012	Stolfo et al.	N/A	N/A
8402529	12/2012	Green et al.	N/A	N/A
8464340	12/2012	Ahn et al.	N/A	N/A
8479174	12/2012	Chiriac	N/A	N/A
8479276	12/2012	Vaystikh et al.	N/A	N/A
8479291	12/2012	Bodke	N/A	N/A
8510827	12/2012	Leake et al.	N/A	N/A
8510828	12/2012	Guo et al.	N/A	N/A
8510842	12/2012	Amit et al.	N/A	N/A
8516478	12/2012	Edwards et al.	N/A	N/A
8516590	12/2012	Ranadive et al.	N/A	N/A
8516593	12/2012	Aziz	N/A	N/A
8522348	12/2012	Chen et al.	N/A	N/A
8528086	12/2012	Aziz	N/A	N/A
8533824	12/2012	Hutton et al.	N/A	N/A
8539582	12/2012	Aziz et al.	N/A	N/A
8549638	12/2012	Aziz	N/A	N/A
8555391	12/2012	Demir et al.	N/A	N/A
8561177	12/2012	Aziz et al.	N/A	N/A
8566476	12/2012	Shiffer et al.	N/A	N/A
8566946	12/2012	Aziz et al.	N/A	N/A
8584094	12/2012	Dadhia et al.	N/A	N/A
8584234	12/2012	Sobel et al.	N/A	N/A
8584239	12/2012	Aziz et al.	N/A	N/A
8595834	12/2012	Xie et al.	N/A	N/A
8627476	12/2013	Satish et al.	N/A	N/A
8635696	12/2013	Aziz	N/A	N/A
8682054	12/2013	Xue et al.	N/A	N/A
8682812	12/2013	Ranjan	N/A	N/A
8689333	12/2013	Aziz	N/A	N/A

8695096	12/2013	Zhang	N/A	N/A
8713631	12/2013	Pavlyushchik	N/A	N/A
8713681	12/2013	Silberman et al.	N/A	N/A
8726392	12/2013	McCorkendale et al.	N/A	N/A
8739280	12/2013	Chess et al.	N/A	N/A
8776229	12/2013	Aziz	N/A	N/A
8782792	12/2013	Bodke	N/A	N/A
8789172	12/2013	Stolfo et al.	N/A	N/A
8789178	12/2013	Kejriwal et al.	N/A	N/A
8793278	12/2013	Frazier et al.	N/A	N/A
8793787	12/2013	Ismael et al.	N/A	N/A
8805947	12/2013	Kuzkin et al.	N/A	N/A
8806647	12/2013	Daswani et al.	N/A	N/A
8832829	12/2013	Manni et al.	N/A	N/A
8850570	12/2013	Ramzan	N/A	N/A
8850571	12/2013	Staniford et al.	N/A	N/A
8881234	12/2013	Narasimhan et al.	N/A	N/A
8881271	12/2013	Butler, II	N/A	N/A
8881282	12/2013	Aziz et al.	N/A	N/A
8898788	12/2013	Aziz et al.	N/A	N/A
8935779	12/2014	Manni et al.	N/A	N/A
8949257	12/2014	Shiffer et al.	N/A	N/A
8984638	12/2014	Aziz et al.	N/A	N/A
8990939	12/2014	Staniford et al.	N/A	N/A
8990944	12/2014	Singh et al.	N/A	N/A
8997219	12/2014	Staniford et al.	N/A	N/A
9009822	12/2014	Ismael et al.	N/A	N/A
9009823	12/2014	Ismael et al.	N/A	N/A
9027135	12/2014	Aziz	N/A	N/A
9071638	12/2014	Aziz et al.	N/A	N/A
9104867	12/2014	Thioux et al.	N/A	N/A
9106630	12/2014	Frazier et al.	N/A	N/A
9106694	12/2014	Aziz et al.	N/A	N/A
9118715	12/2014	Staniford et al.	N/A	N/A
9159035	12/2014	Ismael et al.	N/A	N/A
9171160	12/2014	Vincent et al.	N/A	N/A
9176843	12/2014	Ismael et al.	N/A	N/A
9189627	12/2014	Islam	N/A	N/A
9195829	12/2014	Goradia et al.	N/A	N/A
9197664	12/2014	Aziz et al.	N/A	N/A
9223972	12/2014	Vincent et al.	N/A	N/A
9225740	12/2014	Ismael et al.	N/A	N/A
9241010	12/2015	Bennett et al.	N/A	N/A
9251343	12/2015	Vincent et al.	N/A	N/A
9262635	12/2015	Paithane et al.	N/A	N/A
9268936	12/2015	Butler	N/A	N/A
9275229	12/2015	LeMasters	N/A	N/A
9282109	12/2015	Aziz et al.	N/A	N/A
9292686	12/2015	Ismael et al.	N/A	N/A
9294501	12/2015	Mesdaq et al.	N/A	N/A

9300686	12/2015	Pidathala et al.	N/A	N/A
9306960	12/2015	Aziz	N/A	N/A
9306974	12/2015	Aziz et al.	N/A	N/A
9311479	12/2015	Manni et al.	N/A	N/A
9355247	12/2015	Thioux et al.	N/A	N/A
9356944	12/2015	Aziz	N/A	N/A
9363280	12/2015	Rivlin et al.	N/A	N/A
9367681	12/2015	Ismael et al.	N/A	N/A
9398028	12/2015	Karandikar et al.	N/A	N/A
9413781	12/2015	Cunningham et al.	N/A	N/A
9426071	12/2015	Caldejon et al.	N/A	N/A
9430646	12/2015	Mushtaq et al.	N/A	N/A
9432389	12/2015	Khalid et al.	N/A	N/A
9438613	12/2015	Paithane et al.	N/A	N/A
9438622	12/2015	Staniford et al.	N/A	N/A
9438623	12/2015	Thioux et al.	N/A	N/A
9459901	12/2015	Jung et al.	N/A	N/A
9467460	12/2015	Otvagin et al.	N/A	N/A
9483644	12/2015	Paithane et al.	N/A	N/A
9495180	12/2015	Ismael	N/A	N/A
9497213	12/2015	Thompson et al.	N/A	N/A
9507935	12/2015	Ismael et al.	N/A	N/A
9516057	12/2015	Aziz	N/A	N/A
9519782	12/2015	Aziz et al.	N/A	N/A
9536091	12/2016	Paithane et al.	N/A	N/A
9537972	12/2016	Edwards et al.	N/A	N/A
9560059	12/2016	Islam	N/A	N/A
9565202	12/2016	Kindlund et al.	N/A	N/A
9591015	12/2016	Amin et al.	N/A	N/A
9591020	12/2016	Aziz	N/A	N/A
9594904	12/2016	Jain et al.	N/A	N/A
9594905	12/2016	Ismael et al.	N/A	N/A
9594912	12/2016	Thioux et al.	N/A	N/A
9609007	12/2016	Rivlin et al.	N/A	N/A
9626509	12/2016	Khalid et al.	N/A	N/A
9628498	12/2016	Aziz et al.	N/A	N/A
9628507	12/2016	Haq et al.	N/A	N/A
9633134	12/2016	Ross	N/A	N/A
9635039	12/2016	Islam et al.	N/A	N/A
9641546	12/2016	Manni et al.	N/A	N/A
9654485	12/2016	Neumann	N/A	N/A
9661009	12/2016	Karandikar et al.	N/A	N/A
9661018	12/2016	Aziz	N/A	N/A
9674298	12/2016	Edwards et al.	N/A	N/A
9680862	12/2016	Ismael et al.	N/A	N/A
9690606	12/2016	Ha et al.	N/A	N/A
9690933	12/2016	Singh et al.	N/A	N/A
9690935	12/2016	Shiffer et al.	N/A	N/A
9690936	12/2016	Malik et al.	N/A	N/A
9736179	12/2016	Ismael	N/A	N/A

9740857	12/2016	Ismael et al.	N/A	N/A
9747446	12/2016	Pidathala et al.	N/A	N/A
9756074	12/2016	Aziz et al.	N/A	N/A
9773112	12/2016	Rathor et al.	N/A	N/A
9781144	12/2016	Otvagin et al.	N/A	N/A
9787700	12/2016	Amin et al.	N/A	N/A
9787706	12/2016	Otvagin et al.	N/A	N/A
9792196	12/2016	Ismael et al.	N/A	N/A
9824209	12/2016	Ismael et al.	N/A	N/A
9824211	12/2016	Wilson	N/A	N/A
9824216	12/2016	Khalid et al.	N/A	N/A
9825976	12/2016	Gomez et al.	N/A	N/A
9825989	12/2016	Mehra et al.	N/A	N/A
9838408	12/2016	Karandikar et al.	N/A	N/A
9838411	12/2016	Aziz	N/A	N/A
9838416	12/2016	Aziz	N/A	N/A
9838417	12/2016	Khalid et al.	N/A	N/A
9846776	12/2016	Paithane et al.	N/A	N/A
9870298	12/2017	Jackson	N/A	G06F 11/3433
9876701	12/2017	Caldejon et al.	N/A	N/A
9888016	12/2017	Amin et al.	N/A	N/A
9888019	12/2017	Pidathala et al.	N/A	N/A
9910988	12/2017	Vincent et al.	N/A	N/A
9911319	12/2017	Malhotra	N/A	G08B 29/185
9912644	12/2017	Cunningham	N/A	N/A
9912681	12/2017	Ismael et al.	N/A	N/A
9912684	12/2017	Aziz et al.	N/A	N/A
9912691	12/2017	Mesdaq et al.	N/A	N/A
9912698	12/2017	Thioux et al.	N/A	N/A
9916440	12/2017	Paithane et al.	N/A	N/A
9921978	12/2017	Chan et al.	N/A	N/A
9934376	12/2017	Ismael	N/A	N/A
9934381	12/2017	Kindlund et al.	N/A	N/A
9946568	12/2017	Ismael et al.	N/A	N/A
9953185	12/2017	Bendersky	N/A	G06F 16/435
9954890	12/2017	Staniford et al.	N/A	N/A
9973531	12/2017	Thioux	N/A	N/A
10002252	12/2017	Ismael et al.	N/A	N/A
10019338	12/2017	Goradia et al.	N/A	N/A
10019573	12/2017	Silberman et al.	N/A	N/A
10025691	12/2017	Ismael et al.	N/A	N/A
10025927	12/2017	Khalid et al.	N/A	N/A
10027689	12/2017	Rathor et al.	N/A	N/A
10027690	12/2017	Aziz et al.	N/A	N/A
10027696	12/2017	Rivlin et al.	N/A	N/A
10033747	12/2017	Paithane et al.	N/A	N/A
10033748	12/2017	Cunningham et al.	N/A	N/A

10033753	12/2017	Islam et al.	N/A	N/A
10033759	12/2017	Kabra et al.	N/A	N/A
10050998	12/2017	Singh	N/A	N/A
10068091	12/2017	Aziz et al.	N/A	N/A
10075455	12/2017	Zafar et al.	N/A	N/A
10083302	12/2017	Paithane et al.	N/A	N/A
10084813	12/2017	Evada	N/A	N/A
10089461	12/2017	Ha et al.	N/A	N/A
10097573	12/2017	Aziz	N/A	N/A
10104102	12/2017	Neumann	N/A	N/A
10108446	12/2017	Steinberg et al.	N/A	N/A
10121000	12/2017	Rivlin et al.	N/A	N/A
10122746	12/2017	Manni et al.	N/A	N/A
10133863	12/2017	Bu et al.	N/A	N/A
10133866	12/2017	Kumar et al.	N/A	N/A
10146810	12/2017	Shiffer et al.	N/A	N/A
10148693	12/2017	Singh et al.	N/A	N/A
10165000	12/2017	Aziz et al.	N/A	N/A
10169585	12/2018	Pilipenko et al.	N/A	N/A
10176321	12/2018	Abbasi et al.	N/A	N/A
10181029	12/2018	Ismael et al.	N/A	N/A
10191861	12/2018	Steinberg et al.	N/A	N/A
10192052	12/2018	Singh et al.	N/A	N/A
10198574	12/2018	Thioux et al.	N/A	N/A
10200384	12/2018	Mushtaq et al.	N/A	N/A
10210329	12/2018	Malik et al.	N/A	N/A
10216927	12/2018	Steinberg	N/A	N/A
10218740	12/2018	Mesdaq et al.	N/A	N/A
10242185	12/2018	Goradia	N/A	N/A
10567402	12/2019	Comeaux et al.	N/A	N/A
11017321	12/2020	Mishra	N/A	G05B 23/0283
11637862	12/2022	Sopan	726/23	G06N 5/025
11870799	12/2023	Imrem	N/A	H04L 63/102
2001/0005889	12/2000	Albrecht	N/A	N/A
2001/0047326	12/2000	Broadbent et al.	N/A	N/A
2002/0018903	12/2001	Kokubo et al.	N/A	N/A
2002/0038430	12/2001	Edwards et al.	N/A	N/A
2002/0091819	12/2001	Melchione et al.	N/A	N/A
2002/0095607	12/2001	Lin-Hendel	N/A	N/A
2002/0116627	12/2001	Tarbotton et al.	N/A	N/A
2002/0144156	12/2001	Copeland	N/A	N/A
2002/0162015	12/2001	Tang	N/A	N/A
2002/0166063	12/2001	Lachman et al.	N/A	N/A
2002/0169952	12/2001	DiSanto et al.	N/A	N/A
2002/0184528	12/2001	Shevenell et al.	N/A	N/A
2002/0188887	12/2001	Largman et al.	N/A	N/A
2002/0194490	12/2001	Halperin et al.	N/A	N/A

2003/0021728	12/2002	Sharpe et al.	N/A	N/A
2003/0074578	12/2002	Ford et al.	N/A	N/A
2003/0084318	12/2002	Schertz	N/A	N/A
2003/0101381	12/2002	Mateev et al.	N/A	N/A
2003/0115483	12/2002	Liang	N/A	N/A
2003/0188190	12/2002	Aaron et al.	N/A	N/A
2003/0191957	12/2002	Hypponen et al.	N/A	N/A
2003/0200460	12/2002	Morota et al.	N/A	N/A
2003/0212902	12/2002	van der Made	N/A	N/A
2003/0229801	12/2002	Kouznetsov et al.	N/A	N/A
2003/0237000	12/2002	Denton et al.	N/A	N/A
2004/0003323	12/2003	Bennett et al.	N/A	N/A
2004/0006473	12/2003	Mills et al.	N/A	N/A
2004/0015712	12/2003	Szor	N/A	N/A
2004/0019832	12/2003	Arnold et al.	N/A	N/A
2004/0047356	12/2003	Bauer	N/A	N/A
2004/0083408	12/2003	Spiegel et al.	N/A	N/A
2004/0088581	12/2003	Brawn et al.	N/A	N/A
2004/0093513	12/2003	Cantrell et al.	N/A	N/A
2004/0111531	12/2003	Staniford et al.	N/A	N/A
2004/0117478	12/2003	Triulzi et al.	N/A	N/A
2004/0117624	12/2003	Brandt et al.	N/A	N/A
2004/0128355	12/2003	Chao et al.	N/A	N/A
2004/0165588	12/2003	Pandya	N/A	N/A
2004/0236963	12/2003	Danford et al.	N/A	N/A
2004/0243349	12/2003	Greifeneder et al.	N/A	N/A
2004/0249911	12/2003	Alkhatib et al.	N/A	N/A
2004/0255161	12/2003	Cavanaugh	N/A	N/A
2004/0268147	12/2003	Wiederin et al.	N/A	N/A
2005/0005159	12/2004	Oliphant	N/A	N/A
2005/0021740	12/2004	Bar et al.	N/A	N/A
2005/0033960	12/2004	Vialen et al.	N/A	N/A
2005/0033989	12/2004	Paletta et al.	N/A	N/A
2005/0050148	12/2004	Mohammadioun et al.	N/A	N/A
2005/0086523	12/2004	Zimmer et al.	N/A	N/A
2005/0091513	12/2004	Mitomo et al.	N/A	N/A
2005/0091533	12/2004	Omote et al.	N/A	N/A
2005/0091652	12/2004	Ross et al.	N/A	N/A
2005/0108562	12/2004	Khazan et al.	N/A	N/A
2005/0114663	12/2004	Cornell et al.	N/A	N/A
2005/0125195	12/2004	Brendel	N/A	N/A
2005/0149726	12/2004	Joshi et al.	N/A	N/A
2005/0157662	12/2004	Bingham et al.	N/A	N/A
2005/0183143	12/2004	Anderholm et al.	N/A	N/A
2005/0201297	12/2004	Deikari	N/A	N/A
2005/0210533	12/2004	Copeland et al.	N/A	N/A
2005/0238005	12/2004	Chen et al.	N/A	N/A
2005/0240781	12/2004	Gassoway	N/A	N/A
2005/0262562	12/2004	Gassoway	N/A	N/A

2005/0265331	12/2004	Stolfo	N/A	N/A
2005/0283839	12/2004	Cowburn	N/A	N/A
2006/0010495	12/2005	Cohen et al.	N/A	N/A
2006/0015416	12/2005	Hoffman et al.	N/A	N/A
2006/0015715	12/2005	Anderson	N/A	N/A
2006/0015747	12/2005	Van de Ven	N/A	N/A
2006/0021029	12/2005	Brickell et al.	N/A	N/A
2006/0021054	12/2005	Costa et al.	N/A	N/A
2006/0031476	12/2005	Mathes et al.	N/A	N/A
2006/0047665	12/2005	Neil	N/A	N/A
2006/0070130	12/2005	Costea et al.	N/A	N/A
2006/0075496	12/2005	Carpenter et al.	N/A	N/A
2006/0095968	12/2005	Portolani et al.	N/A	N/A
2006/0101516	12/2005	Sudaharan et al.	N/A	N/A
2006/0101517	12/2005	Banzhaf et al.	N/A	N/A
2006/0117385	12/2005	Mester et al.	N/A	N/A
2006/0123477	12/2005	Raghavan et al.	N/A	N/A
2006/0143709	12/2005	Brooks et al.	N/A	N/A
2006/0150249	12/2005	Gassen et al.	N/A	N/A
2006/0161983	12/2005	Cothrell et al.	N/A	N/A
2006/0161987	12/2005	Levy-Yurista	N/A	N/A
2006/0161989	12/2005	Reshef et al.	N/A	N/A
2006/0164199	12/2005	Gilde et al.	N/A	N/A
2006/0173992	12/2005	Weber et al.	N/A	N/A
2006/0179147	12/2005	Tran et al.	N/A	N/A
2006/0184632	12/2005	Marino et al.	N/A	N/A
2006/0191010	12/2005	Benjamin	N/A	N/A
2006/0221956	12/2005	Narayan et al.	N/A	N/A
2006/0236393	12/2005	Kramer et al.	N/A	N/A
2006/0242709	12/2005	Seinfeld et al.	N/A	N/A
2006/0248519	12/2005	Jaeger et al.	N/A	N/A
2006/0248582	12/2005	Panjwani et al.	N/A	N/A
2006/0251104	12/2005	Koga	N/A	N/A
2006/0288417	12/2005	Bookbinder et al.	N/A	N/A
2007/0006288	12/2006	Mayfield et al.	N/A	N/A
2007/0006313	12/2006	Porras et al.	N/A	N/A
2007/0011174	12/2006	Takaragi et al.	N/A	N/A
2007/0016951	12/2006	Piccard et al.	N/A	N/A
2007/0019286	12/2006	Kikuchi	N/A	N/A
2007/0033645	12/2006	Jones	N/A	N/A
2007/0038943	12/2006	Fitzgerald et al.	N/A	N/A
2007/0064689	12/2006	Shin et al.	N/A	N/A
2007/0074169	12/2006	Chess et al.	N/A	N/A
2007/0094730	12/2006	Bhikkaji et al.	N/A	N/A
2007/0101435	12/2006	Konanka et al.	N/A	N/A
2007/0128855	12/2006	Cho et al.	N/A	N/A
2007/0142030	12/2006	Sinha et al.	N/A	N/A
2007/0143827	12/2006	Nicodemus et al.	N/A	N/A
2007/0156895	12/2006	Vuong	N/A	N/A
2007/0157180	12/2006	Tillmann et al.	N/A	N/A

2007/0157306	12/2006	Elrod et al.	N/A	N/A
2007/0168988	12/2006	Eisner et al.	N/A	N/A
2007/0171824	12/2006	Ruello et al.	N/A	N/A
2007/0174915	12/2006	Gribble et al.	N/A	N/A
2007/0192500	12/2006	Lum	N/A	N/A
2007/0192858	12/2006	Lum	N/A	N/A
2007/0198275	12/2006	Malden et al.	N/A	N/A
2007/0208822	12/2006	Liang et al.	N/A	N/A
2007/0220607	12/2006	Sprosts et al.	N/A	N/A
2007/0240218	12/2006	Tuvell et al.	N/A	N/A
2007/0240219	12/2006	Tuvell et al.	N/A	N/A
2007/0240220	12/2006	Tuvell et al.	N/A	N/A
2007/0240222	12/2006	Tuvell et al.	N/A	N/A
2007/0250930	12/2006	Aziz et al.	N/A	N/A
2007/0256132	12/2006	Oliphant	N/A	N/A
2007/0271446	12/2006	Nakamura	N/A	N/A
2008/0005782	12/2007	Aziz	N/A	N/A
2008/0018122	12/2007	Zierler et al.	N/A	N/A
2008/0028463	12/2007	Dagon et al.	N/A	N/A
2008/0040710	12/2007	Chiriac	N/A	N/A
2008/0046781	12/2007	Childs et al.	N/A	N/A
2008/0066179	12/2007	Liu	N/A	N/A
2008/0072326	12/2007	Danford et al.	N/A	N/A
2008/0077793	12/2007	Tan et al.	N/A	N/A
2008/0080518	12/2007	Hoeflin et al.	N/A	N/A
2008/0086720	12/2007	Lekel	N/A	N/A
2008/0098476	12/2007	Syversen	N/A	N/A
2008/0120722	12/2007	Sima et al.	N/A	N/A
2008/0134178	12/2007	Fitzgerald et al.	N/A	N/A
2008/0134334	12/2007	Kim et al.	N/A	N/A
2008/0141376	12/2007	Clausen et al.	N/A	N/A
2008/0184367	12/2007	McMillan et al.	N/A	N/A
2008/0184373	12/2007	Traut et al.	N/A	N/A
2008/0189787	12/2007	Arnold et al.	N/A	N/A
2008/0201778	12/2007	Guo et al.	N/A	N/A
2008/0209557	12/2007	Herley et al.	N/A	N/A
2008/0215742	12/2007	Goldszmidt et al.	N/A	N/A
2008/0222729	12/2007	Chen et al.	N/A	N/A
2008/0263665	12/2007	Ma et al.	N/A	N/A
2008/0295172	12/2007	Bohacek	N/A	N/A
2008/0301810	12/2007	Lehane	N/A	N/A
2008/0307524	12/2007	Singh et al.	N/A	N/A
2008/0313738	12/2007	Enderby	N/A	N/A
2008/0320594	12/2007	Jiang	N/A	N/A
2009/0003317	12/2008	Kasralikar et al.	N/A	N/A
2009/0007100	12/2008	Field et al.	N/A	N/A
2009/0013408	12/2008	Schipka	N/A	N/A
2009/0031423	12/2008	Liu et al.	N/A	N/A
2009/0036111	12/2008	Danford et al.	N/A	N/A
2009/0037835	12/2008	Goldman	N/A	N/A

2009/0044024	12/2008	Oberheide et al.	N/A	N/A
2009/0044274	12/2008	Budko et al.	N/A	N/A
2009/0064332	12/2008	Porras et al.	N/A	N/A
2009/0077666	12/2008	Chen et al.	N/A	N/A
2009/0083369	12/2008	Marmor	N/A	N/A
2009/0083855	12/2008	Apap et al.	N/A	N/A
2009/0089879	12/2008	Wang et al.	N/A	N/A
2009/0094697	12/2008	Provos et al.	N/A	N/A
2009/0113425	12/2008	Ports et al.	N/A	N/A
2009/0125976	12/2008	Wassermann et al.	N/A	N/A
2009/0126015	12/2008	Monastyrsky et al.	N/A	N/A
2009/0126016	12/2008	Sobko et al.	N/A	N/A
2009/0133125	12/2008	Choi et al.	N/A	N/A
2009/0144823	12/2008	Lamastra et al.	N/A	N/A
2009/0158430	12/2008	Borders	N/A	N/A
2009/0172815	12/2008	Gu et al.	N/A	N/A
2009/0187992	12/2008	Poston	N/A	N/A
2009/0193293	12/2008	Stolfo et al.	N/A	N/A
2009/0198651	12/2008	Shiffer et al.	N/A	N/A
2009/0198670	12/2008	Shiffer et al.	N/A	N/A
2009/0198689	12/2008	Frazier et al.	N/A	N/A
2009/0199274	12/2008	Frazier et al.	N/A	N/A
2009/0199296	12/2008	Xie et al.	N/A	N/A
2009/0228233	12/2008	Anderson et al.	N/A	N/A
2009/0241187	12/2008	Troyansky	N/A	N/A
2009/0241190	12/2008	Todd et al.	N/A	N/A
2009/0265692	12/2008	Godefroid et al.	N/A	N/A
2009/0271867	12/2008	Zhang	N/A	N/A
2009/0300415	12/2008	Zhang et al.	N/A	N/A
2009/0300761	12/2008	Park et al.	N/A	N/A
2009/0328185	12/2008	Berg et al.	N/A	N/A
2009/0328221	12/2008	Blumfield et al.	N/A	N/A
2010/0005146	12/2009	Drako et al.	N/A	N/A
2010/0011205	12/2009	McKenna	N/A	N/A
2010/0017546	12/2009	Pao et al.	N/A	N/A
2010/0030996	12/2009	Butler, II	N/A	N/A
2010/0031353	12/2009	Thomas et al.	N/A	N/A
2010/0037314	12/2009	Derdisci et al.	N/A	N/A
2010/0043073	12/2009	Kuwamura	N/A	N/A
2010/0054278	12/2009	Stolfo et al.	N/A	N/A
2010/0058474	12/2009	Hicks	N/A	N/A
2010/0064044	12/2009	Nonoyama	N/A	N/A
2010/0077481	12/2009	Polyakov et al.	N/A	N/A
2010/0083376	12/2009	Pereira et al.	N/A	N/A
2010/0115621	12/2009	Staniford et al.	N/A	N/A
2010/0132038	12/2009	Zaitsev	N/A	N/A
2010/0154056	12/2009	Smith et al.	N/A	N/A
2010/0180344	12/2009	Malyshev et al.	N/A	N/A
2010/0192223	12/2009	Ismael et al.	N/A	N/A
2010/0220863	12/2009	Dupaquis et al.	N/A	N/A

2010/0235831	12/2009	Dittmer	N/A	N/A
2010/0251104	12/2009	Massand	N/A	N/A
2010/0281102	12/2009	Chinta et al.	N/A	N/A
2010/0281541	12/2009	Stolfo et al.	N/A	N/A
2010/0281542	12/2009	Stolfo et al.	N/A	N/A
2010/0287260	12/2009	Peterson et al.	N/A	N/A
2010/0299754	12/2009	Amit et al.	N/A	N/A
2010/0306173	12/2009	Frank	N/A	N/A
2011/0004737	12/2010	Greenebaum	N/A	N/A
2011/0025504	12/2010	Lyon et al.	N/A	N/A
2011/0041179	12/2010	Stahlberg	N/A	N/A
2011/0047594	12/2010	Mahaffey et al.	N/A	N/A
2011/0047620	12/2010	Mahaffey et al.	N/A	N/A
2011/0055907	12/2010	Narasimhan et al.	N/A	N/A
2011/0078794	12/2010	Manni et al.	N/A	N/A
2011/0093951	12/2010	Aziz	N/A	N/A
2011/0099620	12/2010	Stavrou et al.	N/A	N/A
2011/0099633	12/2010	Aziz	N/A	N/A
2011/0099635	12/2010	Silberman et al.	N/A	N/A
2011/0113231	12/2010	Kaminsky	N/A	N/A
2011/0145918	12/2010	Jung et al.	N/A	N/A
2011/0145920	12/2010	Mahaffey et al.	N/A	N/A
2011/0145934	12/2010	Abramovici et al.	N/A	N/A
2011/0167493	12/2010	Song et al.	N/A	N/A
2011/0167494	12/2010	Bowen et al.	N/A	N/A
2011/0173213	12/2010	Frazier et al.	N/A	N/A
2011/0173460	12/2010	Ito et al.	N/A	N/A
2011/0219449	12/2010	St. Neitzel et al.	N/A	N/A
2011/0219450	12/2010	McDougal et al.	N/A	N/A
2011/0225624	12/2010	Sawhney et al.	N/A	N/A
2011/0225655	12/2010	Niemela et al.	N/A	N/A
2011/0247072	12/2010	Staniford et al.	N/A	N/A
2011/0265182	12/2010	Peinado et al.	N/A	N/A
2011/0289582	12/2010	Kejriwal et al.	N/A	N/A
2011/0302587	12/2010	Nishikawa et al.	N/A	N/A
2011/0307954	12/2010	Melnik et al.	N/A	N/A
2011/0307955	12/2010	Kaplan et al.	N/A	N/A
2011/0307956	12/2010	Yermakov et al.	N/A	N/A
2011/0314546	12/2010	Aziz et al.	N/A	N/A
2012/0023593	12/2011	Puder et al.	N/A	N/A
2012/0054869	12/2011	Yen et al.	N/A	N/A
2012/0066698	12/2011	Yanoo	N/A	N/A
2012/0079596	12/2011	Thomas et al.	N/A	N/A
2012/0084859	12/2011	Radinsky et al.	N/A	N/A
2012/0096553	12/2011	Srivastava et al.	N/A	N/A
2012/0110667	12/2011	Zubrilin et al.	N/A	N/A
2012/0117652	12/2011	Manni et al.	N/A	N/A
2012/0121154	12/2011	Xue et al.	N/A	N/A
2012/0124426	12/2011	Maybee et al.	N/A	N/A
2012/0174186	12/2011	Aziz et al.	N/A	N/A

2012/0174196	12/2011	Bhogavilli et al.	N/A	N/A
2012/0174218	12/2011	McCoy et al.	N/A	N/A
2012/0198279	12/2011	Schroeder	N/A	N/A
2012/0210423	12/2011	Friedrichs et al.	N/A	N/A
2012/0222121	12/2011	Staniford et al.	N/A	N/A
2012/0255015	12/2011	Sahita et al.	N/A	N/A
2012/0255017	12/2011	Sallam	N/A	N/A
2012/0260342	12/2011	Dube et al.	N/A	N/A
2012/0266244	12/2011	Green et al.	N/A	N/A
2012/0278886	12/2011	Luna	N/A	N/A
2012/0297489	12/2011	Dequevy	N/A	N/A
2012/0330801	12/2011	McDougal et al.	N/A	N/A
2012/0331553	12/2011	Aziz et al.	N/A	N/A
2013/0014259	12/2012	Gribble et al.	N/A	N/A
2013/0036472	12/2012	Aziz	N/A	N/A
2013/0047257	12/2012	Aziz	N/A	N/A
2013/0074185	12/2012	McDougal et al.	N/A	N/A
2013/0086684	12/2012	Mohler	N/A	N/A
2013/0097699	12/2012	Balupari et al.	N/A	N/A
2013/0097706	12/2012	Titonis et al.	N/A	N/A
2013/0111587	12/2012	Goel et al.	N/A	N/A
2013/0117852	12/2012	Stute	N/A	N/A
2013/0117855	12/2012	Kim et al.	N/A	N/A
2013/0139264	12/2012	Brinkley et al.	N/A	N/A
2013/0160125	12/2012	Likhachev et al.	N/A	N/A
2013/0160127	12/2012	Jeong et al.	N/A	N/A
2013/0160130	12/2012	Mendelev et al.	N/A	N/A
2013/0160131	12/2012	Madou et al.	N/A	N/A
2013/0167236	12/2012	Sick	N/A	N/A
2013/0174214	12/2012	Duncan	N/A	N/A
2013/0185789	12/2012	Hagiwara et al.	N/A	N/A
2013/0185795	12/2012	Winn et al.	N/A	N/A
2013/0185798	12/2012	Saunders et al.	N/A	N/A
2013/0191915	12/2012	Antonakakis et al.	N/A	N/A
2013/0196649	12/2012	Padden et al.	N/A	N/A
2013/0227691	12/2012	Aziz et al.	N/A	N/A
2013/0246370	12/2012	Bartram et al.	N/A	N/A
2013/0247186	12/2012	LeMasters	N/A	N/A
2013/0263260	12/2012	Mahaffey et al.	N/A	N/A
2013/0291109	12/2012	Staniford et al.	N/A	N/A
2013/0298243	12/2012	Kumar et al.	N/A	N/A
2013/0318038	12/2012	Shiffer et al.	N/A	N/A
2013/0318073	12/2012	Shiffer et al.	N/A	N/A
2013/0325791	12/2012	Shiffer et al.	N/A	N/A
2013/0325792	12/2012	Shiffer et al.	N/A	N/A
2013/0325871	12/2012	Shiffer et al.	N/A	N/A
2013/0325872	12/2012	Shiffer et al.	N/A	N/A
2014/0032875	12/2013	Butler	N/A	N/A
2014/0053260	12/2013	Gupta et al.	N/A	N/A
2014/0053261	12/2013	Gupta et al.	N/A	N/A

2014/0130158	12/2013	Wang et al.	N/A	N/A
2014/0137180	12/2013	Lukacs et al.	N/A	N/A
2014/0169762	12/2013	Ryu	N/A	N/A
2014/0179360	12/2013	Jackson et al.	N/A	N/A
2014/0181131	12/2013	Ross	N/A	N/A
2014/0189687	12/2013	Jung et al.	N/A	N/A
2014/0189866	12/2013	Shiffer et al.	N/A	N/A
2014/0189882	12/2013	Jung et al.	N/A	N/A
2014/0237600	12/2013	Silberman et al.	N/A	N/A
2014/0280245	12/2013	Wilson	N/A	N/A
2014/0283037	12/2013	Sikorski et al.	N/A	N/A
2014/0283063	12/2013	Thompson et al.	N/A	N/A
2014/0328204	12/2013	Klotsche et al.	N/A	N/A
2014/0337836	12/2013	Ismael	N/A	N/A
2014/0344926	12/2013	Cunningham et al.	N/A	N/A
2014/0351935	12/2013	Shao et al.	N/A	N/A
2014/0380473	12/2013	Bu et al.	N/A	N/A
2014/0380474	12/2013	Paithane et al.	N/A	N/A
2015/0007312	12/2014	Pidathala et al.	N/A	N/A
2015/0096022	12/2014	Vincent et al.	N/A	N/A
2015/0096023	12/2014	Mesdaq et al.	N/A	N/A
2015/0096024	12/2014	Haq et al.	N/A	N/A
2015/0096025	12/2014	Ismael	N/A	N/A
2015/0163242	12/2014	Laidlaw	726/22	H04L 63/1425
2015/0180886	12/2014	Staniford et al.	N/A	N/A
2015/0186645	12/2014	Aziz et al.	N/A	N/A
2015/0199513	12/2014	Ismael et al.	N/A	N/A
2015/0199531	12/2014	Ismael et al.	N/A	N/A
2015/0199532	12/2014	Ismael et al.	N/A	N/A
2015/0220735	12/2014	Paithane et al.	N/A	N/A
2015/0372980	12/2014	Evada	N/A	N/A
2016/0004869	12/2015	Ismael et al.	N/A	N/A
2016/0006756	12/2015	Ismael et al.	N/A	N/A
2016/0044000	12/2015	Cunningham	N/A	N/A
2016/0127393	12/2015	Aziz et al.	N/A	N/A
2016/0191547	12/2015	Zafar et al.	N/A	N/A
2016/0191550	12/2015	Ismael et al.	N/A	N/A
2016/0261612	12/2015	Mesdaq et al.	N/A	N/A
2016/0285914	12/2015	Singh et al.	N/A	N/A
2016/0301703	12/2015	Aziz	N/A	N/A
2016/0335110	12/2015	Paithane et al.	N/A	N/A
2017/0063901	12/2016	Muddu et al.	N/A	N/A
2017/0083703	12/2016	Abbasi et al.	N/A	N/A
2017/0243133	12/2016	Zavesky	N/A	G06N 20/00
2018/0013770	12/2017	Ismael	N/A	N/A
2018/0048660	12/2017	Paithane et al.	N/A	N/A
2018/0121316	12/2017	Ismael et al.	N/A	N/A
2018/0150758	12/2017	Niininen	N/A	N/A

2018/0288077	12/2017	Siddiqui et al.	N/A	N/A
2018/0367561	12/2017	Givental et al.	N/A	N/A
2019/0020667	12/2018	Parker	N/A	N/A
2019/0236458	12/2018	Taylor	N/A	G06F 17/17
2019/0260779	12/2018	Bazalgette	N/A	H04L 51/224
2019/0334849	12/2018	Bostick et al.	N/A	N/A
2020/0145358	12/2019	Yegorin et al.	N/A	N/A
2020/0151326	12/2019	Patrich	N/A	G06F 21/554
2020/0401696	12/2019	Ringlein et al.	N/A	N/A
2021/0058357	12/2020	Baughman et al.	N/A	N/A
2022/0174088	12/2021	Zorlular	N/A	H04L 41/22

FOREIGN PATENT DOCUMENTS

Patent No.	**Application Date**	**Country**	**CPC**
112567367	12/2018	CN	N/A
2990984	12/2015	EP	N/A
2439806	12/2007	GB	N/A
2490431	12/2011	GB	N/A
2520987	12/2014	GB	N/A
WO 0206928	12/2001	WO	N/A
WO 0223805	12/2001	WO	N/A
WO 2007117636	12/2006	WO	N/A
WO 2008041950	12/2007	WO	N/A
WO 2011084431	12/2010	WO	N/A
WO 2011112348	12/2010	WO	N/A
WO 2012075336	12/2011	WO	N/A
WO 2012145066	12/2011	WO	N/A
WO 2013067505	12/2012	WO	N/A
WO-2018126286	12/2017	WO	G06N 20/00

OTHER PUBLICATIONS

Abdullah et al., “Visualizing Network Data for Intrusion Detection.”, 2005 Institute of Electrical and Electronics Engineers Workshop on Information Assurance and Security, United States Military Academy, West Point, New York, United States, pp. 100-108. cited by applicant

Adetoye et al., “Network Intrusion Detection & Response System.”, Adetoye, Sep. 2003. cited by applicant

Apostolopoulos, “V-eM: A cluster of Virtual Machines for Robust, Detailed, and High-Performance Network Emulation.”, Fourteenth Institute of Electrical and Electronics Engineers International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Sep. 11-14, 2006, pp. 117-126. cited by applicant

Aura et al., “Scanning Electronic Documents for Personally Identifiable Information.”, WPES '06: Fifth Association for Computing Machinery Workshop on Privacy in Electronic Society, Alexandria, Virginia, United States, Oct. 30, 2006, pp. 41-50. cited by applicant

Baecher et al., “The Nepenthes Platform: An Efficient Approach to Collect Malware.”, RAID '06: Ninth International Symposium on Recent Advances in Intrusion Detection, Hamburg, Germany, Sep. 20-22, 2006, pp. 165-184. cited by applicant

Bayer et al., “Dynamic Analysis of Malicious Code.”, Journal in Computer Virology, vol. 2, Nov. 2006, pp. 67-77. cited by applicant

Boubalos, “Extracting Syslog Data Out of Raw PCAP Dumps.”, seclists.org, retrieved on Aug. 8, 2023, Honeypots Mailing List Archives, https://seclists.org/honeypots/2003/q2/319, Jun. 5, 2003, pages. cited by applicant

Chaudet, et al., “Optimal Positioning of Active and Passive Monitoring Devices.”, CoNEXT '05: 2005 Association for Computing Machinery Conference on Emerging Network Experiment and Technology, Toulouse, France, Oct. 24-27, 2005, pp. 71-82. cited by applicant

Chen et al., “When Virtual is Better Than Real.”, Eighth Workshop on Hot Topics in Operating Systems, Elmau, Germany, May 20-22, 2001, 6 pages. cited by applicant

Christodorescu et al., “Mining Specifications of Malicious Behavior.”, ESEC-FSE '07: Sixth Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, Dubrovnik, Croatia, Sep. 3-7, 2007, pp. 5-14. cited by applicant

Cisco, “Intrusion Prevention for the Cisco ASA 5500-x Series.”, Data Sheet 2012. cited by applicant

Cohen, “PyFlag—An Advanced Network Forensic Framework.”, Digital Investigation: The International Journal of Digital Forensics & Incident Response, vol. 5, Sep. 2008, pp. S112-S120. cited by applicant

Costa et al., “Vigilante: End-to-End Containment of Internet Worms.”, Association for Computing Machinery Transactions on Computer Systems, vol. 26, No. 4, Article 9, Dec. 2008, 68 pages. cited by applicant

Distler, “Malware Analysis: An Introduction.”, SANS Institute InfoSec Reading Room, SANS Institute, Feb. 12, 2008, 67 pages. cited by applicant

Dunlap et al., “ReVirt: Enabling Intrusion Analysis through Virtual-Machine Logging and Replay.”, Fifth Symposium on Operating Systems Design and Implementation, OSDI '02 Fifth Symposium on Operating Systems Design and Implementation, USENIX Association, Boston, Massachusetts, United States, Dec. 9-11, 2002, 14 pages. cited by applicant

FireEye, “Malware Analysis.”, Modern Malware Forensics, FireEye Inc., 2010. cited by applicant

FireEye.com, “FireEye Malware Analysis & Exchange Network, Malware Protection System.”, Feb. 2010, chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://people.ucsc.edu/˜warner/fireeye.pdf, retrieved on Sep. 19, 2023, 2 pages. cited by applicant

Goel et al., “Reconstructing System State for Intrusion Analysis.”, Apr. 2008 Special Interest Group on Operating Systems Review, vol. 42, Issue 3, Apr. 2008, pp. 21-28. cited by applicant

Idika et al., “A Survey of Malware Detection Techniques.”, Department of Computer Science, Purdue University, Feb. 2, 2007, 48 pages. cited by applicant

Keizer, “Microsoft's HoneyMonkeys Show Patching Windows Works.”, Aug. 8, 2005, https://www.informationweek.com/it-life/microsoft-s-honeymonkeys-show-patching-windows-works#, retrieved Jun. 1, 2006, 4 pages. cited by applicant

Kim et al., “Autograph: Toward Automated, Distributed Worm Signature Detection.”, Thirteenth USENIX Security Symposium, San Diego, California, United States, Aug. 9-13, 2004, pp. 271-286. cited by applicant

King et al., “Operating System Support for Virtual Machines.”, 2003 USENIX Annual Technical Conference, San Antonio, Texas, United States, Jun. 9-14, 2003, 15 pages. cited by applicant

Kreibich et al., “Honeycomb-Creating Intrusion Detection Signatures Using Honeypots.”, ACM SIGCOMM Computer Communication Review, vol. 34, Issue 1, Jan. 2004, pp. 51-56. cited by applicant

Kristoff, “Botnets, Detection and Mitigation: DNS-Based Techniques.”, Security Day, 2005, 23 pages. cited by applicant

Lastline Labs, “The Threat of Evasive Malware.”, Feb. 25, 2013, Lastline Labs, pp. 1-8. cited by applicant

Li et al., “A VMM-Based System Call Interposition Framework for Program Monitoring.”, Institute of Electrical and Electronics Engineers Sixteenth International Conference on Parallel and Distributed Systems, Shanghai, China, Dec. 2010, pp. 706-711. cited by applicant

Lindorfer et al., “Detecting Environment-Sensitive Malware.”, RAID 2011: Fourteenth International Symposium on Recent Advances in Intrusion Detection, Menlo Park, California, United States, Sep. 20-21, 2011. cited by applicant

Marchette, “Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint.”, 2001. cited by applicant

Moore et al., “Internet Quarantine: Requirements for Containing Self-Propagating Code.”, IEEE INFOCOM 2003: Twenty-second Annual Joint Conference of the Institute of Electrical and Electronics Engineers Computer and Communications Societies, San Francisco, California, United States, Mar. 30-Apr. 3, 2003, pp. 1901-1910. cited by applicant

Morales et al., “Analyzing and Exploiting Network Behaviors of Malware.”, Security and Privacy in Communication Networks. Springer Berlin Heidelberg, 2010, pp. 20-34. cited by applicant

Mori, “Detecting Unknown Computer Viruses.”, 2004, Springer-Verlag Berlin Heidelberg. cited by applicant

Natvig, “Sandboxii: Internet.”, Virus Bulletin Conference 2001, Sep. 2002, 18 pages. cited by applicant

NetBIOS Working Group, “Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods.”, STD 19, RFC 1001, Mar. 1987. cited by applicant

“Network Security NetDetector-Network Intrusion Forensic System (NIFS).”, NetDetector Whitepaper, 2003, 11 pages. cited by applicant

Newsome et al., “Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software.”, Twelfth Annual Network and Distributed System Security Symposium (NDSS '05), San Diego, California, United States, Feb. 2-4, 2005, 17 pages. cited by applicant

Nojiri et al., “Cooperation Response Strategies for Large Scale Attack Mitigation.”, DARPA Information Survivability Conference and Exposition, vol. 1, Apr. 22-24, 2003, pp. 293-302. cited by applicant

Oberheide et al., “CloudAV: N-Version Antivirus in the Network Cloud.”, Seventeenth USENIX Security Symposium USENIX Security '08 Jul. 28-Aug. 1, 2008, San Jose, California, United States, 20 pages. cited by applicant

Ptacek et al., “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection.”, Secure Networks, Jan. 1998, 63 pages. cited by applicant

Roundy et al., “Hybrid Analysis and Control of Malware.”, RAID 2010: Thirteenth International Symposium Recent Advances in Intrusion Detection, Ottawa, Ontario, Canada, Sep. 15-17, 2010, pp. 317-338. cited by applicant

Sailer et al., “sHype: Secure Hypervisor Approach to Trusted Virtualized Systems.”, IBM Research Division, Feb. 2, 2005, 13 pages. cited by applicant

Salah et al., “Using Cloud Computing to Implement a Security Overlay Network.”, Institute of Electrical and Electronics Engineers Security and Privacy Magazine, vol. 11, No. 1, Jan. 2013, pp. 44-53. cited by applicant

Shinotsuka, “Malware Authors Using New Techniques to Evade Automated Threat Analysis Systems.”, Oct. 26, 2012, http://www.symantec.com/connect/blogs/, pp. 1-4. cited by applicant

Silicon Defense, “Worm Containment in the Internal Network.”, Mar. 2003, pp. 1-25. cited by applicant

Singh et al., “Automated Worm Fingerprinting.”, Proceedings of the ACM/USENIX Symposium on Operating System Design and Implementation, San Francisco, California, United States, Dec. 2004. cited by applicant

Stevens, “Malicious PDF Documents Explained.”, Security & Privacy, Institute of Electrical and Electronics Engineers Security & Privacy, vol. 9, No. 1, Jan.-Feb. 2011, pp. 80-82. cited by applicant
Yin et al., “Panorama: Capturing System-Wide Information Flow for Malware Detection and Analysis.”, CCS '07: Fourteenth ACM Conference on Computer and Communications Security, Oct. 29-Nov. 2, 2007, Alexandria, Virginia, United States, pp. 116-127. cited by applicant

Primary Examiner: Su; Sarah

Attorney, Agent or Firm: DORITY & MANNING P.A.

Background/Summary

PRIORITY CLAIM (1) The present application is a continuation of U.S. application Ser. No. 16/588,967 having a filing date of Sep. 30, 2019, now U.S. Pat. No. 11,637,862. Applicant claims priority to and the benefit of each of such applications and incorporate all such applications herein by reference in its entirety.

FIELD OF THE INVENTION

(1) The present disclosure relates, generally, to cyber-security and more specifically to techniques to facilitate the analysis and remediation of cyberattacks.

BACKGROUND

(2) Cyber-security threats are a major risk to enterprises and individuals alike. Enterprises rely on security operations centers (“SOC”) and the analysts operating SOCs, to identify, respond to, and mitigate the consequences of cyber-security threats targeting the enterprise's systems. SOC analysts are inundated with cyber-security alerts received from a variety of cyber-security products deployed to protect an enterprise. To reduce the vast volume of alerts to be addressed by SOC analysts, some SOCs filter alerts (e.g., for duplicates, known false positives, and low priority alerts, etc.) before they are presented to a SOC analyst.

Description

BRIEF DESCRIPTION OF THE FIGURES

(1) Embodiments of the disclosure are illustrated by way of example and not by way of limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

(2) FIG. **1** is an exemplary block diagram of an automated analyst alerting system communicatively coupled to one or more cyber-security devices via a communication network, in accordance with an embodiment of the invention.

(3) FIG. **2** illustrates a logical representation of the automated analyst alerting system of FIG. **1**.

(4) FIG. **3** is an exemplary flowchart of the operations of the automated analyst alerting system of FIGS. **1** and **2**.

DETAILED DESCRIPTION

(5) The detailed description below, describes a technology wherein a cyber-security automated analyst alerting system receives one or more cyber-security alerts, the alerts are analyzed by an alert recommendation engine which automatically determines a recommended course of action related to the one or more received cyber-security alerts by application of a predictive machine learning model generated by a predictive machine learning logic (or predictive model generation

logic). The predictive machine learning logic generates a machine learning model (or more simply, “model”), for use by the alert recommendation engine, in response to changes in a knowledge store. More specifically, to automatically determine a recommended course of action (i.e. a set of one or more instructions, or commands, issued by the described system to mitigate a cyber-security threat), the alert analysis and labeling engine generates a modified alert including at least one classification, classification confidence level, and contextual data for each alert according to the predictive machine learning model, to create a modified alert which are provided to the action generator. The action generator (or in some embodiments through the execution of an engine processing a separate action predictive machine learning model) determines a recommended course of action according to the predictive machine learning model and generates a signal through a reporting logic to present the modified alert for display to an analyst.

(6) More specifically, the automated analyst alerting system (“AAAS”) is configured to receive an alert (the received alert is received from one or more alert-generating cyber-security devices), analyze the alert according to a model generated by a machine learning procedure applied to data in a knowledge store. The knowledge store includes data that associates previously detected alerts, cyber-security threats, and undesirable computing device configurations with one or more classifications as determined by a cyber-security analyst. Such classifications may include labels (e.g., “malicious”, “non-malicious”, “phishing”, “misconfiguration”, etc.) and a confidence level associated with the classification. For example, a received cyber-security alert received by the system and analyzed by the AAAS may classify the alert as “malicious” with a 17% confidence level, “non-malicious” with an 89% confidence level, and “misconfiguration” with a 91% confidence level. The classifications and their associated confidence levels are provided with the received alert, as well as with additional context related to the received alert, to create a modified alert and are provided to an action generator. The additional context may be based on prior selections of analysts, the prior selections stored in a knowledge store, and/or prior selections made by an expert system configured to make recommended actions based on associated received alerts. The knowledge store may be located locally and/or remotely via a network connection. In some embodiments the additional context may include information generated by the AAAS identifying a set of prior alerts (e.g., stored in the knowledge store) as being associated with a received alert and thereby identifying an advanced persistent cyber-security threat (i.e. a prolonged and targeted cyberattack in which an intruder may repeatedly attempt to gain access to a targeted network, computing device or user thereof). Based on the persistent cyber-security threat, the AAAS may modify the classifications and/or further classify the received alert as associated with the persistent cyber-security threat.

(7) The predictive machine learning logic is configured to generate a predictive model based on data stored in the knowledge store. The data stored in the knowledge store may include the classifications associated with alerts that were previously received and classified (confirmed or reclassified) by cyber-security analysts. The knowledge store may also store mitigative actions selected by and/or input by a cyber-security analyst. The knowledge store may also be used to store meta-information associated with the success or failure of automated or manually selected mitigations and consequently create a self-learning feedback loop. The self-learning feedback loop surfaces classifications and actions for the cyber-security analysts.

(8) The predictive machine learning logic may be co-located with the alert recommendation engine and/or remotely located. The predictive machine learning logic generates a predictive model according to conventional machine learning techniques (e.g., support vector machines, artificial neural networks, etc.) applied to the data stored in the knowledge store, in a process known as “training”. The training system may include information extracted from received alerts and stored as data in the knowledge store. The information extracted from the received alert may include received alert message content as well as well as meta-information associated with the received alert (e.g., time of receipt, IP address of the source cyber-security device, etc.). The training system

may also include information associated with the received alert (e.g., modifying a label associated with alert or associating a course of action with the alert) by the cyber-security analyst and stored in the knowledge store. Based on information stored in the knowledge store, the predictive machine learning logic may generate the predictive model which, when applied to a received alert, may be used to classify and determine one or more courses of action related to the received alert using machine learning.

(9) In some embodiments, the generated predictive model may be used by one or more classifiers to determine a probability of the accuracy (i.e. confidence level) of a label for each alert. The classifiers may classify each alert based on a label as determined by an analyst and/or the alert recommendation engine according to the predictive model. In some embodiments, analysts may select from a pre-defined set of labels, whereas, in other embodiments, labeling may be done automatically. A classifier may generate a probability of association with a label relating to each received alert.

(10) Upon receipt of new data in the knowledge store, or periodically or aperiodically to account for any such newly stored data, the predictive machine learning logic generates a new predictive model by analyzing the data to determine associative relationships. In some embodiments, the application of a predictive model to a received alert may generate one or more labels and/or courses of actions, each associated with a confidence level. The confidence levels are correlated with a likelihood of the alert being associated with the label and/or course of action. The newly generated predictive model may be based on additional data—e.g., verification of a prior classification (e.g., of a classification made by the alert recommendation engine and, in some embodiments confirmed by the analyst), newly associated courses of actions (i.e. mitigative actions responsive to a received alert), where the association may be made automatically or made or confirmed by an analyst, and/or new information associated with alert classification provided to the knowledge store via an update mechanism. The newly generated predictive model is applied to newly received alerts by the alert recommendation engine for classification, thereby creating a self-learning feedback loop. The classification is responsive to the labels resulting from application of the predictive model to the received alert.

(11) The action generator receives the modified alerts and associated context information to determine a recommended course of action for presentation via the reporting logic. The action generator determines a recommended course of action based on the application of a predictive model generated by the predictive model generation logic. The received modified alerts are analyzed by the action generator to determine a priority for presentation to an analyst. To determine a priority associated with the modified alert, the action generator may analyze the confidence levels (e.g., associated with a course of action determined by application of the predictive model, associated with a classification label, etc.). The priority assigned to a received alert may be based, at least in part, on a numerical distance of the confidence level a threshold, such as, for example, an automated execution threshold. For example, if the confidence associated with an action is 55% and the confidence threshold for automated execution of an action is 90%, the action generator may determine that the confidence associated with an action is too far from the threshold to be automatically actionable and should be displayed to an analyst and therefore given a higher priority for the analyst's attention. Similarly, if the confidence if the confidence associated with an action is 85% and the confidence threshold for automated execution of an action remains 90%, the action generator may determine that the confidence associated with an action is near the threshold, however, because it is not above the automatically actionable threshold, the received alert should be displayed to an analyst and therefore given a lesser priority than in the prior example. If a cyber-security threat or serious configuration issue requiring mitigation is detected (e.g., based on a classification and/or course of action), the action generator may determine whether the mitigation requires analyst attention (e.g., for selection) or if a recommended course of action may be automatically processed. To determine if analyst attention is required, the action generator

determines if a course of action from the knowledge store and/or the expert system is applicable. A course of action is applicable if the action generator determines a level of correlation (i.e. confidence level) between a course of action and the modified alert exceeds a confidence threshold. If a course of action is automatically executed and fails to resolve the alert, the system may provide the modified alert associated with the failed action to the reporting logic for display to the analyst. If the action generator receives an alert associated with a persistent cyber-security threat, it may assign a priority to the modified alert and provide the priority to the presentation logic for display to an analyst. The action generator provides a further modified alert, the further modified alert combining the modified alert received by the action generator with the resulting course of actions, if applicable.

(12) The further modified alert is provided to the presentation logic for layout composition. A layout is the way in which the modified alerts are composed for further review by the analyst. In some embodiments the layout may be composed for presentation to an analyst, in different layouts, according to the analyst's role. In some embodiments the modified alert may be presented to the analyst in different windows or otherwise highlighted, according to the assigned priority.

(13) The presentation logic receives the further modified alert to determine if the further modified alert is to be presented to an analyst for further review. The presentation logic may determine, based on the assigned priority of the further modified alert, to present the further modified alert to a cyber-security analyst. The presentation logic may determine, that a further modified alert shall not be presented to the cyber-security analyst due the relative priority (e.g., lesser) compared to other further modified alerts presented to the analyst at the same time. The relative priority of a further modified alert may increase (or decrease) based on selections made by a cyber-security analyst (e.g., as an analyst processes and addresses a first further modified alert, the relative priority of other further modified alerts may increase and be presented to the analyst).

(14) The presentation logic may also process the course of action data included in the further modified alert to determine if a course of action may be automatically executed. A course of action to be automatically executed may be identified by the further modified alert. Automatic execution of the course of action may require communication with a conventional external computing device that is configured to effectuate the course of action (e.g., a firewall, switch, server or endpoint system) connected to the network via the network interface. The mitigation logic receives a course of action for processing, the course of action may be received via the presentation logic if automatically selected or via an analyst interface when selected by an analyst. The mitigation logic initiates an external computing device (e.g., a cyber-security device, etc.) to execute a mitigation (i.e. via a course of action) sent by the mitigation logic.

(15) More specifically, the mitigation logic processes the course of action received and launches processes based on the course of action. The executed course of action includes at least one process to be executed. Some processes to be executed as a course of action may require communication with one or more external computing devices through an interface (e.g., API calls to external computing devices, etc.). In some embodiments, courses of action may include more than one process, each process may be required by the course of action to be processed in series or parallel (in a temporally overlapping manner). A process may be required to be executed in series if the output of a first process is required as input of a subsequent process. If a process of the course of action executed does not process successfully, an alert may be generated by the mitigation logic and provided to the presentation logic for display to the cyber-security analyst. For example, a course of action may require a process A and a process B to operate in series. Process A may include the execution of an API call to a network connected firewall requesting the status of port **8081**, while Process B executes a process receiving the status, and if the status is “open”, executes an API call to the network connected firewall to close port **8081**. Based on the success of the execution of the processes of the course of action, the mitigation logic communicates to the presentation logic. In some embodiments, the mitigation logic may provide an error message to the

presentation logic, describing the nature of the failure if the course of action did not successfully complete. The meta-information associated with the processing by the mitigation logic (e.g., error messages, process success or failure, course of action success or failure, etc.) is provided in the form of an execution message. The mitigation logic may be configured to automatically, manually, or semi-automatically process courses of action.

(16) The presentation logic receives data associated with the processing of a course of action by the mitigation logic (i.e. an execution message), via the mitigation logic. The data included in the received execution message is associated with the further modified alert and a determination is made by the presentation logic to present to an analyst. For example, the analyst may be provided a notification of a successful (or failed) execution of a course of action. In some embodiments an analyst may be presented with an alert describing the failed execution of a course of action as well as the associated further modified alert. The presentation logic provides the further modified alert to the storage logic for further processing.

(17) The storage logic receives the further modified alert, from the presentation logic, and the associated execution message, and determines if the content received (e.g., the data associated with the further modified alert obtained from the execution message) should be stored in the knowledge store. The further modified alert may contain information about selections and results of course of action selected by an analyst and/or automatically selected by the presentation logic. The storage logic may parse the further modified alert to extract the selection of a course of action by an analyst to store in the knowledge store. In some embodiments, the storage logic may determine that a selected course of action need not be stored in the knowledge store based on the success and/or failure of the course of action. In some other embodiments an execution message may be received directly from the mitigation logic, instead of being received via the presentation logic. Once processed by the storage logic, the presentation alert is provided to the reporting engine for display to the analyst.

(18) The reporting logic is configured to provide reports via an interface to an analyst and/or a system administrator. The reporting logic may provide reports via an analyst interface and/or a network interface. The reporting logic generates the report for the analyst based on information provided by a received further modified alert. The reporting logic may be configured to generate discrete reports and/or dynamic interfaces for interaction by an analyst. The further modified alert to be displayed by the reporting interface, in combination with the system interface, may be displayed in addition to other further modified alerts that have been received by a dynamic interface. The analyst may interact with each further modified alert for analysis of the alert using additional information provided by the system and/or to select a course of action (which may also be included in the further modified alert). The interaction with the further modified alert may be received by an interface (e.g., a network interface and/or the analyst interface). The information received by the interface may be provided to the knowledge store via the storage logic. The information stored in the knowledge store is used by the predictive machine learning logic to generate a predictive model to implement a self-learning feedback loop. The self-learning feedback loop aids an analyst in efficiently addressing cyber-security alerts received by a cyber-security automated analyst alerting system.

(19) Elements of the invention employ computerized techniques to generate machine learning models used to classify received alerts, initiate the display of classified received alerts, and re-generate the machine learning models in response to input receive from a cyber-security analyst responsive to the displayed classified received alert.

I. Terminology

(20) In the following description, certain terminology is used to describe features of the invention. For example, in certain situations, both terms “logic” and “engine” are representative of hardware, firmware and/or software that is configured to perform one or more functions. As hardware, logic (or engine) may include circuitry having data processing or storage functionality. Examples of such

circuitry may include, but is not limited or restricted to a microprocessor, one or more processor cores, a programmable gate array, a microcontroller, an application specific integrated circuit, wireless receiver, transmitter and/or transceiver circuitry, semiconductor memory, or combinatorial logic.

(21) Logic (or engine) may be software in the form of one or more software modules, such as executable code in the form of an executable application, an application programming interface (API), a subroutine, a function, a procedure, an applet, a servlet, a routine, source code, object code, a shared library/dynamic load library, or one or more instructions. These software modules may be stored in any type of a suitable non-transitory storage medium, or transitory storage medium (e.g., electrical, optical, acoustical or other form of propagated signals such as carrier waves, infrared signals, or digital signals). Examples of non-transitory storage medium may include, but are not limited or restricted to a programmable circuit; a semiconductor memory; non-persistent storage such as volatile memory (e.g., any type of random access memory “RAM”); persistent storage such as non-volatile memory (e.g., read-only memory “ROM”, power-backed RAM, flash memory, phase-change memory, etc.), a solid-state drive, hard disk drive, an optical disc drive, or a portable memory device. As firmware, the executable code is stored in persistent storage. The term “computerized” generally represents that any corresponding operations are conducted by hardware in combination with software and/or firmware.

(22) The term “transmission medium” (or “transmission media”) may refer to a communication path between two or more systems (e.g. any electronic devices with data processing functionality such as, for example, a security appliance, server, mainframe, computer, netbook, tablet, smart phone, router, switch, bridge or router). The communication path may include wired and/or wireless segments. Examples of wired and/or wireless segments include electrical wiring, optical fiber, cable, bus trace, or a wireless channel using infrared, radio frequency (RF), or any other wired/wireless signaling mechanism.

(23) The term “alert” may refer to a signal or notification (e.g., report) received from, or issued by, a source. The alert conveys information regarding an event. An event may refer to an observed (or in some cases, inferred) occurrence that has significance to an associated alert type. An alert type may indicate an alert classification (e.g., an alert indicating a user login attempt may be classified as a “user alert”—i.e. an alert with a “user” type). A cyber-security event may be relevant to a cyber-threat. Relationships between events may be determined based on information provided by received cyber-security alerts describing events monitored by the cyber-security devices (or software). For example, a user-operated endpoint may be monitored by resident cyber-security software (e.g., an embedded agent), the software monitoring the execution of a process “opening” a file. An alert may be associated with, or triggered by, any of a variety of computing activities, for example: a granting or denial of administrative rights or escalation of privileges, an unauthorized access of an access-restricted compute device, detection of a new device on a restricted network, multiple different user login(s) made by a single compute device, an unexpected/unusual login of a user, detection of an internal vulnerability, etc.

(24) The term “message” generally refers to signaling (wired or wireless) as either information placed in a prescribed format and transmitted in accordance with a suitable delivery protocol or information made accessible through a logical data structure such as an API. Hence, each message may be in the form of one or more packets, frame, or any other series of bits having the prescribed, structured format.

(25) The term “object” generally refers to a collection of data, such as a group of related packets associated with a request-response message pairing for example, normally having a logical structure or organization that enables classification for purposes of analysis. For instance, an object may be a self-contained element, where different types of such objects may include an executable file, non-executable file (such as a document or a dynamically link library), a Portable Document Format (PDF) file, a JavaScript file, Zip file, a Flash file, a document (for example, a Microsoft

Office® document), an electronic mail (email), downloaded web page, an instant messaging element in accordance with Session Initiation Protocol (SIP) or another messaging protocol, or the like.

(26) The term “appliance” refers to any type of general-purpose or special-purpose computer, including a dedicated computing device, adapted to implement any variety of existing, or future, software architectures relating to detection of, and protection from, cyberattack and related functionality. The term appliance should therefore be taken broadly to include such arrangements, in addition to any systems or subsystems configured to support such functionality, whether implemented in one or more network computing devices or other electronic devices, equipment, systems or subsystems.

(27) The terms “computer”, “processor”, “computer processor”, “compute device”, or the like should be expansively construed to cover any kind of electronic device with data processing capabilities including, by way of non-limiting example, a digital signal processor (DSP), a microcontroller, a field programmable gate array (FPGA), an application specific integrated circuit (ASIC), a graphics processing unit (GPU), or any other electronic computing device comprising one or more processors of any kind, or any combination thereof.

(28) As used herein, the phrase “for example,” “such as”, “for instance”, and variants thereof describe non-limiting embodiments of the presently disclosed subject matter. Reference in the specification to “one case”, “some cases”, “other cases”, or variants thereof means that a particular feature, structure or characteristic described in connection with the embodiment(s) is included in at least one embodiment of the presently disclosed subject matter. Thus the appearance of the phrase “one case”, “some cases”, “other cases” or variants thereof does not necessarily refer to the same embodiment(s).

(29) It is appreciated that, unless specifically stated otherwise, certain features of the presently disclosed subject matter, which are, for clarity, described in the context of separate embodiments, may also be provided in combination in a single embodiment. Conversely, various features of the presently disclosed subject matter, which are, for brevity, described in the context of a single embodiment, may also be provided separately or in any suitable sub-combination.

(30) Lastly, the terms “or” and “and/or” as used herein are to be interpreted as inclusive or meaning any one or any combination. Therefore, “A, B or C” or “A, B and/or C” mean “any of the following: A; B; C; A and B; A and C; B and C; A, B and C.” An exception to this definition will occur only when a combination of elements, functions, steps or acts are in some way inherently mutually exclusive.

II. Architecture

(31) Referring to FIG. **1**, an exemplary block diagram of an automated analyst alerting system **100** is communicatively coupled, via a network interface **110**, to at least one communication network **105**. The communication network **105** may couple the automated analyst alerting system **100** with cyber-security devices **102** and/or a remote analyst console **197** via transmission media to exchange information with the communication network directly or via the Internet. The communication network **105** may be coupled directly or indirectly to cyber-security device(s) **102**. The cyber-security devices **102** may operate within the same or different networks. Each cyber-security device represents a logical entity, operating on objects, to determine if they represent a cyber-security risk. In some embodiments a cyber-security device **102** may include a software application operating on a user operated endpoint device (e.g., a laptop, mobile phone, etc.) while in some other embodiments the cyber-security device may include a dedicated cyber-security appliance. The cyber-security device **102** may detect potential cyber-security threats and generate and issue a cyber-security alert. The cyber-security device **102** may be configured to direct issued alerts to the automated analyst alerting system **100**.

(32) The automated analyst alerting system **100** includes a network interface **110**, an alert parser **120**, an alert recommendation engine **130**, a knowledge store **140**, a predictive model generation

logic **150**, a presentation logic **160**, a mitigation logic **170**, a storage logic **180** a reporting engine **190** and an analyst interface **195**. Upon receipt by the automated analyst alerting system **100** of an alert generated by a cyber-security device **102**, via the network interface **110**, the alert is provided to the alert parser **120**. The alert parser **120** analyzes the received alert and normalizes the contents according to a set of normalization rules that normalize the received alert into a known alert format, comprehensible by the alert recommendation engine **130**. In some embodiments the normalization rules may be user-defined (and/or user-modifiable). In some embodiments the alert parser may be updated with additional (modified) processing (normalizing) rules. Such updates may be received by the automated analyst alerting system **100** periodically or aperiodically via the network interface **110**. The rule update may be processed by the alert parser **120** directly or via a separate logic (not shown).

(33) The alert parser **120** provides the normalized alert to the alert recommendation engine **130** for further analysis. In some embodiments, the alert parser **120** may, limit further analysis of an alert based on contextual information. If a received alert received by the alert parser **120** includes a classification of the alert the alert parser may provide the alert recommendation engine **130** with the received alert classification and the alert recommendation engine **130** may include this classification (in some embodiments this classification may be added to the modified received alert without a confidence level). Contextual information may include data with respect to available system resources (e.g., processor load, memory availability, etc.), quality of alerts from particular cyber-security devices **120** (e.g., information related to reliability of cyber-security alerts in identifying cyberthreats associated with a particular cyber-security device), duplication (i.e. information that associates a set of alerts identifying identical alerts from cyber-security devices and associates them together for de-duplication by the various logics of the automated analyst alerting system), etc. Analysis of contextual information may be performed by the alert parser **120** by evaluating normalization rules by the alert parser **120**. By reducing the number of received alerts to be analyzed by the automated analyst alerting system **100**, the system may aid an analyst focus on high value alerts.

(34) The alert recommendation engine **130** includes at least an alert analysis and labeling engine **132** and an action generator **134**. The alert recommendation engine **130** receives, from the alert parser **120**, an alert transformed according to the normalization rules and via the alert analysis and labeling engine **132**, generates at least one label associated with the alert as well as a confidence level associated with each label. The action generator **134** of the alert recommendation engine **130** receives the label and associated confidence levels from the alert analysis and labeling engine **132** and determines if an action may be associated with the alert. The components of the alert recommendation engine **130** (i.e. the alert analysis and labeling engine **132** and the action generator **134**) operate in concert with information provided by the knowledge store **140**.

(35) The knowledge store **140**, operating in concert with the alert recommendation engine **130**, provides information generated from the predictive model generation logic **150** and information received from analyst operation. The information provided to the knowledge store **140** may include, by non-limiting example, information associated with execution of mitigations by cyber-security device(s) **102**, information associated with the result of instructed mitigations by cyber-security device(s), classification of a received alert by an analyst, etc. Additionally, in some embodiments, the knowledge store **140** may include the predictive model generated by the predictive model generation logic **150**. In some embodiments the predictive model may be stored in a separate store (e.g., a store provided by the alert recommendation engine **130**, etc.). In some embodiments, the knowledge store **140** may receive and store information, from the analyst, associated with a classification of a received alert (e.g., related alerts, identifiers associated with the alert, intelligence associated with a received alert, etc.).

(36) The predictive model generation logic **150** may periodically or aperiodically generate a predictive model to be used by the alert recommendation engine **130**. The predictive model

generation logic may generate the predictive model in response to the receipt of a signal indicating new information has been stored in the knowledge store **140**. In some embodiments, the predictive model generation logic **150** may only generate a new model in response to the receipt by the knowledge store **140** of information received from an analyst (e.g., a new alert classification, a modification and/or update to an existing classification, correction of a previously mis-classified alert, etc.). The predictive model generated by the predictive machine learning model **150** may be generated according to a known machine learning recommendation techniques. In some embodiments the predictive machine learning logic **150** may train a predictive model based on the labelled data stored in the knowledge store **140**. In some embodiments, the predictive machine learning logic **150** may generate the predictive machine learning model "offline" (i.e., "out of band"). In some embodiments (not shown) the predictive machine learning logic **150** may be remotely located from the automated analyst alert system **100** and communicatively coupled, for example, via communication network **105**, with cloud computing resources (not shown). The generated predictive model generates at least one classification and/or association of the classification with an alert. In some embodiments the classification generated by the predictive model may correspond to a numerical association with the classification. For example, based on analysis of the alert by the predictive model generated by the predictive model generation logic **150**, an alert may be associated with (a) maliciousness (31% confidence level), (b) phishing (51% confidence level), and (c) benign (67% confidence level).

(37) In some embodiments, the predictive model generation logic **150** may generate a predictive model associating mitigation actions ("actions") with identified classifications. In other embodiments, a separate logic (e.g., the action generator **134**) may determine an action associated with identified classifications. A set of known actions may be stored in the knowledge store **140**. In some embodiments, the analyst may generate (i.e. user-define) an action to be stored in the knowledge store. In certain embodiments, actions generated by an analyst, in response to an alert are automatically stored in the knowledge store **140**.

(38) The alert analysis and labeling engine **132** receives from the alert parser **120** the received alert for further analysis and obtains a predictive model from the knowledge store **140**. The alert analysis and labeling engine **132** is configured to apply the obtained predictive model and apply the predictive model to the received alert. By applying the predictive model to the received alert the alert analysis and labeling engine **132** generates at least one classification label and confidence level. If a plurality of classification labels and levels of association of classifications are generated, the alert analysis and labeling engine **132** will determine a classification for the received alert. In some embodiments the alert analysis and labeling engine **132** may apply more than one classification to an alert. In some embodiments the classification determination of the alert analysis and labeling engine **132** may, by way of non-limiting example, include the classification corresponding to the highest confidence level, each classification where an associated level of classification exceeds a threshold, a classification associated with a level of classification exceeding a second threshold, higher than a first threshold, etc. In some embodiments the alert analysis and labeling engine **132** may provide the classification of the alert and the alert to the action generator **134** while in other embodiments, the alert analysis and labeling engine may provide the classification and the received alert directly to the presentation logic **160**.

(39) The action generator **134** is configured to analyze the received alerts and classification to determine if a known action may be recommended to a receiving analyst. In some embodiments, the predictive model generation logic **150** may generate a predictive action model, stored in the knowledge store **140**. The predictive action model is adapted to, in combination with the action generator **134**, associate a known action with a received alert. In other embodiments the action generator may be configured with a set of rules associating specified actions with selected alerts. For example, an alert received and classified by the alert analysis and labeling engine **132** as being associated with "phishing" may cause the action generator **134** to associate an action to the alert,

the action, upon execution, quarantines the cyber-security device **102** associated with the alert (i.e. the computing device associated with the phishing alert). Rules to be processed by the action generator **134** may be factor-set, and/or user (e.g., security administrator, analyst, etc.) configurable. The action generator may rely on information processed by the alert parser **120** associated with affected devices protected by the automated analyst alerting system **100**. In some embodiments the action generator **134** may identify an action associated with the alert to be automatically executed (e.g., not require execution approval from analyst). The action generator **134** may determine that no known (e.g., in the knowledge store **140**, and/or in configured rules of the action generator) action may be associated with the received alert and classification. Once an alert is analyzed by the action generator **134**, the alert is provided to the presentation logic **160**.

(40) The presentation logic **160** receives, from the alert recommendation engine **130**, the received alert and associated classifications and actions. The presentation logic **160** determines if an associated action should be provided directly to the mitigation logic **170** or be presented to an analyst for determination. The presentation logic **160** may be configured to determine if and how an alert should be presented to an analyst by the reporting engine **190**. The presentation logic **160** may determine an alert whose associated action is to be automatically executed by the mitigation logic **170** should be presented to the analyst despite its automated execution. In some embodiments the presentation logic **160** may generate a graphical user interface (GUI) for the reporting engine **190** to present to the analyst. The presentation logic **160** may receive results associated with the execution of an action by the mitigation logic **170** and/or instructions received from the analyst related to alerts that were presented to the analyst. The presentation logic **160** provides the received alert and associated results and/or analyst instruction to the storage logic **180**.

(41) The storage logic **180** determines if a received action, alert classification, or analyst instruction (e.g., action instruction, creation of a new action, etc.) should be stored in the knowledge store **140**. The determination as to whether an action should be stored in the knowledge store **140** may be based on whether the action is duplicative (e.g., a similar action exists in the knowledge store), not in opposition to existing actions stored in the knowledge store, etc. In some embodiments, a modification to an existing action may be received by the storage logic **180** and the contents of the knowledge store **140** may be modified. If no action needs to be stored in the knowledge store **140** or if it has already been stored in the knowledge store, the received information is provided to the reporting engine **190** for presentation to the analyst.

(42) The mitigation logic **170** receives from the presentation logic **160** actions for execution by cyber-security device(s) **102**. The action generator **134** may identify, to the presentation logic **160** whether an action associated with an alert should be automatically executed by the mitigation logic. Similarly, the mitigation logic **170** may receive, via the network interface(s) **110**, an action instruction from an analyst (e.g. via the analyst interface **195**). The action instructed by the analyst to the mitigation logic **170** may be provided to the presentation logic **160** for further processing (as described above) and be further processed by the mitigation logic **170** for execution. The execution of actions by the mitigation logic **170** may be direct (e.g., an action which may be executed directly by the automated analyst system **100**) or indirect (e.g., issuing instructions, via the network interface(s) **110** to cyber-security device(s) **102**). In some embodiments the mitigation logic **170** may be configured with credentials for interaction with systems requiring authorization for executing cyber-security actions. The mitigation logic **170** may be configured to generate application programming interface (API) calls to cyber-security device(s) **102** in response to receiving an action for execution. In other embodiments an action may include the execution details and the mitigation logic **170** does not generate API calls to the cyber-security device(s) **102**. The result of an execution is received by the mitigation logic **170** via the network interface(s) **110** and provided to the presentation logic **160**.

(43) The reporting engine **190** may be configured to generate an alert for transmission to an external display of an analyst. The reporting engine **190** may be configured to provide a GUI to the

analyst display and/or other known display systems (e.g., command line terminal, etc.). The reporting engine **190** is configured to provide reports via the network interface(s) **110**, for example, the remote analyst console **197**. In some embodiments the reporting engine **190** may provide interactive alert which may allow an analyst to provide responsive instructions to the mitigation logic **170** for further processing by the automated analyst alerting system **100**. The analyst may provide an interactive response and consume alerts via the remote analyst console **197**.

(44) As illustrated in FIG. **2** in greater detail, the automated analyst recommendation system **200** has physical hardware including hardware processors **210**, network interface(s) **220**, a memory **230**, a system interconnect **270**, and optionally, a user interface **290**. The memory **230** may contain software comprising an alert parser **240**, an alert analysis and labeling engine **242**, an action generator **244**, presentation logic **250**, a mitigation logic **252**, a reporting engine **254**, an storage logic **260**, and a predictive model generation logic **265**. The physical hardware (e.g. hardware processors **210**, network interface(s) **220**, memory **230**) may be connected for communication by the system interconnect **270**, such as a bus. Generally speaking, an automated analyst recommendation system **200** is a network-connected alert analysis system configured to enhance the operation of a security operations center (SOC) by providing a SOC analyst with relevant alerts and meta-information.

(45) The hardware processor **210** is a multipurpose, programmable device that accepts digital data as input, processes the input data according to instructions stored in its memory, and provides results as output. One example of the hardware processor **210** is an Intel® microprocessor with its associated instruction set architecture, which is used as a central processing unit (CPU) of the automated analyst recommendation system **200**. Alternatively, the hardware processor **210** may include another type of CPU, a digital signal processor (DSP), an application specific integrated circuit (ASIC), or the like.

(46) The network device(s) **280** may include various input/output (I/O) or peripheral devices, such as a storage device, for example. One type of storage device may include a solid state drive (SSD) embodied as a flash storage device or other non-volatile, solid-state electronic device (e.g., drives based on storage class memory components). Another type of storage device may include a hard disk drive (HDD). Each network device **280** may include one or more network ports containing the mechanical, electrical and/or signaling circuitry needed to connect the automated analyst recommendation system **200** to the private network **120** to thereby facilitate communications over the communication network **105**. To that end, the network interface(s) **220** may be configured to transmit and/or receive messages using a variety of communication protocols including, inter alia, TCP/IP and HTTPS.

(47) The memory **230** may include a plurality of locations that are addressable by the hardware processor **210** and the network interface(s) **220** for storing software (including software applications) and data structures associated with such software. The hardware processor **210** is adapted to manipulate the stored data structures as well as execute the stored software, which includes an alert parser **240**, an alert analysis and labeling engine **242**, an action generator **244**, presentation logic **250**, an mitigation logic **252**, a reporting engine **254**, an storage logic **260**, and a predictive model generation logic **265**.

(48) The alert parser **240** is a software application, operating on data (i.e. alerts) provided to the automated analyst recommendation system **200** via the network interface(s) **220** according to the description of alert parser **120** of FIG. **1**. The alert parser **240** receives an alert and processes the alert according a set of normalization rules residing within the memory **230**. The alerts processed by the alert parser **240** are provided to the alert analysis and labeling engine **242** for further processing.

(49) The alert analysis and labeling engine **242** processes received alerts according to a generated predictive model stored in memory **230**. The alert analysis and labeling engine generates a set of classifications in response to the processing of the received alert by the predictive model. The

classifications may correspond to a set of labels applied to the received alert and to be used in further processing of the alert by other components of the automated analyst recommendation system **200**. The classification labels generated by the alert analysis and labeling engine **242** may include a likelihood of association (i.e. confidence level) with the alert. The likelihood of association may be applied to the alert and provided, in addition to the associated classification label and alert, to the action generator **244**. In some embodiments the alert analysis and labeling engine **242** may also generate a set of associated alerts related to the received alert. The association may result from the predictive model and/or be associated with correlating meta-information of the alert. The predictive model is generated by the predictive model generation logic **265**.

(50) The predictive model generation logic **265** generates predictive models and stores in the memory **230**. In some embodiments the predictive model generation logic **265** may generate a separate second predictive action model (based on the actions previously associated with alerts and stored in the knowledge store **140**) for use by the action generator **244**, distinct and trained separately from the predictive model used by the alert analysis and labeling engine **242** (based on prior classifications of alerts and stored in the knowledge store **140**). In other embodiments the predictive model generation logic may associate prior analyzed alerts with the received alert to determine if they are related and may need to be processed by the analyst together. If so, they may be associated together in meta-information and provided to the presentation logic **250**. The predictive model generation logic **265** generates models based on information stored in memory **230** related to prior alerts and actions. The predictive model generation logic **265** analyzes stored information to generate a predictive model according to known machine learning techniques. A random forest classifier is an exemplary technique that creates a set of decision trees from randomly selected subset of training set. The random forest classifier then aggregates the decisions from the set of decision trees to decide the final classification associated with the targeted alert. In some embodiments an alternative technique may be used (e.g., convolutional neural networks, support vector machines, etc.). The generated predictive models are stored in memory **230** to be accessed by the analytic logics of the automated analyst recommendation system **200**.

(51) The action generator **244** receives from the alert analysis and labeling engine **242** the received alert and at least the classification label(s) determined by the alert analysis and labeling engine. The action generator **244** analyzes the received alert and classification and may determine an action which may be executed in response to the alert. The determined action may be an action recommended (to the analyst) to mitigate the cyber-security threat identified by the alert. In some embodiments the determined action may include instructions to obtain additional information regarding the alert (e.g., an instruction to the alert originating cyber-security device for additional meta-information related to the first alert). The action generator **244** may generate an action based on rules stored in memory **230** and/or based on model provided by the predictive model generation logic **265**. The predictive model generation logic **265** may generation a predictive action model in response to storage in memory **230** of new actions. New actions may be stored in memory **230** based on an update action received by the automated analyst recommendation engine via the network interface(s) **220** and/or via analyst selecting a recommended action or submitting an action. The predictive action model is generated based on actions stored in memory **230**. The action generator **244** may associate no actions or one or more actions in response to further analysis of the received alert and/or classification information (the classification information including the at least classification label and associated likelihood of association). In some embodiments the action generator **244** determines that a recommended action shall be executed without confirmation by the analyst and the action is labelled with such an indicator. Once the action generator **244** determines whether an action may be associated with the alert, the alert and any associated information is provided to the presentation logic **250**.

(52) The presentation logic **250** is provided with the alert from either the alert analysis and labeling engine **242** or the action generator **244** as well as with additional meta-information (e.g.,

recommended action(s), classification(s) and associated confidence levels) generated during prior processing for presentation to the analyst. If an action is labeled for automatic execution the action is provided to the mitigation logic **252** by the presentation logic **250**. Similarly, if responsive to presentation to an analyst, the presentation logic **250** receives instructions from the analyst, the action instructed is provided to the mitigation logic **252** for processing. The presentation logic **250** may further analyze the alert and associated meta-information to determine a priority and arrangement of the alert and associated information to the analyst. For example, alerts associated with low confidence levels (e.g., the system cannot properly label the alert), may be assigned a higher priority and presented to the analyst. In some other embodiments, analysis of the meta-information associated with an alert may indicate duplicative alerts having been received, consequently, the presentation logic may generate a modified GUI to aggregate and/or filter the duplicative alerts to the analyst. In still yet other embodiments the presentation logic **250** may receive from the mitigation logic **252** the results of an executed action for presentation to the analyst and storage by the action logic **260**. Upon receipt, the execution results are associated with the associated alert's meta-information and provided to storage logic **260**.

(53) The mitigation logic **252** receives action instructions via the presentation logic **250**. Actions may be provided to the mitigation logic **252** automatically or in response to an instruction from an analyst. The action may require communication via the network interface(s) **220** to third party systems (e.g., cyber-security devices **102**). Communication with third party systems may require authentication credentials for authorization, which may be configured by the security administrator and/or an analyst in advance of action execution or as needed. The mitigation logic **252** may also operate via the analyst alert recommendation system **200** directly. An action execution result may be generated upon receipt of results from an execution. In some embodiments, if no result response is received within a specified time period (e.g., 60 seconds) the mitigation logic may generate an action execution result indicating a “timeout”. The results response is provided to the storage logic **260** via the presentation logic **250**.

(54) The storage logic **260** processes the received alert and meta-information (including results information provided by the mitigation logic **252**. The storage logic **260** analyzes the alert and associated meta-information and determines if the action and/or classifications may be stored in memory **230**. The determination, as to whether or not the meta-information may be stored in memory **230**, may be based on the duplicative nature of the meta-information (i.e. determine if the same information is stored in the memory), modification of existing meta-information stored in the memory and/or if the meta-information to be stored is inconsistent with prior stored meta-information.

(55) The reporting engine **254** receives the alert and associated meta-information for presentation to the analyst. The reporting engine may provide the alert and associated meta-information to the user interface **290** and/or to the network device(s) **220** for presentation to the analyst. The user interface **290** may produce a graphical or textual based representation to a user of the endpoint **10** device **200**. The user interface **290** provides the user with the ability to interact with the computer. The user interface **290** may not be present for an endpoint device that is not dedicated to a single user or does not require the interaction with a user. The user interface **290** may receive input via the network device(s) **280** which include various input/output devices.

(56) FIG. **3** represents an exemplary flowchart of a computerized method **300** for operating an automated analyst recommendation system **100**. The exemplary method **300** starts at step **305** and proceeds to step **310** where the automated analyst recommendation system **100** receives an alert from cyber-security device(s) **102** transmitted over the communication network **105** via the network interface(s) **110**. During step **310**, the alert parser **120** processes the alert to generate processible meta-information for further analysis by subsequent analytics logics (e.g., the alert analysis and labeling engine **132**, the action generator **134**, etc.). Upon completion of processing by the alert parser **120**, the alert and associated meta-information is provided to alert analysis and

labeling engine **132** for further analysis in step **315**.

(57) The alert analysis and labeling engine **132**, during step **315** applies the predictive machine learning model stored in the knowledge store **140**, to the received alert and associated meta-information. The results of the analysis of the received alert and meta-information with the predictive model is at least one classification label and a confidence level (e.g., likelihood of association, etc.). In some embodiments the predictive model may also generate a set of alerts associated as meta-information with the received alert. The associated alerts may be relevant to the assessment of the received alert when reviewed by an analyst. If associated alerts are identified, the alert and associated meta-information is retrieved from the knowledge store **140** and added to the received alert's associated meta-information for further processing in step **320**. In step **325** the meta-information and the received alert are associated and provided to the action generator **134** for further analysis.

(58) In step **330** the action generator **134** receives the alert received by the automated analyst recommendation system **100** and associated meta-information for analysis. The analysis may include the processing of factory-set and/or user-defined rules. For example, an alert associated with a “phishing” email cyberattack may identify the source cyber-security device(s) **102** from the meta-information and generate an action targeting the phishing email for quarantine. In some embodiments a predictive action model may be applied to the received alert and meta-information to generate a set of recommended actions based on prior actions taken and/or recorded by the automated analyst recommendation system **100**. If an action is generated in step **335**, the method continues step **340** where the alert and its associated meta-information is modified with the generated action(s). Further processing by the action generator **134** may further determine if at least one of the generated action(s) should be automatically processed by the mitigation logic **170** in step **345**. If the generated action is determined to be automatically run in step **350**, and succeeds, the method ends at step **390**. If the generated action is determined to be manually run in step **345**, the generated action is presented to the analyst via the analyst interface **195** in step **348**. Upon selection by the analyst, the analyst interface **195** provides the selection to the mitigation logic for execution and if in step **350** the executed action succeeds, the method ends at step **390**.

(59) If the action generator cannot identify a recommendable action in step **335** or the executed action fails in step **350**, the presentation logic **160** determines a priority for presentation of the alert to the analyst in step **360**. The determination of priority is based, at least in part, on the success of an action executed by the mitigation logic **170**. In some embodiments, the priority for presentation of an alert to the analyst may be based on the confidence level associated with a classification of the alert. In some embodiments, the presentation logic **160** determines a priority level of an alert in step **360** then in step **365** determines if the alert, based in part on the priority level, should be presented to the analyst. If the presentation logic **160** determines that the alert need not be presented to the analyst in step **370**, the method ends at step **390**.

(60) If the alert is determined to be presented to the analyst in step **370** by the presentation logic **160**, the alert is presented to the analyst for further interaction. In some embodiments the further interaction with the analyst may be done through a user interface **290** or via the reporting engine **190** once the alert has been further processed by the storage logic **180**. In step **375** the analyst is presented with the modified alert. The analyst may select an action associated with the modified alert, modify a classification of the modified alert, and/or generate an action or classification associated with the alert based on the context received. The result of step **375** is provided to the knowledge store via the storage logic **180** in step **380**. In step **380**, upon receipt of a new and/or modified alert and/or action result, the storage logic **180** may store the received information in the knowledge store **140**. If information received by the knowledge store in step **380**, the alert analysis and labeling engine **132** may regenerate a new predictive model based on the new information and re-analyze the received alert in step **385**. By this method, the system will identify alerts requiring additional action by an analyst while minimizing the time spent by analysts on low value alerts.

(61) The foregoing description has been directed to specific embodiments. It will be apparent, however, that other variations and modifications may be made to the described embodiments, with the attainment of some or all their advantages. For instance, it is expressly contemplated that the components and/or elements described herein can be implemented as software encoded on a tangible (non-transitory) computer-readable medium (e.g., disks, electronic memory, and/or CDs) having program instructions executing on a computer, hardware, firmware, or a combination thereof. Moreover, the embodiments or aspects thereof can be implemented in hardware, firmware, software, or a combination thereof. In the foregoing description, for example, in certain situations, terms such as “engine,” “component” and “logic” are representative of hardware, firmware and/or software that is configured to perform one or more functions. As hardware, engine (or component/logic) may include circuitry having data processing or storage functionality. Examples of such circuitry may include, but is not limited or restricted to a microprocessor, one or more processor cores, a programmable gate array, a microcontroller, an application specific integrated circuit, semiconductor memory, or combinatorial logic. Accordingly, this description is to be taken only by way of example and not to otherwise limit the scope of the embodiments herein. Therefore, it is the object of the appended claims to cover all such variations and modifications as come within the true spirit and scope of the invention.

Claims

1. A computer-implemented method to perform self-learning for a predictive machine learning model of a cyber-security alert system, the method comprising: obtaining, by a computing system, the predictive machine learning model, the predictive machine learning model trained based on data in a knowledge store; receiving, by the computing system, an alert associated with a monitored network; classifying, by the computing system, the received alert according to the predictive machine learning model to generate at least one alert classification; automatically generating, by the computing system, a one or more recommended actions responsive to, and associated with, the received alert based on the alert classification; automatically causing, by the computing system, execution of the one or more recommended actions; updating, by the computing system, the knowledge store to include a result of the one or more recommended actions in the knowledge store; and generating an updated predictive machine learning model based on the updated knowledge store.
2. The computer-implemented method of claim 1, wherein the result indicates a success or failure of the one or more recommended actions.
3. The computer-implemented method of claim 1, further comprising surfacing, by the computing system, the at least one alert classification or the one or more recommended actions for modification by an analyst.
4. The computer-implemented method of claim 3, wherein said surfacing is performed in response to the result of the one or more recommended actions indicating a failure of the one or more recommended actions.
5. The computer-implemented method of claim 3, further comprising, prior to generating the updated predictive machine learning model: updating the knowledge store based on a modification entered by the analyst.
6. The computer-implemented method of claim 1, wherein the predictive machine learning model comprises an artificial neural network.
7. The computer-implemented method of claim 1, wherein the predictive machine learning model generates a confidence score for the at least one alert classification and wherein automatically causing, by the computing system, execution of the one or more recommended actions occurs in response to the confidence score exceeding a confidence threshold.
8. The computer-implemented method of claim 1, wherein the one or more recommended actions

comprise communication with a conventional external computing device that is configured to effectuate the one or more recommended actions.

9. The computer-implemented method of claim 8, wherein said communication occurs via an Application Programming Interface (API) call.

10. A computing system configured to perform a self-learning loop for a predictive machine learning model of a cyber-security alert system, the computing system comprising: one or more processors; and one or more non-transitory computer-readable media that collectively store: a knowledge store; a predictive machine learning model; and instructions that, when executed by the one or more processors cause the computing system to perform operations, the operations comprising: obtaining, by the computing system, the predictive machine learning model, the predictive machine learning model trained based on data in the knowledge store; receiving, by the computing system, an alert associated with a monitored network; classifying, by the computing system, the received alert according to the predictive machine learning model to generate at least one alert classification; automatically generating, by the computing system, a one or more recommended actions responsive to, and associated with, the received alert based on the alert classification; automatically causing, by the computing system, execution of the one or more recommended actions; updating, by the computing system, the knowledge store to include a result of the one or more recommended actions in the knowledge store; and generating an updated predictive machine learning model based on the updated knowledge store.

11. The computing system of claim 10, wherein the result indicates a success or failure of the one or more recommended actions.

12. The computing system of claim 10, further comprising surfacing, by the computing system, the at least one alert classification or the one or more recommended actions for modification by an analyst.

13. The computing system of claim 12, wherein said surfacing is performed in response to the result of the one or more recommended actions indicating a failure of the one or more recommended actions.

14. The computing system of claim 12, further comprising, prior to generating the updated predictive machine learning model: updating the knowledge store based on a modification entered by the analyst.

15. The computing system of claim 10, wherein the predictive machine learning model comprises an artificial neural network.

16. The computing system of claim 10, wherein the predictive machine learning model generates a confidence score for the at least one alert classification and wherein automatically causing, by the computing system, execution of the one or more recommended actions occurs in response to the confidence score exceeding a confidence threshold.

17. The computing system of claim 10, wherein the one or more recommended actions comprise communication with a conventional external computing device that is configured to effectuate the one or more recommended actions.

18. The computing system of claim 17, wherein said communication occurs via an Application Programming Interface (API) call.

19. The computing system of claim 10, wherein the operations further comprise employing the updated predictive machine learning model to process a new alert.

20. One or more non-transitory computer-readable media that collectively store: a knowledge store; a predictive machine learning model; and instructions that, when executed by one or more processors of a computing system cause the computing system to perform operations, the operations comprising: obtaining, by the computing system, the predictive machine learning model, the predictive machine learning model trained based on data in the knowledge store; receiving, by the computing system, an alert associated with a monitored network; classifying, by the computing system, the received alert to generate at least one alert classification; automatically generating, by

the computing system and using the predictive machine learning model, one or more recommended actions responsive to, and associated with, the received alert based on the alert classification; automatically causing, by the computing system, execution of the one or more recommended actions; updating, by the computing system, the knowledge store to include a result of the one or more recommended actions in the knowledge store; and generating an updated predictive machine learning model based on the updated knowledge store.