(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
07 May 2020 (07.05.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/091591 A1

(51) International Patent Classification:
G06F 21/57 (2013.01) *G06F 9/455* (2006.01)

(21) International Application Number:
PCT/MY2019/050078

(22) International Filing Date:
16 October 2019 (16.10.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PI 2018001830 30 October 2018 (30.10.2018) MY

(71) Applicant: **MIMOS BERHAD** [MY/MY]; Technology Park Malaysia, Kuala Lumpur, 57000 (MY).

(72) Inventors: **GOORTANI, Ehsan Mostajeran**; c/o Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **ISMAIL, Bukhary Ikhwan Bin**; c/o Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **MOHD MYDIN, Mohd Nizam Bin**; c/o Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **KANDAN, Rajendar**; c/o Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **KHALID, Mohammad Fairus Bin**; c/o Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY).

(74) Agent: **MIRANDAH ASIA (MALAYSIA) SDN BHD**; Suite 3B-19-3, Plaza Sentral, Jalan Stesen Sentral 5, Kuala Lumpur, 50470 (MY).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP,

(54) Title: A SYSTEM AND METHOD FOR ENABLING VULNERABILITY DETECTION OF CLOUD CONTAINER BASED SERVICE DEPLOYMENT

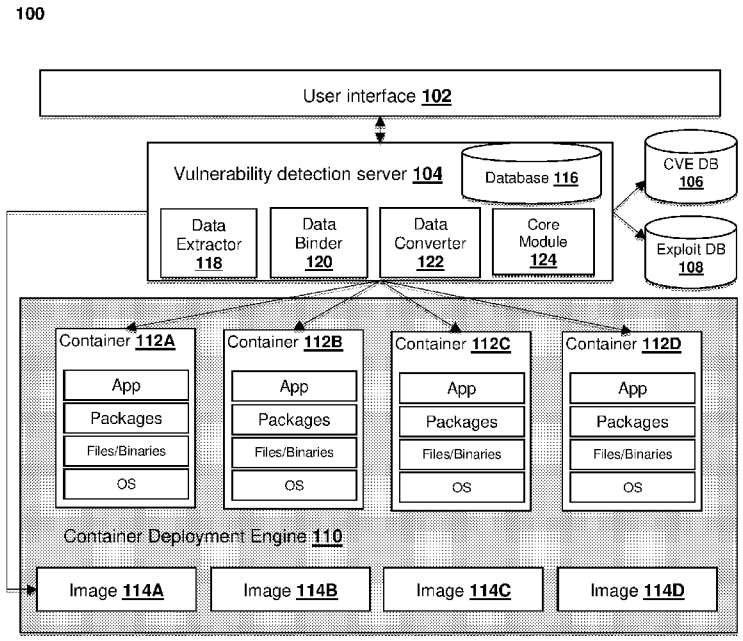

FIG. 1.0

(57) Abstract: The present invention relates to a system and method for enabling vulnerability detection of cloud container based service. In particular, the present invention relates to a system and method that enables efficient and intelligent flow of vulnerabilities detection utilizing at least one vulnerability detection server (104) comprising at least one Data Extractor (118) for generating vectorized document of each container, at least one Data Binder (120) for generating vectorized document of each common vulnerabilities and exposures, at least one Data Converter (122) for converting vectorized documents of each container and vectorized documents of each common vulnerabilities of exposures and at least on Core module (124) for relevancy comparison. The present invention provides time to time prediction of vulnerabilities by utilizing machine learning instructions. The present invention allows vulnerabilities detection during dynamic runtime stage and static image.

WO 2020/091591 A1

[Continued on next page]

KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report (Art. 21(3))*

A SYSTEM AND METHOD FOR ENABLING VULNERABILITY DETECTION OF CLOUD CONTAINER BASED SERVICE DEPLOYMENT

FIELD OF INVENTION

The present invention relates to a system and method for enabling vulnerability detection of cloud container based service deployment. In particular, the present invention provides time to time prediction of vulnerabilities through machine learning.

BACKGROUND OF INVENTION

A cloud container is designed to virtualize a single application which creates an isolation boundary at the application level. Container is easier for scalable rapid deployment as it consists of independent subsystem of network, memory and file system. However, as the container is becoming more popular the traditional virtual machine, new challenges of security control are introduced.

Vulnerabilities exist within applications and services, and appear as a serious threat to cloud infrastructure. Cloud containers may also be at risk due to the injection of vulnerabilities to the images which can harm the cloud infrastructure. Exposure to the external resources may expose cloud containers to vulnerabilities.

Current system and method for detecting vulnerabilities in the cloud container provides limitation in regards to sources to scan and detect vulnerabilities. Thus, detection for vulnerabilities provided in the current invention is not accurate. Current method also requires more time to be executed due to the detection of vulnerability being done separately such as in the system, packages, files, binaries and etc.

United States Patent Application Publication No. US 20170098072 A1 (hereinafter referred to as the US 072 A1 Publication) entitled RUNTIME DETECTION OF VULNERABILITIES IN AN APPLICATION LAYER OF SOFTWARE CONTAINERS having a filing date of 28 September 2016 (Applicant: Twistlock, Ltd.) discloses a system and method for detecting vulnerabilities in software containers at runtime. The US 072 A1 Publication provides runtime vulnerability detection and utilizes CVE database and other security databases for known threats data. The US 072 A1 Publication only able to detect vulnerabilities in software containers at runtime. Further, the US 072 A1 Publication only detect for vulnerability when a

new file is created, modified or both and requires changes in the container for initiating vulnerability detection.

United States Patent Application Publication No. US 20170109536 A1 (hereinafter referred to as the US 536 A1 Publication) entitled STATIC DETECTION OF VULNERABILITIES IN BASE IMAGES OF SOFTWARE CONTAINERS having a filing date of 13 October 2016 (Applicant: Twistlock, Ltd.) discloses a system and method for detecting vulnerabilities in base images of software containers. The US 536 A1 Publication provides static image in a non-runtime environment vulnerability detection and utilizes CVE database and other security databases for known threats data. The US 536 A1 Publication only detects vulnerabilities in static images of software container. Further, the US 536 A1 Publication is configured to receive an event indicating that a base image in an image registry has been changed or added to initiate detection and utilises unitary signature such as a check-sum of a layer and a hash function computed over the contents of the layer generation when no vulnerability is detected.

United States Patent No. US 9438634 B1 (hereinafter referred to as the US 634 B1 Patent) entitled MICROSEGMENTED NETWORKS THAT IMPLEMENT VULNERABILITY SCANNING having a filing date of 28 August 2015 (Patentee: VARMOUR NETWORKS Inc.) discloses systems for providing vulnerability scanning within distributed microservices. The US 634 B1 Patent provides real-time vulnerabilities which utilizes active probe controller to implement execution of scanning schedules or cause vulnerability scanning to occur when potential malicious or suspicious activity, or network traffic is detected for any workload. The database for known threats, vulnerabilities or exploits is not disclose in The US 634 B1 Patent.

Due to the drawbacks and limitation of the currently available system and method, the present invention discloses a system and method which enable vulnerability detection of the cloud container based service deployment by through machine learning which enables time to time prediction of vulnerabilities and providing efficient and intelligent vulnerabilities detection at dynamic runtime stage and static image.

SUMMARY OF INVENTION

The present invention relates to a system and method for enabling vulnerability detection of cloud container based service deployment. In particular, the present invention provides time to time prediction of vulnerabilities through machine learning.

One aspect of the invention provides that a system (100) for enabling vulnerability detection of cloud container based service deployment comprising a user interface (102) for displaying vulnerabilities information to a user; at least one vulnerability detection server (104) enabling detection of vulnerabilities; a container deployment engine (110) comprising a plurality of containers (112A, 112B, 112C, 112D) for deploying services on at least one host or across a plurality of host; a plurality of images (114A, 114B, 114C, 114D) within the container deployment engine (110) for distributing services in a series of layers; and at least one common vulnerability and exposure database (106) and at least one exploit database (108) connected to the vulnerability detection server (104) for providing vulnerabilities information.

The vulnerability detection server (104) further comprising at least one Data Extractor (118) having means for obtaining a container identifier and name of the container; extracting information of the container in the container deployment engine; and merging all information obtained for generating a vectorized document of each container; at least one Data Binder (120) having means for obtaining common vulnerabilities and exposures identifier; submitting query to at least one common vulnerabilities and exposures database (106) and at least one exploit database (108); and merging information obtained for generating a vectorized document of each common vulnerabilities and exposures; at least one Data Converter (122) having means for obtaining the vectorized document of each container from at least one Data Extractor (118) and the vectorized document of each common vulnerabilities and exposures from at least one Data Binder (120); and handling conversions of the said documents; at least one Core module (124) having means for obtaining relevancy between the vectorized document of each container and the vectorized document of each common vulnerabilities and exposures, and associating the vectorised documents to the corresponding common vulnerabilities exposures identifier; and at least one database (116) for handling detection of vulnerability in the container deployment engine (110).

Another aspect of the invention provides that the information obtained for generating a vectorized document of each common vulnerabilities and exposures through at least one Data Binder may comprise of analysis description, product affected by, number of affected

version, vulnerability conditions, Metasploit modules and description of the patch or guide information of common vulnerabilities exposures identification.

A further aspect of the invention provides that a method (200) for enabling vulnerability detection of cloud container based service deployment comprises steps of binding, filtering and vectorizing common vulnerabilities and exposures identifier for obtaining a vectorized document of each common vulnerabilities and exposures through at least one Data Binder (202); extracting, filtering and vectorizing container information for obtaining a vectorized document of each container through at least one Data Extractor (204); determining availability of evaluation history (206); evaluating the vectorized document of each common vulnerabilities and exposures and the vectorized document of each container, if evaluation history is not available (208); and performing prediction of vulnerabilities from outcome of relevancy comparison between vectorised document of each common vulnerabilities and exposures and the vectorised document of each container, if evaluation history is available (210).

Yet another aspect of the invention provides that the binding, filtering and vectorizing common vulnerabilities and exposures identifier for obtaining a vectorized document of each common vulnerabilities and exposures through at least one Data Binder (202) further comprises steps of (300) obtaining common vulnerabilities exposures identifier of each vulnerability (302); connecting to a common vulnerabilities exposures database (304); obtaining information of common vulnerabilities exposures identifier (306); connecting to an exploit database (308) for obtaining information of exploits (310); performing data cleansing for filtering information of common vulnerabilities exposures and exploits (312); merging information obtained for vectorizing information (314); tagging information merged as a vectorized document of each common vulnerabilities and exposures (316); and presenting the vectorized document of each common vulnerabilities and exposures to at least one Data Converter (122) (318).

Still another aspect of the invention provides that the extracting, filtering and vectorizing container information for obtaining a vectorized document of each container through at least one Data Extractor (204) further comprises steps of (400): connecting to a container deployment engine (110) for obtaining container identifier (402); identifying container deployment directory from a host to a container (404); obtaining access from the host to the container (406); obtaining a package list of the container (408); obtaining file name and binaries of the container (410); obtaining operating system information of the container (412); merging obtained information for vectorising information upon filtering and cleansing

the obtained information (414); tagging the information as a vectorized document of each container (416); and presenting the vectorized document of each container to at least one Data Converter (122) (418).

Another aspect of the invention provides that presenting the vectorized document of each common vulnerabilities and exposures to at least one Data Converter (122) (318) further comprising steps of (500): appending vectorized documents of each common vulnerabilities and exposures as training set (502a); separating each work in vectorized documents of each common vulnerabilities and exposures and applying inverse document frequency to the vectorized documents of each common vulnerabilities and exposures (504a); applying term frequency to vectorized documents of each common vulnerabilities and exposures (506a); generating vector space values of vectorized documents of each common vulnerabilities and exposures and indexing vectorized documents of each common vulnerabilities and exposures with common vulnerabilities exposures identifier (508a); generating vector space matrix of vectorized documents of each common vulnerabilities and exposures (510a); and submitting vector space matrix to at least one Core module (124) for relevancy comparison between vectorized documents of each common vulnerabilities and exposures and vectorized document of each container (512a).

A further aspect of the invention provides that presenting the vectorized document of each container to at least one Data Converter (122) (418) further comprising steps of (500): receiving the vectorized document of each container as testing set (502b); separating each work in the vectorized document of each container and applying inverse document frequency data to the vectorized document of each container (504b); applying term frequency to the vectorized document of each container (506b); generating vector space values of the vectorized document of each container and storing vector space values for prediction (508b); generating vector space matrix of the vectorized document of each container (510b); and submitting vector space matrix to at least one Core module (124) for relevancy comparison (512b) between vectorized documents of each common vulnerabilities and exposures and the vectorized document of each container.

Yet another aspect of the invention provides that sending the vector space matrix to the Core module (124) for relevancy comparison and evaluation (512a) further comprises steps of (600); receiving vector space matrix of vectorized documents of each common vulnerabilities and exposures and vectorized documents of each container from at least one Data Converter (601); determining availability of prediction training set for relevancy comparison to detect vulnerabilities (602); if prediction training set is available (602a):

performing relevancy comparison between vector space matrix of the vectorized document of each container with a vectorized document of outcome as classifier to predict (604); determining if relevancy is detected from relevancy comparison (605); if relevancy is detected (605a); retrieving common vulnerabilities exposures identifier of vectorized documents of each common vulnerabilities and exposures to be included in a report (608); and displaying the report to a user (618); if relevancy is not detected (605b); generating a report based on relevancy comparison outcome, (606); and displaying the report to a user (618); if prediction training set is not available (602b); performing relevancy comparison vector space matrix of the vectorized document of each container with vector space matrix of vectorized documents of each common vulnerabilities and exposures for relevancy comparison (610); determining if relevancy is detected from the relevancy comparison (611); if relevancy is detected (611a); tagging the vectorized document of each container to be included in the vectorized document of outcome as classifier to predict, (614); obtaining the vectorized document of outcome as classifier to predict as a train data set for future prediction (616a); reiterating steps 604, 605, 650a, 608, 618, 605b, 606, 618; tagging the vectorized document of each container to be included in the vectorized document of outcome as classifier to predict (614); retrieving the common vulnerabilities exposures identifier of vectorized documents of each common vulnerabilities and exposures to be included in a report (616b); and displaying the report to a user (618); if relevancy is not detected (611b); generating a report based on relevancy comparison outcome (612); and displaying the report to a user (618).

Still another aspect of the invention provides that sending the vector space matrix to the Core module (124) for relevancy comparison and evaluation (512b) further comprises steps of (600): receiving vector space matrix of vectorized documents of each common vulnerabilities and exposures and the vectorized document of each container from at least one Data Converter (601); determining availability of prediction training set for relevancy comparison for detecting vulnerabilities (602); if prediction training set is available (602a): performing relevancy comparison between vector space matrix of the vectorized document of each container with a vectorized document of outcome as classifier to predict (604); determining if relevancy is detected from relevancy comparison (605); if relevancy is detected (605a); retrieving common vulnerabilities exposures identifier of vectorized documents of each common vulnerabilities and exposures to be included in a report (608); and displaying the report to a user (618); if relevancy is not detected (605b); generating a report based on relevancy evaluation outcome, (606); and displaying the report to a user (618); if prediction training set is not available (602b); performing relevancy comparison vector space matrix of the vectorized document of each container with vector space matrix of

vectorized documents of each common vulnerabilities and exposures for relevancy comparison (610); determining if relevancy is detected from the relevancy comparison (611); if relevancy is detected (611a); tagging the vectorized documents of each container to be included in the vectorized document of outcome as classifier to predict (614); obtaining the vectorized document of outcome as classifier to predict as a train data set for future prediction (616a); reiterating steps 604, 605, 650a, 608, 618, 605b, 606, 618; tagging the vectorized document of each container to be included in the vectorized document of outcome as classifier to predict, (614); retrieving the common vulnerabilities exposures identifier of vectorized documents of each common vulnerabilities and exposures to be included in a report (616b); and displaying the report to a user (618); if relevancy is not detected (611b); generating a report based on relevancy comparison outcome (612); and displaying the report to a user (618).

The present invention consists of features and a combination of parts hereinafter fully described and illustrated in the accompanying drawings, it being understood that various changes in the details may be made without departing from the scope of the invention or sacrificing an of the advantages of the present invention.

BRIEF DESCRIPTION OF ACCOMPANING DRAWINGS

To further clarify various aspects of some embodiments of the present invention, a more particular description of the invention will be rendered by references to specific embodiments thereof, which are illustrated in the appended drawings. It is appreciated that these drawings depict only typical embodiments of the invention and are therefore not to be considered limiting of its scope. The invention will be described and explained with additional specificity and detail through the accompanying drawings.

Fig. 1.0 illustrates a general system architecture of the present invention for enabling vulnerability detection of a cloud container based service deployment.

Fig 2.0 is a flowchart illustrating a general methodology of the present invention for detecting vulnerabilities in the cloud container.

Fig. 2.0a illustrates a schematic diagram of the configuration of the system architecture of the present invention.

Fig. 3.0 is a flowchart illustrating the steps involved in binding, filtering and vectorizing common vulnerabilities and exposures identifier for obtaining a vectorized document of each container.

Fig. 4.0 is a flowchart illustrating the steps involved in extracting, filtering and vectorizing container information for obtaining a vectorized document of each common vulnerabilities and exposures.

Fig. 5.0 is a flowchart illustrating the steps involved in converting information for obtaining vector space values and vector space matrix.

Fig. 6.0 is a flowchart illustrating the steps involved for relevancy comparison for detecting vulnerabilities.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention relates to a system and method for enabling vulnerability detection of cloud container based service deployment. In particular, the present invention relates to a system and method that enables efficient and intelligent flow of vulnerabilities detection with time to time prediction of vulnerabilities through machine learning. The present invention also allows vulnerabilities detection at dynamic runtime stage and static image. Hereinafter, it is to be understood that limiting the description to the preferred embodiments of the invention is merely to facilitate discussion of the present invention and it is envisioned without departing from the scope of the appended claims.

Reference is first made to FIG. 1.0 which illustrates a general system architecture of the present invention for enabling vulnerability detection of a cloud container based service deployment. As illustrated in Fig. 1.0, the system comprising a user interface (102), at least one vulnerability detection server (104), a container deployment engine (110), a plurality of images (114A, 114B, 114C, 114D), at least one common vulnerabilities and exposures database (106) and at least one exploits database (108).

The user interface (102) displays vulnerabilities information and a report to a user wherein the vulnerabilities information is provided to the user interface (102) through the vulnerability detection server (104). The vulnerability detection server (104) interacts with the common vulnerabilities exposures database (106) and the exploit database (108) for obtaining information pertaining to the vulnerabilities and exploits. However, the vulnerabilities and exploits information may also be obtained from other external databases and not limited to the common vulnerabilities and exposures database (106) and the exploit database (108).

The vulnerability detection server (104) also interacts with the container deployment engine (110) for detecting vulnerabilities and exploits in at least one host. As illustrated in FIG. 1.0, the container deployment engine (110) comprising a plurality of containers (112A, 112B, 112C, 112D) which deploys services in an isolated environment of the host or across a plurality of hosts, and the plurality of images (114A, 114B, 114C, 114D) in a layered architecture. The services of the container deployment engine (110) are distributed by packaging services in a series of layers of the plurality of images (114A, 114B, 114C, or 114D). The layers may include a system to host applications and services, files and binaries, and packages and libraries.

As illustrated in FIG. 1.0, the vulnerability detection server (104) comprising a plurality of components which utilizes machine learning instructions for handling detection of vulnerabilities, wherein the plurality of components includes at least one database (116), at least one Data Extractor (118), at least one Data Binder (120), at least one Data Converter (122), and at least one Core module (124). The database (116) within the vulnerability detection server (104) allows the information obtained in regard to vulnerabilities and exploits, and other relevant information and documents to be stored and managed. Further, the Data Extractor (118) within the vulnerability detection server (104) obtains and extracts a container identifier along with the container information such as package list, files, and binaries. The extracted container information extracted is merged and presented into a vectorized document of each container, also known as a ContDoc.

The Data Binder (120) on the other hand obtains a common vulnerabilities and exposures identifier as an input and generates a query to common vulnerabilities and exposures database (106) and exploit database (108) for obtaining information pertaining to the vulnerabilities and exploits. The information to be obtained from the common vulnerabilities and exposures database (106) and exploit database (108) is defined specifically prior to each query. The information obtained will be merged and presented into a vectorized document of each common vulnerabilities and exposures also known as a VulDoc.

The vectorized document of each container generated through the Data Extractor (118) and the vectorized document of each common vulnerabilities and exposures generated through the Data Binder (120) are then presented to the Data Converter (122). The Data Converter (122) provides conversions of the vectorized documents into trained documents for predicting vulnerabilities and exploits allowing efficient and intelligent flow for detecting vulnerabilities from time to time at dynamic runtime stage and static image. The Data Converter (122) utilizes two components namely a Re-trainer or a PredDoc within the Core module (124) which is also known as a vectorized document of outcome as classifier to predict) and a vector author. The Core module (124) obtains relevancy between the vectorized document of each container and the vectorized document of each common vulnerabilities and exposures, and further associates the said relevancy obtained to the corresponding common vulnerabilities and exposures identifier through two components namely a Relevancy Author and an Indexer within the Core module (124).

Reference is now made to FIG. 2.0. FIG. 2.0 is a flowchart illustrating the general methodology of the present invention for detecting vulnerabilities in cloud container (200). As illustrated in FIG. 2.0, the Data Binder (120) first binds, filters, vectorizes and indexes the

common vulnerabilities and exposures identifier for obtaining the vectorized document of each common vulnerabilities and exposures (202). The common vulnerabilities and exposures information including exploits information is first gathered from both common vulnerabilities and exposures database (106) and exploit database (108) to be filtered accordingly. The filtered information is subsequently vectorized, indexed, and stored in the database (116).

Subsequently, the Data Extractor (118) extracts, filters and vectorizes the container information for obtaining the vectorized document of each container (204). The name and the information of the container which obtained from the container deployment engine (110) is extracted and filtered accordingly. The information of container which have been filtered is subsequently vectorized.

Availability of evaluation history is determined (206). If the evaluation history is not available, the evaluation is performed between the vectorized document of each common vulnerabilities and exposures and the vectorized document of each container through the Core module (124) (208). However, if the evaluation history of the container information is available, the prediction is performed promptly using relevancy outcome from evaluation history through the Core module (124) (210).

Reference is now made to FIG. 2.0a. FIG. 2.0a illustrates a schematic diagram of the configuration of the system architecture of the present invention as illustrated in FIG. 2.0. As illustrated in FIG. 2.0a and step 202 from FIG. 2.0, the common vulnerabilities and exposures identifier is first obtained through the Data Binder (120). The Data Binder (120) then submits a query to the common vulnerabilities and exposures database (106) and the exploit database (108) for obtaining common vulnerabilities and exposures information, wherein the information may include general information of the common vulnerabilities and exposures such as description, platform, available codes, tags as well as exploit information. The common vulnerabilities and exposures information obtained through the Data Binder (120) is therefore merged and presented as a vectorized document of each common vulnerabilities and exposures. Thereafter, the vectorized document of each common vulnerabilities and exposures is presented to the Data Converter (122) wherein the Data Converter (122) subsequently filters and vectorizes the vectorized document of each common vulnerabilities and exposures as a bag of vectors.

Subsequently, with reference to step 204 of FIG. 2.0, the container identifier and container information such as operating system, application and packages, files and binaries which

corresponds to the name of each container is obtained through the Data Extractor (118). The information of the container obtained from the container deployment engine (110) is extracted, filtered, and vectorized accordingly an being presented as a vectorized document of each container. Thereafter, the vectorized document of each container is presented to the Data Converter (122) wherein the Data Converter (122) subsequently filters and vectorizes the vectorized document of each container as a bag of vectors.

The Data Converter (122) subsequently presents the vectorized document of each common vulnerabilities and exposures and the vectorized document of each container to the Core module (124) for evaluating relevancy and indexing, wherein the vectorized document of each common vulnerabilities and exposures is used for detecting vulnerability from the vectorized document of each container. Each vectorized document of each common vulnerabilities and exposures presented to the Core module (124) is indexed and subsequently stored in the database (116) for future evaluation.

Reference is now made to FIG. 3.0 whereby FIG. 3.0 is a flowchart illustrating the steps involves in binding, filtering and vectorizing the common vulnerabilities and exposures identifier for obtaining the vectorized document of each common vulnerabilities and exposures as illustrated in FIG. 2.0, step 202 and FIG. 2.0a. First, the common vulnerabilities and exposures identifier of each vulnerability is obtained by the Data Binder (120) (302). Thereafter, a query is made to the common vulnerabilities and exposures database (106) (304) for obtaining the information of the common vulnerabilities and exposures identifier including information such as analysis description, product affected by, number of affected version, vulnerability conditions, Metasploit modules, and description of the patch and guide (306). Thereafter, query is also made to the exploit database (108) (308) in order to obtain information of exploits which corresponds to the common vulnerabilities and exposures identifier such as exploit identifier title, code comments, platform name, aliases and tags (310).

Subsequently, the Data Binder (120) performs data cleansing for filtering the obtained common vulnerabilities and exposures and exploits information (312) and further merges the common vulnerabilities and exposures and exploits information accordingly for vectorizing the information (314). Further, the vectorized information of each specific common vulnerabilities and exposures identifier is tagged as a vectorized document of each common vulnerabilities and exposures (316). Finally, the vectorized document of each common vulnerabilities and exposures is presented to the Data Converter (122) (318).

Reference is now made to FIG. 4.0. FIG. 4.0 is a flowchart illustrating the steps involves in extracting, filtering and vectorizing container information for obtaining the vectorized document of each container (400) as illustrates in FIG. 2.0, step 204 as well as FIG. 2.0a. As illustrated in FIG. 4.0, the Data Extractor (118) first interacts with the container deployment engine (110) in order to obtain the information of the container including a running container identifier (402). Thereafter, the Data Extractor (118) identifies the container deployment directory from the host to the container (112A, 112B, 112C, 112D) (404) and obtains access with privilege from the host to container (112A, 112B, 112C, 112D) based on the obtained container identifier (406). Subsequently, a package list of specific container (112A, 112B, 112C or 112D) is obtained (408) together with a list of file names and binaries (410). The operating system of the container (112A, 112B, 112C, 112D) and other relevant files pertaining to the container (112A, 112B, 112C, 112D) are also obtained (412). The information obtained is merged accordingly and vectorizes information upon filtering and cleansing of the same (414). The said vectorized information is tagged as a vectorized document of each container (416) and presented to the Data Converter (122) (418).

Reference is now made to FIG. 5.0 whereby FIG. 5.0 is a flowchart illustrating the steps involved in converting information for obtaining vector space values and vector space matrix in both vectorized document of each common vulnerabilities and exposures and vectorized document of each container (500). As illustrated in FIG. 3.0, step 318 and FIG. 4.0, step 418, the vectorized document of each common vulnerabilities and exposures and vectorized document of each container are presented to the Data Converter (122) for further vectorization steps. As illustrated in FIG. 5.0, the steps involved in converting information of the vectorized document of each common vulnerabilities and exposures and the vectorized document of each container are simultaneously performed. All available vectorized documents of each common vulnerabilities and exposures received by the Data Converter (122) is appended as a training set (502a) and subsequently each work in the vectorized document of each common vulnerabilities and exposures is separated and is applied with inverse document frequency, IDF (504a). Further to the application of the inverse document frequency, each vectorized document of each common vulnerabilities and exposures is subsequently applied with term frequency (506a). Thereafter, the Data Converter (122) generates vector space values of each vectorized document of each common vulnerabilities and exposures and indexes the vectorized document of each common vulnerabilities and exposures with the common vulnerabilities and exposures identifier (508a). The Data Converter (122) further generates vector space matrix of each vectorized document of each common vulnerabilities and exposures (510a) and subsequently presented the same to the Core module (124) for relevancy comparison (512a).

Simultaneous to the step 502a, the vectorized document of each container received from the Data Extractor (118) are also appended as training set (502b). Further, each work in the vectorized document of each container are separated and is applied with inverse document frequency, IDF (504b). Subsequent to the application of IDF, the vectorized document of each container is applied with term frequency (506b). Thereafter, the Data Converter (122) generates vector space values of the vectorized document of each container and stores the same for prediction of vulnerabilities (508b). The Data Converter (122) further generates vector space matrix of the vectorized document of each container (510b) and subsequently presented the vector space matrix of the vectorized document of each container to the Core module (124) for relevancy comparison (510b).

Reference is now made to FIG. 6.0. FIG. 6.0 is a flowchart illustrating the steps involved for relevancy comparison in order to detect vulnerabilities. Upon receiving vector space matrix of the vectorized document of each common vulnerabilities and exposures and the vectorized document of each container from the Data Converter (122) (601), the Core module (124) determines the availability of prediction training set for relevancy comparison to detect vulnerabilities (602). If the training set is available (602a), the vector space matrix of the vectorized document of each container will be compared to the vectorized document of outcome as classifier to predict for relevancy comparison (604). Thereafter, the Core module (124) determines whether any relevancy is detected from the outcome of the relevancy comparison (605). If relevancy is detected from the evaluation (605a), the common vulnerabilities and exposures identifier of the vectorized document of each common vulnerabilities and exposures is retrieved to be included in a report (608) and subsequently display the report to the user (618). However, if there is no relevancy detected from the relevancy evaluation (605b), a report will be generated based on the relevancy comparison outcome (606) and the report will be presented to the user (618).

Alternatively, if the prediction training set is not available (602b), vector space matrix of the vectorized document of each container will be compared to vector space matrix of the vectorized document of each common vulnerabilities and exposures for relevancy comparison (610). If relevancy is detected (611a), the vectorized document of each container is tagged to be included in the vectorized document of outcome as classifier to predict (614) and to be utilized as a train data set (616a) for future prediction.

The said vectorized document of outcome as classifier to predict (614) and to be utilized as the train data set (616a) further comprises of two routes. In the first route, the train data set obtained (616a) will be used for relevancy comparison in step 604. Thereafter, the Core

module (124) determines whether any relevancy is detected from the outcome of the relevancy comparison (605). If relevancy is detected from the evaluation (605a), the common vulnerabilities and exposures identifier of the vectorized document of each common vulnerabilities and exposures is retrieved to be included in a report (608) and subsequently display the report to the user (618). Nevertheless, if there is no relevancy detected from the relevancy evaluation (605b), a report will be generated based on the relevancy comparison outcome (606) and the report will be presented to the user (618).

In the second route, the common vulnerabilities and exposures identifier of the vectorized document of each common vulnerabilities and exposures is then retrieved to be included in a report (616b). The report is then displayed to the user (618). However, if there is no relevancy detected (611b) from the relevancy evaluation, a report will be generated based on the relevancy comparison outcome (616) and displayed to the user (618).

The present invention relates to a system and method which enables efficient and intelligent flow of vulnerabilities detection by providing time to time prediction of vulnerabilities and allowing vulnerabilities detection during dynamic runtime stage and static image using machine learning. The present invention utilizes at least one vulnerability detection server (104) which comprising at least one Data Extractor (118) for generating vectorized document of each container, at least one Data Binder (120) for generating vectorized document of each common vulnerabilities and exposures, at least one Data Converter (122) for converting vectorized documents of each container and vectorized documents of each common vulnerabilities of exposures and at least on Core module (124) for relevancy comparison.

Throughout this specification, unless the context requires otherwise, the word “comprise”, or variations such as “comprises” or comprising, will be understood to imply the inclusion of a stated step or element to integer or group of steps or elements or integers, but not the exclusion of any other step or element or integer or group of steps, element or integers. Thus, in the context of this specification, the term “comprising” is used in an inclusive sense and thus should be understood as meaning “including principally, but not necessarily solely”.

CLAIMS

1. A system (100) for enabling vulnerability detection of cloud container based service deployment comprising:

a user interface (102) for displaying vulnerabilities information to a user;

at least one vulnerability detection server (104) enabling detection of vulnerabilities;

a container deployment engine (110) comprising a plurality of containers (112A, 112B, 112C, 112D) for deploying services on at least one host or across a plurality of host;

a plurality of images (114A, 114B, 114C, 114D) within the container deployment engine (110) for distributing services in a series of layers; and

at least one common vulnerability and exposure database (106) and at least one exploit database (108) connected to the vulnerability detection server (104) for providing vulnerabilities information;

characterized in that

the vulnerability detection server (104) further comprising:

at least one Data Extractor (118) having means for:

obtaining a container identifier and name of the container;

extracting information of the container in the container deployment engine; and

merging all information obtained for generating a vectorized document of each container;

at least one Data Binder (120) having means for:

obtaining common vulnerabilities and exposures identifier;

submitting query to at least one common vulnerabilities and exposures database (106) and at least one exploit database (108); and

merging information obtained for generating a vectorized document of each common vulnerabilities and exposures;

at least one Data Converter (122) having means for:

obtaining the vectorized document of each container from at least one Data Extractor (118) and the

vectorized document of each common vulnerabilities and exposures from at least one Data Binder (120); and

handling conversions of the said documents;

at least one Core module (124) having means for:

obtaining relevancy between the vectorized document of each container and the vectorized document of each common vulnerabilities and exposures, and

associating the vectorised documents to the corresponding common vulnerabilities exposures identifier; and

at least one database (116) for handling detection of vulnerability in the container deployment engine (110).

2. The system according to claim 1, wherein information obtained for generating a vectorized document of each common vulnerabilities and exposures through at least one Data Binder comprising analysis description, product affected by, number of affected version, vulnerability conditions, Metasploit modules and description of the patch or guide information of common vulnerabilities exposures identification.

3. A method (200) for enabling vulnerability detection of cloud container based service deployment comprises steps of:

binding, filtering and vectorizing common vulnerabilities and exposures identifier for obtaining a vectorized document of each common vulnerabilities and exposures through at least one Data Binder (202);

extracting, filtering and vectorizing container information for obtaining a vectorized document of each container through at least one Data Extractor (204);

determining availability of evaluation history (206);

evaluating the vectorized document of each common vulnerabilities and exposures and the vectorized document of each container, if evaluation history is not available (208); and

performing prediction of vulnerabilities from outcome of relevancy comparison between vectorised document of each common vulnerabilities and exposures and the vectorised document of each container, if evaluation history is available (210).

4. The method (200) according to claim 3, wherein binding, filtering and vectorizing common vulnerabilities and exposures identifier for obtaining a vectorized document of each common vulnerabilities and exposures through at least one Data Binder (202) further comprises steps of (300):

obtaining common vulnerabilities exposures identifier of each vulnerability (302);

connecting to a common vulnerabilities exposures database (304);

obtaining information of common vulnerabilities exposures identifier (306);

connecting to an exploit database (308) for obtaining information of exploits (310);

performing data cleansing for filtering information of common vulnerabilities exposures and exploits (312);

merging information obtained for vectorizing information (314);

tagging information merged as a vectorized document of each common vulnerabilities and exposures (316); and

presenting the vectorized document of each common vulnerabilities and exposures to at least one Data Converter (122) (318).

5. The method (200) according to claim 3, wherein extracting, filtering and vectorizing container information for obtaining a vectorized document of each container through at least one Data Extractor (204) further comprises steps of (400):

connecting to a container deployment engine (110) for obtaining container identifier (402);

identifying container deployment directory from a host to a container (404);

obtaining access from the host to the container (406);

obtaining a package list of the container (408);

obtaining file name and binaries of the container (410);

obtaining operating system information of the container (412);

merging obtained information for vectorising information upon filtering and cleansing the obtained information (414);

tagging the information as a vectorized document of each container (416); and

presenting the vectorized document of each container to at least one Data Converter (122) (418).

6. The method (200) according to claim 4, wherein presenting the vectorized document of each common vulnerabilities and exposures to at least one Data Converter (122) (318) further comprising steps of (500):

appending vectorized documents of each common vulnerabilities and exposures as training set (502a);

separating each work in vectorized documents of each common vulnerabilities and exposures and applying inverse document frequency to the vectorized documents of each common vulnerabilities and exposures (504a);

applying term frequency to vectorized documents of each common vulnerabilities and exposures (506a);

generating vector space values of vectorized documents of each common vulnerabilities and exposures and indexing vectorized documents of each common vulnerabilities and exposures with common vulnerabilities exposures identifier (508a);

generating vector space matrix of vectorized documents of each common vulnerabilities and exposures (510a); and

submitting vector space matrix to at least one Core module (124) for relevancy comparison between vectorized documents of each common vulnerabilities and exposures and vectorized document of each container (512a).

7. The method (200) according to claim 5, wherein presenting the vectorized document of each container to at least one Data Converter (122) (418) further comprising steps of (500):

receiving the vectorized document of each container as testing set (502b);

separating each work in the vectorized document of each container and applying inverse document frequency data to the vectorized document of each container (504b);

applying term frequency to the vectorized document of each container (506b);

generating vector space values of the vectorized document of each container and storing vector space values for prediction (508b);

generating vector space matrix of the vectorized document of each container (510b); and

submitting vector space matrix to at least one Core module (124) for relevancy comparison (512b) between vectorized documents of each common vulnerabilities and exposures and the vectorized document of each container.

8. The method (200) according to claim 6, wherein sending the vector space matrix to the Core module (124) for relevancy comparison and evaluation (512a) further comprises steps of (600):

receiving vector space matrix of vectorized documents of each common vulnerabilities and exposures and vectorized documents of each container from at least one Data Converter (601);

determining availability of prediction training set for relevancy comparison to detect vulnerabilities (602);

if prediction training set is available (602a):

performing relevancy comparison between vector space matrix of the vectorized document of each container with a vectorized document of outcome as classifier to predict (604);

determining if relevancy is detected from relevancy comparison (605);

if relevancy is detected (605a);

retrieving common vulnerabilities exposures identifier of vectorized documents of each common vulnerabilities and exposures to be included in a report (608); and

displaying the report to a user (618);

if relevancy is not detected (605b);

generating a report based on relevancy comparison outcome, (606); and

displaying the report to a user (618);

if prediction training set is not available (602b);

performing relevancy comparison vector space matrix of the vectorized document of each container with vector space matrix of vectorized documents of each common vulnerabilities and exposures for relevancy comparison (610);

determining if relevancy is detected from the relevancy comparison (611);

if relevancy is detected (611a);

tagging the vectorized document of each container to be included in the vectorized document of outcome as classifier to predict (614);

obtaining the vectorized document of outcome as classifier to predict as a train data set for future prediction (616a);

reiterating steps 604, 605, 650a, 608, 618, 605b, 606, 618;

tagging the vectorized document of each container to be included in the vectorized document of outcome as classifier to predict (614);

retrieving the common vulnerabilities exposures identifier of vectorized documents of each common vulnerabilities and exposures to be included in a report (616b); and

displaying the report to a user (618);

if relevancy is not detected (611b);

generating a report based on relevancy comparison outcome (612); and

displaying the report to a user (618).

9. The method (200) according to claim 7, wherein sending the vector space matrix to the Core module (124) for relevancy comparison and evaluation (512b) further comprises steps of (600):

receiving vector space matrix of vectorized documents of each common vulnerabilities and exposures and the vectorized document of each container from at least one Data Converter (601);

determining availability of prediction training set for relevancy comparison for detecting vulnerabilities (602);

if prediction training set is available (602a):

performing relevancy comparison between vector space matrix of the vectorized document of each container with a vectorized document of outcome as classifier to predict (604);

determining if relevancy is detected from relevancy comparison (605);

if relevancy is detected (605a);

retrieving common vulnerabilities exposures identifier of vectorized documents of each common vulnerabilities and exposures to be included in a report (608); and

displaying the report to a user (618);

if relevancy is not detected (605b);

generating a report based on relevancy evaluation outcome, (606); and

displaying the report to a user (618);

if prediction training set is not available (602b);

performing relevancy comparison vector space matrix of the vectorized document of each container with vector space matrix of vectorized documents of each common vulnerabilities and exposures for relevancy comparison (610);

determining if relevancy is detected from the relevancy comparison (611);

if relevancy is detected (611a);

tagging the vectorized documents of each container to be included in the vectorized document of outcome as classifier to predict (614);

obtaining the vectorized document of outcome as classifier to predict as a train data set for future prediction (616a);

reiterating steps 604, 605, 650a, 608, 618, 605b, 606, 618;

tagging the vectorized document of each container to be included in the vectorized document of outcome as classifier to predict, (614);

retrieving the common vulnerabilities exposures identifier of vectorized documents of each common vulnerabilities and exposures to be included in a report (616b); and

displaying the report to a user (618);

if relevancy is not detected (611b);

generating a report based on relevancy comparison outcome (612); and

displaying the report to a user (618).

100

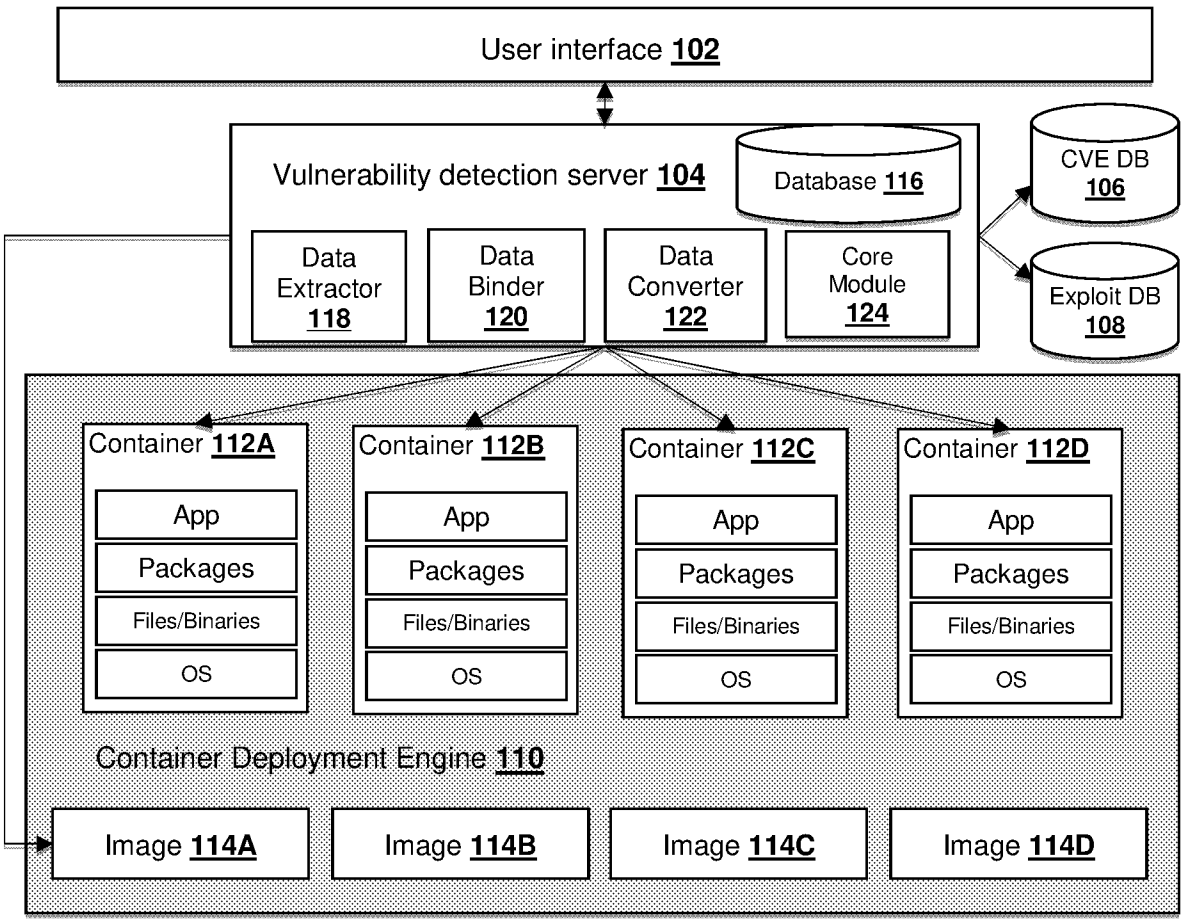
User interface 102
Vulnerability detection server 104
Database 116
CVE DB 106
Exploit DB 108
Data Extractor 118
Data Binder 120
Data Converter 122
Core Module 124
Container 112A
Container 112B
Container 112C
Container 112D
App
Packages
Files/Binaries
OS
Container Deployment Engine 110
Image 114A
Image 114B
Image 114C
Image 114D

FIG. 1.0

200

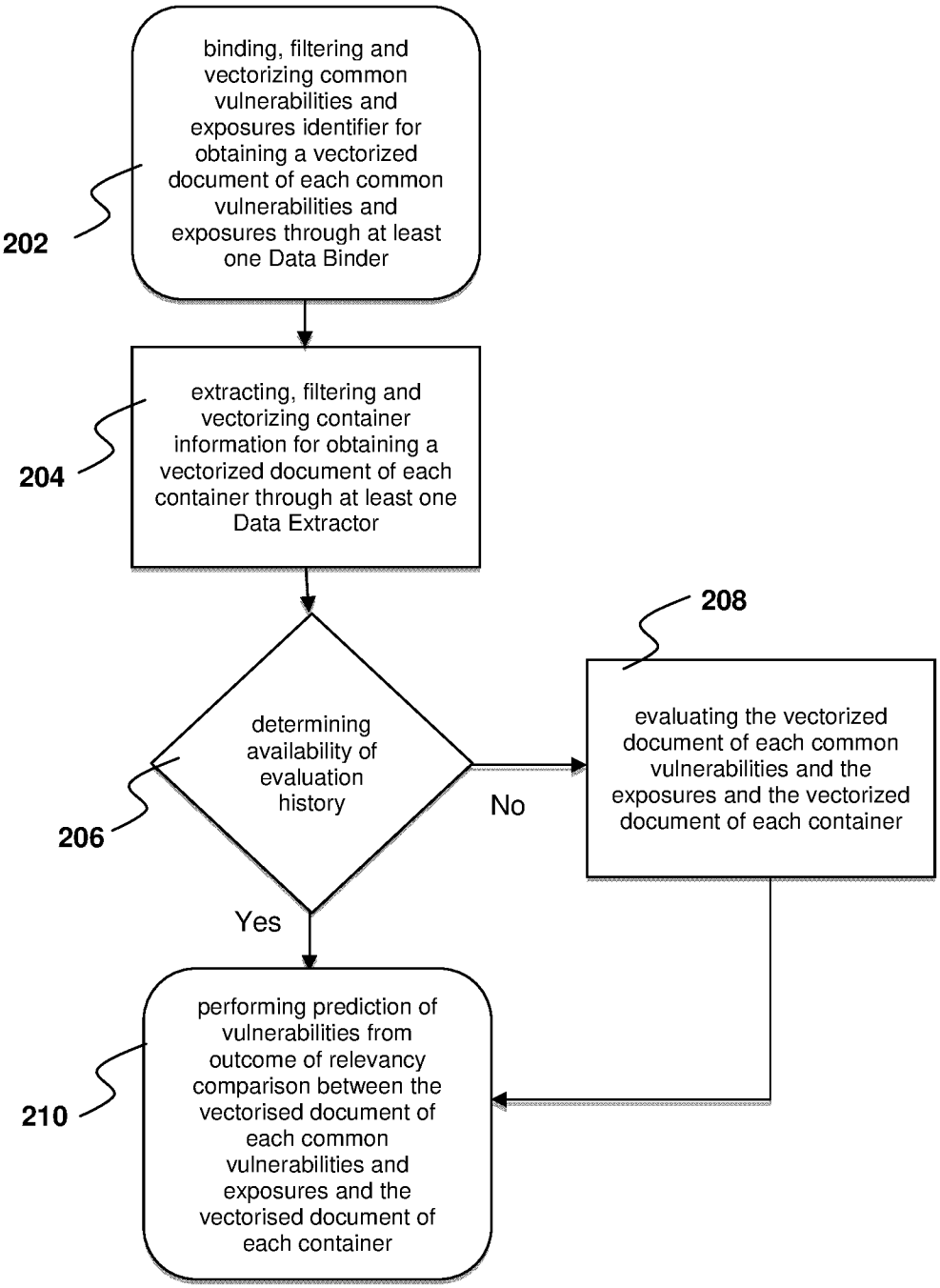
binding, filtering and vectorizing common vulnerabilities and exposures identifier for obtaining a vectorized document of each common vulnerabilities and exposures through at least one Data Binder
202
extracting, filtering and vectorizing container information for obtaining a vectorized document of each container through at least one Data Extractor
204
determining availability of evaluation history
206
No
208
evaluating the vectorized document of each common vulnerabilities and the exposures and the vectorized document of each container
Yes
performing prediction of vulnerabilities from outcome of relevancy comparison between the vectorised document of each common vulnerabilities and exposures and the vectorised document of each container
210

FIG. 2.0

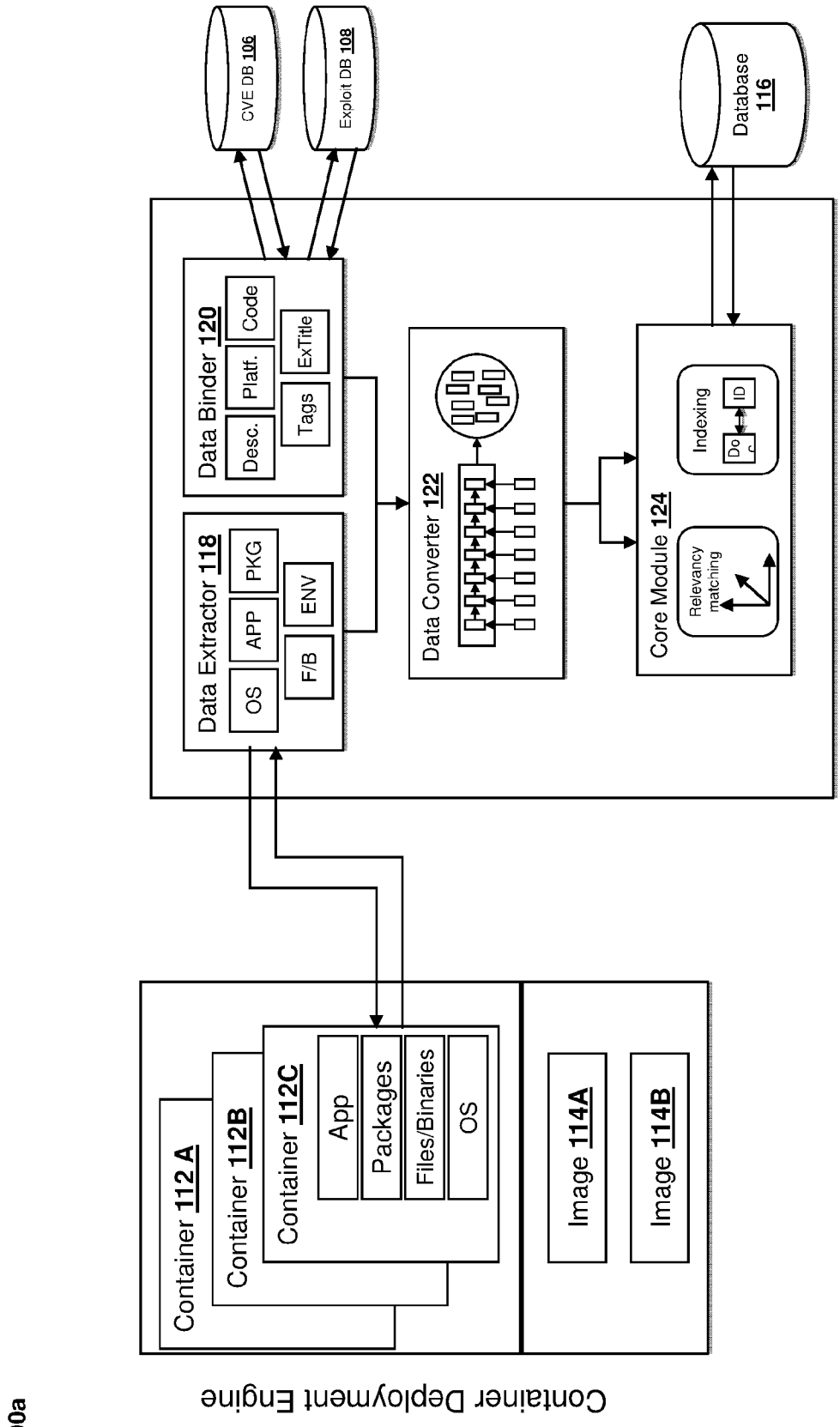
200a
Container Deployment Engine
Container 112 A
Container 112B
Container 112C
App
Packages
Files/Binaries
OS
Image 114A
Image 114B
Data Extractor 118
OS
APP
PKG
F/B
ENV
Data Binder 120
Desc.
Platf.
Tags
ExTitle
Code
CVE DB 106
Exploit DB 108
Data Converter 122
Core Module 124
Relevancy matching
Indexing
Database 116

FIG. 2.0a

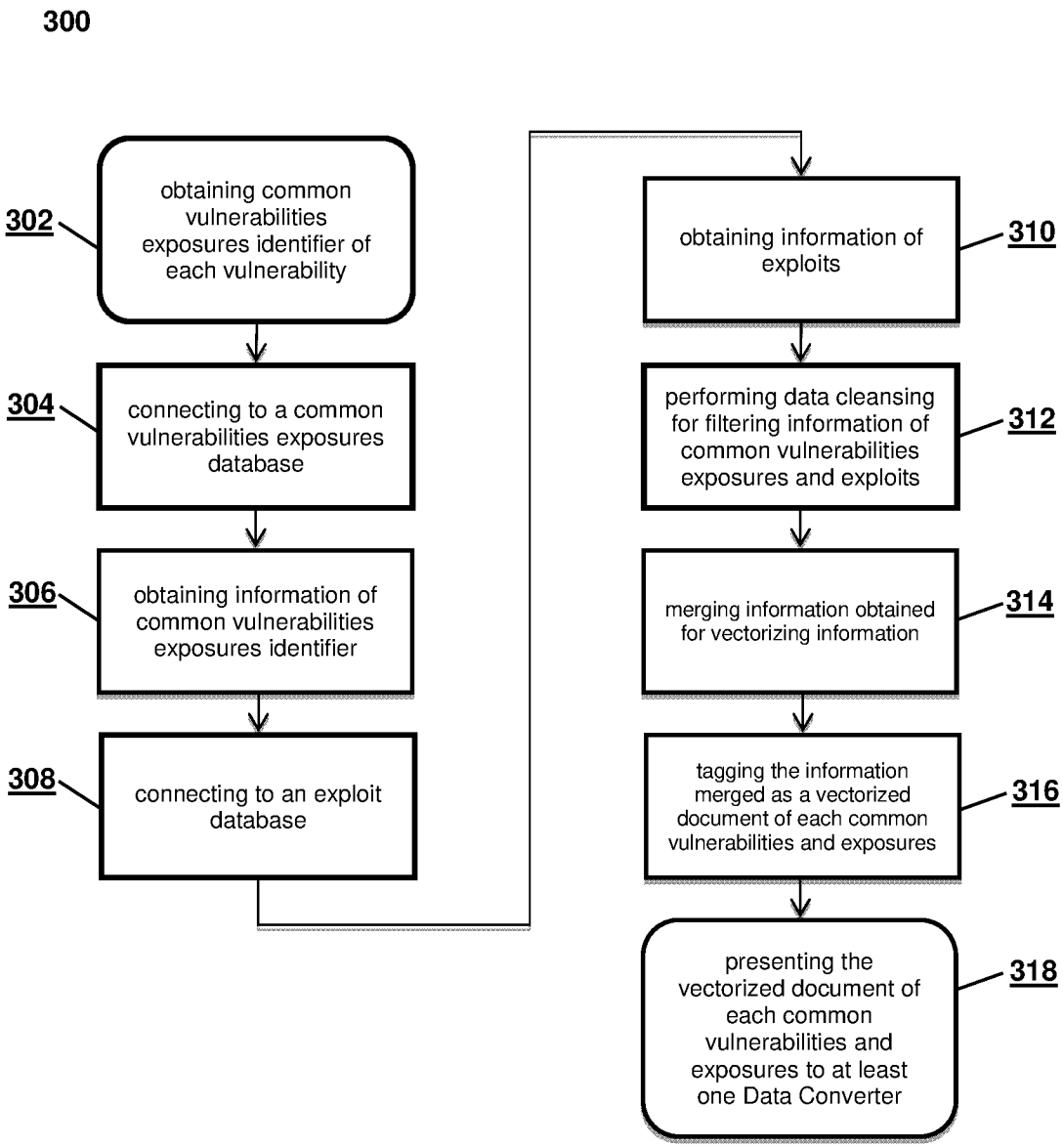
300
302
obtaining common vulnerabilities exposures identifier of each vulnerability
304
connecting to a common vulnerabilities exposures database
306
obtaining information of common vulnerabilities exposures identifier
308
connecting to an exploit database
310
obtaining information of exploits
312
performing data cleansing for filtering information of common vulnerabilities exposures and exploits
314
merging information obtained for vectorizing information
316
tagging the information merged as a vectorized document of each common vulnerabilities and exposures
318
presenting the vectorized document of each common vulnerabilities and exposures to at least one Data Converter

FIG. 3.0

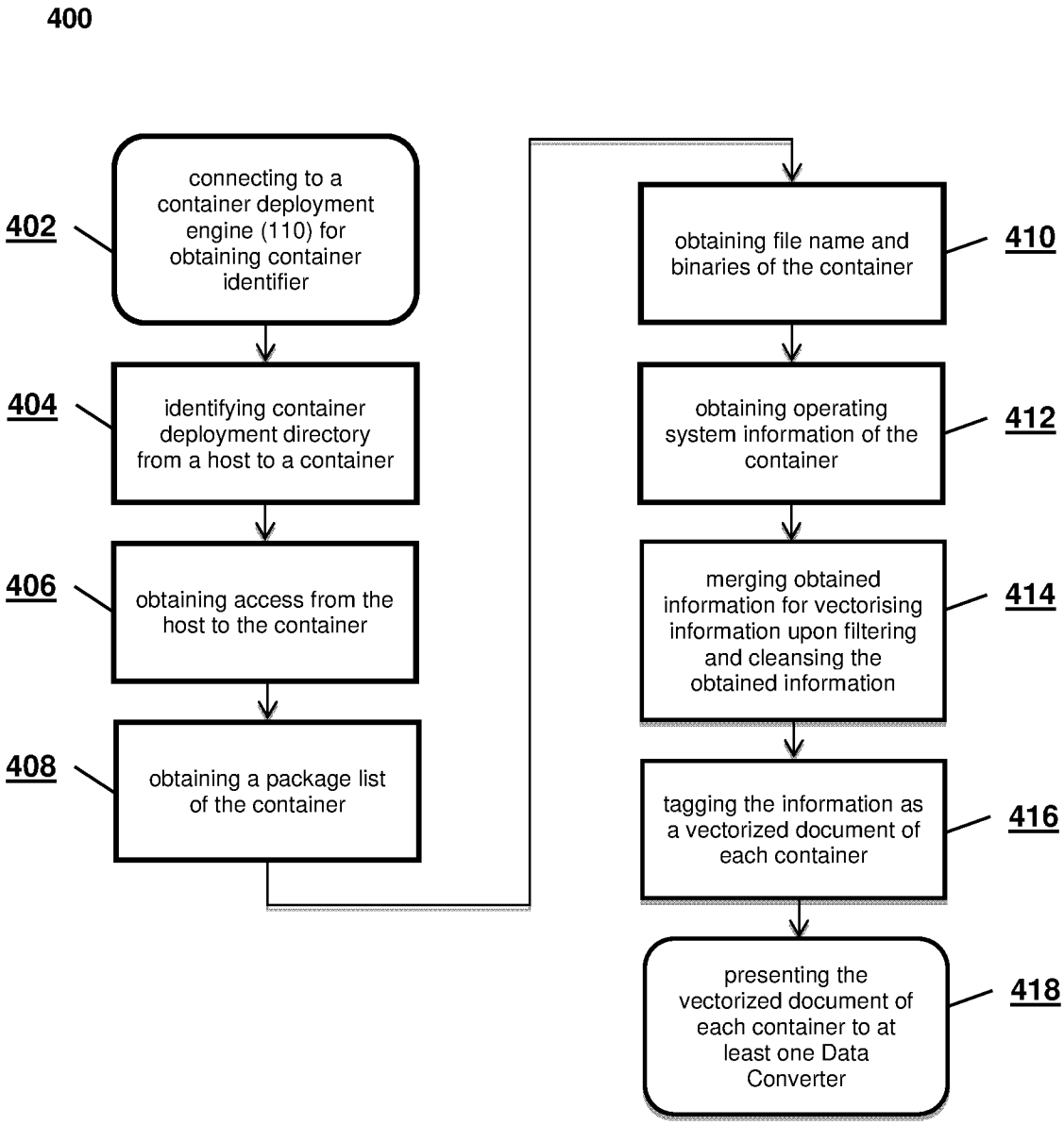
400
402
connecting to a container deployment engine (110) for obtaining container identifier
404
identifying container deployment directory from a host to a container
406
obtaining access from the host to the container
408
obtaining a package list of the container
410
obtaining file name and binaries of the container
412
obtaining operating system information of the container
414
merging obtained information for vectorising information upon filtering and cleansing the obtained information
416
tagging the information as a vectorized document of each container
418
presenting the vectorized document of each container to at least one Data Converter

FIG. 4.0

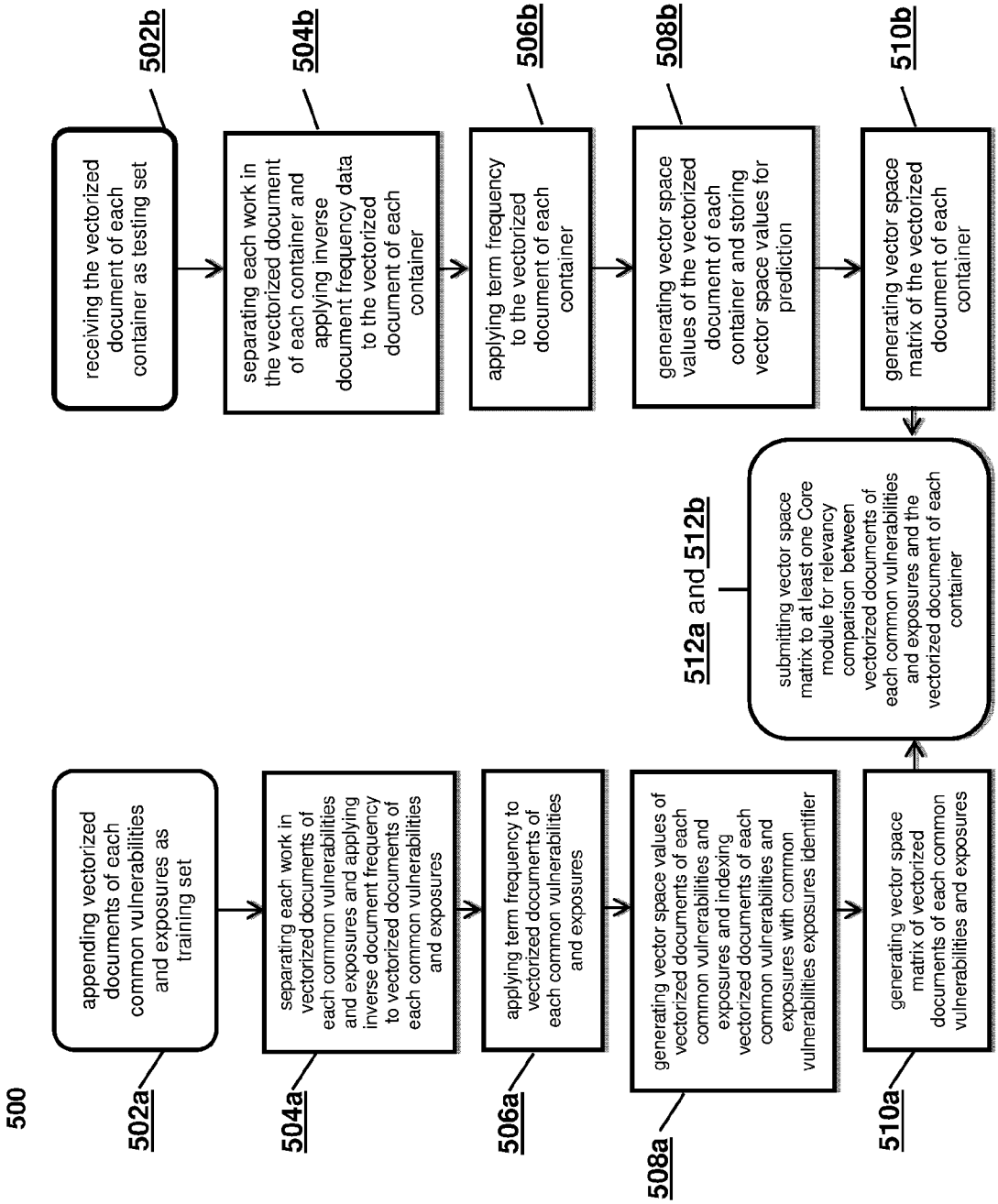
500
502a
appending vectorized documents of each common vulnerabilities and exposures as training set
504a
separating each work in vectorized documents of each common vulnerabilities and exposures and applying inverse document frequency to vectorized documents of each common vulnerabilities and exposures
506a
applying term frequency to vectorized documents of each common vulnerabilities and exposures
508a
generating vector space values of vectorized documents of each common vulnerabilities and exposures and indexing vectorized documents of each common vulnerabilities and exposures with common vulnerabilities exposures identifier
510a
generating vector space matrix of vectorized documents of each common vulnerabilities and exposures
512a and 512b
submitting vector space matrix to at least one Core module for relevancy comparison between vectorized documents of each common vulnerabilities and exposures and the vectorized document of each container
502b
receiving the vectorized document of each container as testing set
504b
separating each work in the vectorized document of each container and applying inverse document frequency data to the vectorized document of each container
506b
applying term frequency to the vectorized document of each container
508b
generating vector space values of the vectorized document of each container and storing vector space values for prediction
510b
generating vector space matrix of the vectorized document of each container

FIG. 5.0

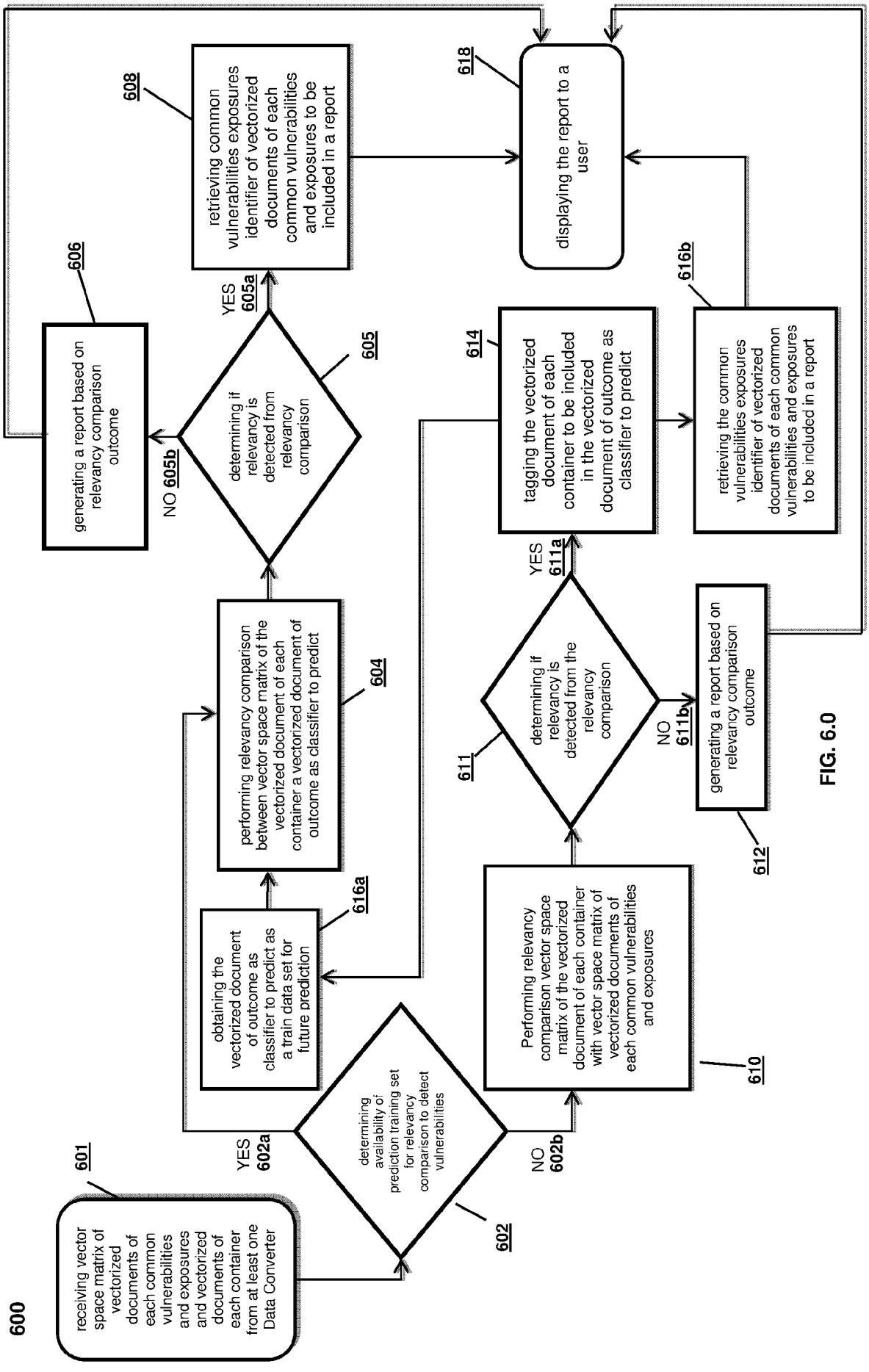
600
receiving vector space matrix of vectorized documents of each common vulnerabilities and exposures and vectorized documents of each container from at least one Data Converter
601
determining availability of prediction training set for relevancy comparison to detect vulnerabilities
602
YES 602a
NO 602b
obtaining the vectorized document of outcome as classifier to predict as a train data set for future prediction
616a
performing relevancy comparison between vector space matrix of the vectorized document of each container a vectorized document of outcome as classifier to predict
604
determining if relevancy is detected from relevancy comparison
605
YES 605a
NO 605b
retrieving common vulnerabilities exposures identifier of vectorized documents of each common vulnerabilities and exposures to be included in a report
608
generating a report based on relevancy comparison outcome
606
Performing relevancy comparison vector space matrix of the vectorized document of each container with vector space matrix of vectorized documents of each common vulnerabilities and exposures
610
determining if relevancy is detected from the relevancy comparison
611
YES 611a
NO 611b
tagging the vectorized document of each container to be included in the vectorized document of outcome as classifier to predict
614
retrieving the common vulnerabilities exposures identifier of vectorized documents of each common vulnerabilities and exposures to be included in a report
616b
generating a report based on relevancy comparison outcome
612
displaying the report to a user
618

FIG. 6.0

INTERNATIONAL SEARCH REPORT

International application No.
PCT/MY2019/050078

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/57(2013.01)i, G06F 9/455(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 21/57; G06F 21/55; G06F 21/56; G06N 99/00; H04L 29/06; G06F 9/455

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: vulnerability, detection, cloud, container, service, deployment, image, extract, obtain, merge, submit, convert, exposure, exploit, vectorized document, database

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2017-0318048 A1 (AUNG HTAY) 02 November 2017 paragraphs [0018], [0022], [0026], [0028], [0036], [0047], [0069]-[0075]; claim 9; and figures 1-2, 5	1-2,5
Y	US 2016-0232358 A1 (CISCO TECHNOLOGY, INC.) 11 August 2016 paragraphs [0018], [0021]-[0023]; and figures 2A-2B	1-5
Y	US 2018-0034842 A1 (BOOZ ALLEN HAMILTON INC.) 01 February 2018 paragraphs [0011], [0044]; claims 1-2; and figures 1, 5	3-5
A	US 2015-0207811 A1 (BEN FEHER et al.) 23 July 2015 paragraphs [0010]-[0033]; and figures 1-4	1-9
A	US 2018-0121649 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 03 May 2018 paragraphs [0021]-[0062]; and figures 1-3	1-9

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
13 February 2020 (13.02.2020)	**13 February 2020 (13.02.2020)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
KIM, Sung Hee
Telephone No. +82-42-481-3516

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/MY2019/050078

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2017-0318048 A1	02/11/2017	US 10050991 B2	14/08/2018
		US 2018-0041539 A1	08/02/2018
		US 9825982 B1	21/11/2017
US 2016-0232358 A1	11/08/2016	US 9928369 B2	27/03/2018
US 2018-0034842 A1	01/02/2018	US 10264009 B2	16/04/2019
		WO 2018-022321 A1	01/02/2018
US 2015-0207811 A1	23/07/2015	CN 104520871 A	15/04/2015
		EP 2880580 A1	10/06/2015
		EP 2880580 A4	20/01/2016
		WO 2014-021866 A1	06/02/2014
US 2018-0121649 A1	03/05/2018	US 9977898 B1	22/05/2018