

# **Automated Software Vulnerability Management and Monitoring - draft**

## Table of Contents

1	Introduction.....	3
1.1	Software Vulnerability Management .....	3
2	Automated Vulnerability Management and Monitoring System.....	4
2.1	System Components .....	4
2.2	System Work flow .....	4
3	Standards.....	6
3.1	SCAP .....	6
4	Alternatives to NVD Repository .....	7
5	Discussion .....	8
6	Conclusion .....	8

**Abstract.** Nowadays, one of the main concerning area for most of organizations is information and software assets security, in this paper we discuss a techniques and systems to automatically monitor the software venerability using open standards and public vulnerability data repositories or alternative sources such the social media and developer blogs.

## 1 Introduction

Nowadays, one of the main concerning area for most of organizations is information and software assets security which can be achieved by bt utilizing the software and network security techniques like firewall, but this is not sufficient to ensure the safety of the information assets due the continuous discovery of software vulnerabilities and put the organization's IT assets in risk of cyber attacks by adversary. Most of organization should be ware and updated about the open software vulnerabilities and perform the need actions like installing the corresponding patch if available, an such information about open vulnerabilities can be retrieved from repositories like National Vulnerability Database NVD.

The manual processing of vulnerabilities data requires a quite amount of human resource and could be not affordable for all organizations that motivates for building an automated software vulnerabilities management systems which in general try to alert in organizations about the open vulnerabilities related to the organization's IT assets utilizing the data of on-line repositories such NVD, also another sources of data can be also used like the software vulnerability information provided on social media or blogs.

### 1.1 Software Vulnerability Management

The software Vulnerability Management concept can be defined as the process of identifying the related vulnerabilities for the software components inside the organization, the process requires firstly managing the inventory of organization's IT assets and periodic search for the related vulnerabilities, and in case of vulnerability discovery a certain action should be performed such alerting the administrators or trying to install the corresponding patch es to avoid possible threats based on those known vulnerabilities.

## 2 Automated Vulnerability Management and Monitoring System

This section presents the proposed technique and system by Takahashi et al. in [5] to automatically monitoring the vulnerability of computing assets inside the organization/enterprise IT infrastructure network. their main contribution is automating the process of cyber security vulnerabilities management, using open standards and tools to make available to wide range of organizations.

The proposed system first collects and compiles the list of IT assets, and the stored data about the organization's computing components is used to find its identifiers, then the system utilizes those identifiers to check the existence of related vulnerabilities by searching the vulnerabilities repositories, and an alert about the identified security defects will be send to system administrator by the proposed system.

### 2.1 System Components

This section describes the elements of the proposed system and defines the role of each one.

The following are the 4 elements type for the automated vulnerability monitoring system:

- I **Terminals:** which include all electronic devices used to perform the different business activities by employees of the organization, in most cases an agent software installed on terminals to communicate with system coordinator server to provide the information about host terminal.
- II **System main coordinator server:** this management server is responsible to communicate the installed agents on terminals to collect information about computing assets of the organization, even server collects information about the assets without installed agents by monitoring and analyzing the enterprise's network traffic, to find the identifiers for all asset's parts, and the server perform the check for the vulnerability presence by the communication with vulnerability database.
- III **Vulnerability Knowledge base:** is the local system's database for the vulnerability information, to composed data from different vulnerability repositories like NVD.
- IV **Administrator terminal:** is the console used by the system administrator which receives the notification alert about the found cyber security vulnerabilities by the coordinator server in proposed system.

### 2.2 System Work flow

This section provides the proposed system's main work flow processes description and output for each stage. The following are the stages of the proposed system work flow:

**I Compile the computing assets information:** the system starts with the process of collecting the information about all organization's It assets by sending request for the install software agents on terminals or by monitoring the network, and the stored information softwares specifications and operating systems and network addresses and mapped to the proposed schema as shown in Fig 1. <sup>1</sup>

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<assetInfo version="1">
  <!-- SNIP -->
  <installedSoftwareInfo version="1">
    <softwareInfo>
      <name>Adobe Flash Player 23.0.0.205</name>
      <version>23.0.0.205</version>
      <publisher> Adobe Systems Software </publisher>
      <size>0x24e23</size>
      <installationDate>20161122</installationDate>
      <cpe id="cpe:/a:adobe:flash-player:23.0.0.205"
matchingRate="7.654244" />
      <cve id="CVE-2016-4273" />
      <cve id="CVE-2016-6982" />
    </softwareInfo>
    <!-- SNIP -->
  </installedSoftwareInfo>
  <networkInfo version="1">
    <hostName>ehab-qadah-pc</hostName>
    <openPorts>
      <port>444</port>
      <!-- SNIP -->
    </openPorts>
    <nicInfoList>
      <gateWay>131.220.207.254</gateWay>
      <ipAddress>131.220.198.146</ipAddress>
      <macAddress>255.255.240.0</macAddress>
      <subnetMask>255.255.252.0</subnetMask>
      <nicName> Network controller: Intel Corporation Wireless 3165 (rev 81)</nicName>
    </nicInfoList>
    <!-- SNIP -->
  </networkInfo>
</assetInfo>
```

**Fig. 1.** An example of the collected information by the proposed system for a terminal.

<sup>1</sup> based on figure 4 in [5]

- II Resolving of computing assets:** the system determines the CPE-IDs for the IT assets using the collected information from the first stage, the CPE dictionary <sup>2</sup> obtained from NVD repository is used as reference CPE-IDs for the proposed system. The basic algorithm to extract the corresponding CPE-IDs for the IT-assets is build a query from the collect information like name,version and owner and calculate the matching rate (percentage of the similar characters) for each cpe-id record in the CPE dictionary and select the CP-IDs with the highest matching rate and these CPE-IDs added to assets stored informations as seen in Figure 1.
- III Finding Vulnerability Information:** The system uses the determined CPE-IDs to query the vulnerability knowledge base for related vulnerabilities and in case of discovering of new vulnerability system's administrator notified by alert containing the cpe and cve data.

### 3 Standards

This section introduces on of the most related standards for the software vulnerability management which is the Security Content Automation Protocol(SCAP) [1].

#### 3.1 SCAP

SCAP is collection of open standards and enumerations of software flaws and configurations related to security allow the communication of that informations, to support the automated vulnerability and security information management.

It provided and maintained by The U.S. National Institute of Standards and Technology (NIST)[2], and the large and public of the SCAP content is the National Vulnerability Database (NVD)[3] that provides data feed for each SCAP standard which can be used free by the public security community, which also is managed by NSIT, SCAP has gained adaption by most of the information security systems. The SCAP standards consist of different open standards and enumerations, and the following subset of them:

- Common Vulnerabilities and Exposures (CVE)
- Common Platform Enumeration (CPE)
- Common Configuration Enumeration (CCE)
- Common Vulnerability Scoring System (CVSS)

The CVE is the naming list of software flaws in security context, to allow identification of cyber-security vulnerabilities to ease the exchange of information about the cyber-security issues, The CVE list is managed by The MITRE Corporation<sup>3</sup>. The CPE is the dictionary of all software product and applications,operating systems and hardware devices which provides identification and

<sup>2</sup> <https://nvd.nist.gov/cpe.cfm>

<sup>3</sup> <https://www.mitre.org/>

naming for each, the official CPE dictionary can be obtained from NVD repository <sup>4</sup> and provided in XML format. The CVSS is scoring system for the software flaw vulnerabilities, and The CCE is a dictionary of systems configuration and settings issues [4].

The CVE list is not a vulnerabilities data repository <sup>5</sup> is just names/IDs list, and NVD is one of vulnerabilities databases based on the CVE list and it provides a data feed for SCAP content such as the XML vulnerability feed which contains the cyber security software issues and it provides CVE link and a CVSS base score and CPE mapping for each vulnerability record [3].

## 4 Alternatives to NVD Repository

This sections describes the an alternative vulnerability data source to the traditional vulnerability repositories as National Vulnerability Database NVD for the vulnerability monitoring task, due the usual delay of revelation of cyber security related defects in by the typical information sources the detection of zero day vulnerabilities is difficult, in order to solve this issue an technique to collect the security information from totally new data source which is the Social Media (Twitter) was proposed in [6].

The proposed system takes the advantage of getting informed about security vulnerabilities information earlier than the normal information sources because of the widespread usage of social media platforms and technical blogs to discuss the software bugs and issues between the software develops and experts communities, in other hand the classical information sources like NVD waits the availability of patches before publish the vulnerability informations.

The introduced system is called SMASH( Social Media Analysis for security on HANA) consist of two subsystems data collection and processing. The data collection part is responsible to gather the security information from the social media( Twitter <sup>6</sup>) by searching the Twitter stream content to related security information and store it in local database to be utilized by the system later. The system also keeps a copy of the information form NVD to be used in recognize the new from the known vulnerability information.

The data processing subsystem is performing the extraction of security information form the stored twitter content to detect the zero-day vulnerabilities which are not published yet, by using various data mining techniques.

The proposed system offers the functionality of monotoning certain software products, the system shows the discovered security information related to user's selected softwares.

<sup>4</sup> <https://nvd.nist.gov/cpe.cfm>

<sup>5</sup> The list of vulnerabilities databases list can be found at [https://cve.mitre.org/compatible/product\\_type.htmlDatabase#Vulnerability](https://cve.mitre.org/compatible/product_type.htmlDatabase#Vulnerability)

<sup>6</sup> Twitter has been used in the prototype but the technique is applicable for the others social media platforms

8

**5 Discussion**

**6 Conclusion**



## References

1. <https://scap.nist.gov/> .
2. <https://www.nist.gov/> .
3. <https://nvd.nist.gov/> .
4. The Technical Specification for the Security Content Automation Protocol(SCAP) NIST Special Publication 800-126 Revision 3.
5. Takahashi, Takeshi, Daisuke Miyamoto, and Koji Nakao. "Toward automated vulnerability monitoring using open information and standardized tools." 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops). IEEE, 2016.
6. Trabelsi, Slim, et al. "Mining social networks for software vulnerabilities monitoring." 2015 7th International Conference on New Technologies, Mobility and Security (NTMS). IEEE, 2015.