Automated Software Vulnerability Management and Monitoring

Seminar Paper

MA-INF 3317 Selected Topics in IT Security

Version 1.0 draft

December 11, 2016

Ehab Qadah

Supervisor: Luis Alberto Benthin Sanguino

Table of Contents

Abstract. One of the main concerning areas for most organizations is software vulnerability analysis. In this paper, we discuss techniques and systems to automatically monitor software vulnerabilities using open standards and public vulnerability data repositories or alternative sources such social media and developer blogs.

1 Introduction

One of the main concerning areas for most organizations is software vulnerability analysis to ensure certain level of security, and this area is important due the continuous discovery of software vulnerabilities that open the doors for cyber-attacks. All organization must be aware of the known software vulnerabilities to perform the needed actions including installing the corresponding patches if available. Information about known software vulnerabilities can be retrieved from online repositories like the National Vulnerability Database (NVD).

1.1 Software Vulnerability Management

Vulnerability is defect or weakness in system result security incident or violation such as inappropriate access to system component or data breaches defined by G. Stonebumer et al. in [1].

The Software Vulnerability Management concept can be defined as the process of identifying related vulnerabilities in software that is installed inside an organization. The process first requires managing the inventory of the organization's IT assets and the periodic search for the related vulnerabilities. In case of the discovery of a vulnerability, a certain action should be performed such as alerting the system administrators or trying to install the corresponding patches 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 monitor the vulnerability of computing assets inside the organization/enterprise IT network infrastructure. their main contribution is automating the process of cyber security, vulnerability management, using open standards and tools to make available in a 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 sent to the system's 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 element type in the automated vulnerability monotoning 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 the 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 performs the check for the vulnerability presence by the communication with vulnerability databases.

III **Vulnerability Knowledge base**: is the local system's database for the vulnerability information, to compose 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 the 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 a 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. [1]

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 [2] 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 builds 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

[1] based on figure 4 in [5]

[2] `https://nvd.nist.gov/cpe.cfm`

```
<?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)</
        </nicInfoList>
        <!-- SNIP -->
    </networkInfo>
</assetInfo>
```

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

the CP-IDs with the highest matching rate and these CPE-IDs added to assets stored information 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) [3].

3.1 SCAP

SCAP is a collection of open standards and enumerations of software flaws and configurations related to security allow the communication of that information, 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 a data feed for each SCAP standard which can be used freely by the public security community, which also is managed by NSIT, SCAP has gained adoption 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[4]. The CPE is the dictionary of all software products and applications,operating systems and hardware devices which provides identification and naming for each, the official CPE dictionary can be obtained from NVD repository [5] 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].

[3] `https://scap.nist.gov/`
[4] `https://www.mitre.org/`
[5] `https://nvd.nist.gov/cpe.cfm`

The CVE list is not vulnerabilities data repository [6] 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 venerabilities is difficult, in order to solve this issue a 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 vulnerability 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 developers and experts communities, in other hand the classical information sources like NVD waits the availability of patches before publishing the vulnerability information.

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 [7]) by searching the Twitter stream content to related security information and store it in the local database to be utilized by the system later. The system also keeps a copy of the information form NVD to be used in recognizing the new from the known vulnerability information.

The data processing subsystem is performing the extraction of security information from 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 monitoring certain software products, the system shows the discovered security information related to user's selected softwares.

5 Discussion

The proposed technique and system in section 2 mainly focuses on the automation of software vulnerabilities monitoring process and using the open standards and public vulnerability repositories, and the action of the system in case of

[6] The list of vulnerabilities databases list can be found at `https://cve.mitre.org/compatible/product_type.htmlDatabase#Vulnerability`

[7] Twitter has been used in the prototype, but the technique is applicable for the others social media platforms

vulnerability discovery can be argued as simple and does not follow the main system focus to avoid the manual operation, it just sends an alert for the system administrator which then perform the need action, so the system should execute fast and immediate actions, such block the corresponding terminal or even the software component to ensure a certain level of security.

The main factor on the accuracy of the system's outcomes is the accuracy of the determination of CPE-IDs for all computing assets in the organization, the proposed systems find the CPE-IDs based on the numerical method of calculating the match rate, and this approach does not always give an accurate matching, and produced CPE-IDs should be validated by the system before moving to the further steps.

The proposed technique and system in section 4 is trying to detect the vulnerable Softwares as fast as possible by utilizing a different vulnerability information source social media (Twitter) by analyzing and exaggerate the security related content to be used by the detection of vulnerabilities task, which offers the opportunity to detect vulnerable softwares before event publishing the vulnerability information in the classical channels such as NVD, but this approach may utilize non validated or wrong information in the detection process that's why the proposed system consists of trust module of the extracted information from the social media data, since it the quality of data is very important to the system's output.

6 Conclusion

The importance of automatic detection and monitoring of vulnerable Softwares inside the organizations to ensure an efficient level of security and avoid cyber-attacks introduce the need software vulnerability management and monitoring systems as the two proposed system we discussed in this paper, those systems does not replace the normal security systems and policies they just complete their work. The role of vulnerability information repositories like NVD and the open standards such SCAP protocol is the main actor in building the software vulnerability management and monitoring systems.

References

1. G Stoneburner, A Goguen, and A Feringa, Risk Management Guide for Information Technology Systems, NIST Special Publication 800- 30, July 2002 http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf
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.