J Inf Process Syst, Vol.14, No.3, pp.740~750, June 2018
https://doi.org/10.3745/JIPS.03.0098

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Service Identification of Internet-Connected Devices Based on Common Platform Enumeration

Sarang Na*, Taeeun Kim*, and Hwankuk Kim*

Abstract

There are a great number of Internet-connected devices and their information can be acquired through an Internet-wide scanning tool. By associating device information with publicly known security vulnerabilities, security experts are able to determine whether a particular device is vulnerable. Currently, the identification of the device information and its related vulnerabilities is manually carried out. It is necessary to automate the process to identify a huge number of Internet-connected devices in order to analyze more than one hundred thousand security vulnerabilities. In this paper, we propose a method of automatically generating device information in the Common Platform Enumeration (CPE) format from banner text to discover potentially weak devices having the Common Vulnerabilities Exposures (CVE) vulnerability. We demonstrated that our proposed method can distinguish as much adequate CPE information as possible in the service banner.

Keywords

Common Platform Enumeration (CPE), Common Vulnerabilities and Exposures (CVE), OS Fingerprinting, Security Vulnerability Analysis, Service Identification

1. Introduction

Internet-connected devices are growing rapidly and diverse Internet of Things (IoT) services are being offered [1,2]. Gartner and IDC expect that there will be more than 20 billion IoT devices, excluding smartphones, tablets, and computers, by 2020 [3]. While numerous computing devices are constantly connected to the Internet, cyber security threats about hacking are also increasing [4]. Recently, hundreds of thousands of IoT devices, such as IP cameras, were infected by Mirai malware, which exploited a device's security vulnerability (e.g., default password) and they were used in DDoS attacks [5,6]. For these reasons, security experts need to find the vulnerable devices and take action as soon as possible.

Approximately one hundred thousand Common Vulnerabilities and Exposures (CVE) entries are provided by the National Vulnerability Database (NVD) [7,8]. The CVE system provides a reference method for publicly known security vulnerabilities. A CVE entry is composed of an overview of the vulnerability, Common Vulnerability Scoring System (CVSS), Common Platform Enumeration (CPE), Common Weakness Enumeration (CWE), etc. CVE entries provide the information about vulnerable

Manuscript received April 6, 2017; first revision June 15, 2017; accepted July 27, 2017.

Corresponding Author: Hwankuk Kim (rinyfeel@kisa.or.kr)

* Security Technology R&D Team, Korea Internet & Security Agency, Seoul, Korea ({no.1.nasa, tekim31, rinyfeel}@kisa.or.kr)

products in the CPE format, which gives the vendor, product name, version, etc. of software releases and hardware products. Thus, by matching a vulnerable product with the relevant CPE name, the device's vulnerability information can be easily found out.

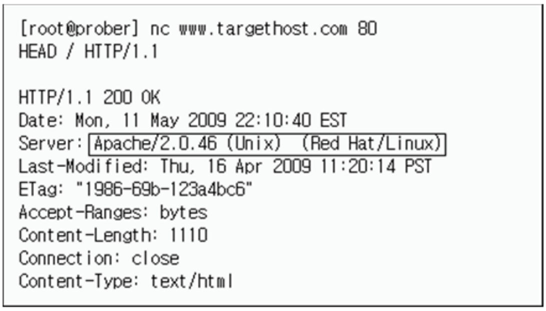
A computing device connected to the Internet is searchable through an Internet-wide scanning tool [9,10]. By scanning its open ports, e.g., HTTP and FTP, and grabbing the banner, the device's HW/SW information which can be used in discovering known security vulnerabilities can be known. It takes a long time to deal with all of the Internet-connected devices and is also inconvenient to identify the related vulnerabilities one by one. Therefore, we need an automatic technology to analyze the collected device information and convert it into the standardized format (CPE form) to easily identify the CVE vulnerability. Currently, it is difficult for the existing methods to automate the identification of the device information and express it as CPE name, just to provide device information in the raw data without security vulnerability information.

There is a limitation in identifying all of the device information of a network host based on the banner response acquired by Internet-wide scanning. Therefore, another analysis of the already-identified OS or application information is necessary to find out additional product information. In the case where just a device's application information (e.g., Internet Explorer 11) is known, the target OS information (e.g., Windows 10) can be additionally identified to find out the OS vulnerabilities. In this paper, we propose an automatic analysis method of identifying device information based on the CPE format to quickly cope with publicly known security threats. The first step is to identify device HW/SW information from service banner text and the next is to add the target OS or the installed application information of the device in order to discover more security vulnerabilities.

The remainder of this paper is organized as follows: we introduce the service banner and CPE dictionary in Section 2 and review the related work in Section 3. In Section 4, the proposed system is presented, and in Section 5, the experimental results are described. In Section 6, we discuss the coverage and limitations of our method. Lastly, this paper concludes in Section 7.

2. Background

2.1 Service Banner



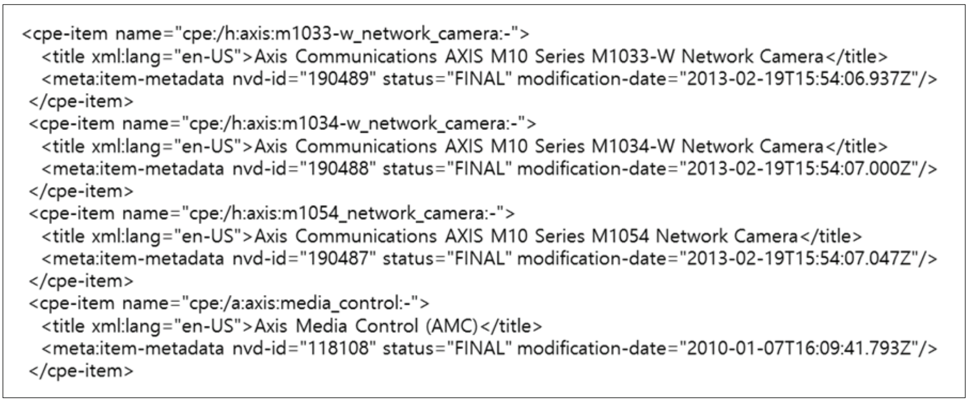
```
[root@prober] nc www.targethost.com 80
HEAD / HTTP/1.1

HTTP/1.1 200 OK
Date: Mon, 11 May 2009 22:10:40 EST
Server: Apache/2.0.46 (Unix)  (Red Hat/Linux)
Last-Modified: Thu, 16 Apr 2009 11:20:14 PST
ETag: "1986-69b-123a4bc6"
Accept-Ranges: bytes
Content-Length: 1110
Connection: close
Content-Type: text/html
```

Fig. 1. HTTP banner example including server information of the network host [11].

We can acquire the information about a computer system and its services through banner grabbing. For instance, if a network host uses a HTTP service port, the banner message can be obtained by sending a HTTP request [11]. A service banner is composed of free-form texts and has the information about the service running, including the server. By analyzing open port banner information, we are able to find out the OS/application information of an Internet-connected device, as shown in Fig. 1. Unfortunately, this information may be used by an attacker to find vulnerable systems that use vulnerable versions of operating systems and applications.

2.2 CPE Dictionary

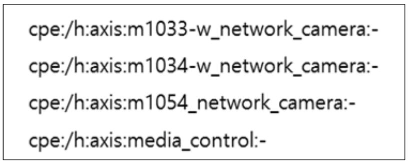
The CPE is a structured naming scheme for the released software packages and hardware products, and includes a formal name format [12]. The CPE dictionary that is given in the XML format provides a list of official CPE names (Fig. 2). CPE name is basically composed of part, vendor name, and product name/version, and also contains update, edition, target, and language information. It is expressed as cpe:/part:vendor:product:version:update:~edition~sw_edition~target_sw~target_hw~other:language [13]. For example, Internet Explorer 8.0.6001 beta made by Microsoft can be presented as cpe:/a:microsoft:internet_explorer:8.0.6001:beta.



```
<cpe-item name="cpe:/h:axis:m1033-w_network_camera:-">
  <title xml:lang="en-US">Axis Communications AXIS M10 Series M1033-W Network Camera</title>
  <meta:item-metadata nvd-id="190489" status="FINAL" modification-date="2013-02-19T15:54:06.937Z"/>
</cpe-item>
<cpe-item name="cpe:/h:axis:m1034-w_network_camera:-">
  <title xml:lang="en-US">Axis Communications AXIS M10 Series M1034-W Network Camera</title>
  <meta:item-metadata nvd-id="190488" status="FINAL" modification-date="2013-02-19T15:54:07.000Z"/>
</cpe-item>
<cpe-item name="cpe:/h:axis:m1054_network_camera:-">
  <title xml:lang="en-US">Axis Communications AXIS M10 Series M1054 Network Camera</title>
  <meta:item-metadata nvd-id="190487" status="FINAL" modification-date="2013-02-19T15:54:07.047Z"/>
</cpe-item>
<cpe-item name="cpe:/a:axis:media_control:-">
  <title xml:lang="en-US">Axis Media Control (AMC)</title>
  <meta:item-metadata nvd-id="118108" status="FINAL" modification-date="2010-01-07T16:09:41.793Z"/>
</cpe-item>
```

Fig. 2. CPE dictionary in the XML format.

We extracted the CPE name from the dictionary file, as shown in Fig. 3 and generated search keywords in the form of a CPE tree structure by analyzing the CPE component.



```
cpe:/h:axis:m1033-w_network_camera:-
cpe:/h:axis:m1034-w_network_camera:-
cpe:/h:axis:m1054_network_camera:-
cpe:/h:axis:media_control:-
```

Fig. 3. CPE names from the CPE dictionary.

3. Related Work

We can use ZMap and Nmap tools for Internet-wide scanning. They scan a network host and find the vulnerable system in regards to specific vulnerabilities [14,15]. Tenable Network Security developed the

Nessus vulnerability scanner, which identifies the OS information of a network host and assesses the vendor's known vulnerabilities and misconfiguration weaknesses [16]. They are able to analyze known vulnerabilities precisely, but their process for actively probing the device is time consuming. In contrast, our proposed system rapidly obtains the device information and matches it with publicly known vulnerability information collected from NVD by passively analyzing network hosts.

John Matherly is operating a device search service called Shodan to retrieve Internet-connected device information acquired by its Internet-wide scanning [9]. Shodan provides response information such as TCP/IP handshake and banner message for open service ports. Durumeric et al. [10] proposed a search engine called Censys which can analyze potentially vulnerable devices. Censys can identify the partial vulnerability information of Heartbleed, Poodle, etc., and also provide device information including banner text for fourteen protocols. Censys is available for free using its search interface. It is inconvenient for security experts to manually convert device information into CPE using these systems because they simply deliver a service banner without processing its texts. However, our proposed method is able to provide a device's service banner and identify its HW/SW information, such as OS or application.

Genge and Enachescu [17] proposed a vulnerability assessment method called ShoVAT. ShoVAT analyzes device information acquired from Shodan API and identifies the CPE name and CVE entry information of a vulnerable device. Due to its mechanism of creating the CPE name based on the device's version pattern ("Integer.Integer.Integer"), ShoVAT has a limitation in identifying several CPE candidates and also may be incorrectly matched if the product name is located far away from the product's version number. Our proposed method can correct CPE information through an entire keyword search for vendor and product names within banner texts and can then look up the version number based on the CPE dictionary. It can further identify the installed software information related to the already-identified CPE information.

4. Proposed System

We have designed a system for identifying a device's HW/SW information from banner texts and generating its CPE name [18]. Moreover, the way that supplementary software information is added (e.g., target OS or an installed application) was studied so that more security vulnerabilities can be discovered.

4.1 Concept

Fig. 4 shows the concept of our proposed method. We generated a CPE tree that is used in the keyword search based on the CPE dictionary. Then, the keyword analysis for the pre-processed banner texts was conducted and the device's most relevant HW/SW information was returned in the CPE format.

4.2 Device Information Identification Based on the Banner

4.2.1 CPE tree generation

We generated a CPE tree based on the CPE dictionary to create the keyword list used in the banner text analysis. As shown in Fig. 5, the CPE tree is made up of a total of 6 levels according to the CPE format. Level 1 is comprised of vendor name and Level 2 has its product information as a child node. Likewise, levels 3, 4, 5, and 6 are built in the same manner. In case of level 5, one node can include various items by separating the character strings with the "~" character. For each level, the character strings are separated with the "_" character. For example of "cpe:/a:microsoft:internet_explorer:11", level 1 is Microsoft; level 2 is internet, Explorer; level 3 is 11; and the information on levels 4, 5, and 6 is absent.

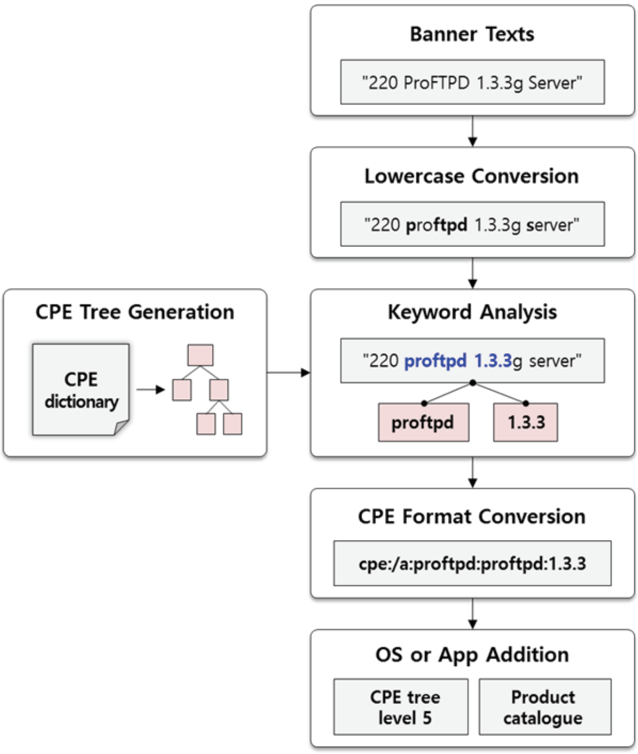


Fig. 4. Concept of the proposed method.

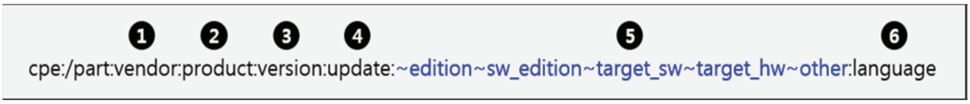

1
2
3
4
5
6
cpe:/part:vendor:product:version:update:~edition~sw_edition~target_sw~target_hw~other:language

Fig. 5. Composition of the CPE name.

4.2.2 Banner keyword analysis

First, the banner texts are converted to lowercase to facilitate keyword analysis. Next, the keyword analysis for the banner is conducted using a character string at each level of the CPE tree, as described in Fig. 6. The name of the vendor or product is extracted for in levels 1 and 2 by running the entire keyword search and then the version information is identified in levels 3, 4, 5, and 6. In contrast with levels 1 and 2, levels 3, 4, 5, and 6 only search for the keyword related to the searched product. For version information, the substring combination (e.g., version 4.6.1) that can be separated with the "." character is taken into account to figure out the higher version (e.g., version 4 or 4.6).

4.2.3 CPE format conversion

The extracted keywords are converted into the closest CPE name from level 1 to level 6. Among the possible CPE candidates from the keyword combination, the CPE name with the longest character string is selected. Since two or more CPEs can exist, if there is another keyword other than the keyword that uses the longest character string, the CPE can be additionally created. Through the abovementioned process, as much device information in the CPE format as possible is identified and this is finally used for the CVE vulnerability information analysis.

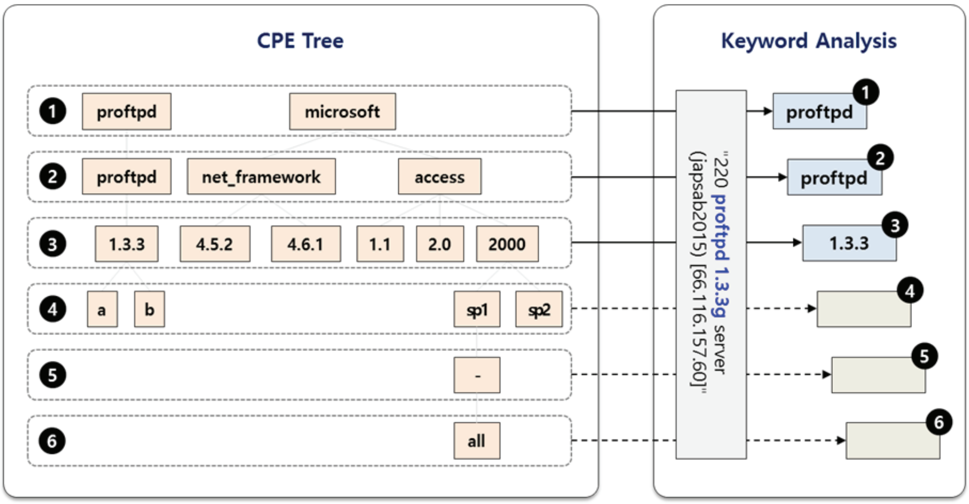


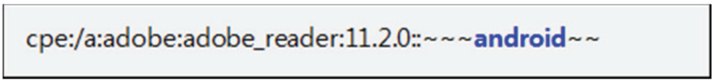
Fig. 6. Keyword analysis for the processed banner texts.

4.2.4 Prototype system

We implemented a prototype system of our proposed method and obtained banner data from our collection system using Censys API [10]. This system processes the banner texts of the most common service protocols (e.g., HTTP, FTP, SSH, and so on) by converting it into lowercase letters. The CPE tree is pre-created by parsing the recent CPE dictionary from NVD. Additionally, the function that measures the number of discovered CPE names per device to analyze the precision of this method was implemented.

4.3 Target OS Identification

In cases where only the application information of a device is known, its OS information can be additionally identified at CPE tree level 5: the information target_sw, target_hw. As shown in Fig. 7, we can find out the target OS by acquiring target_sw information in the CPE name without the additional analysis process.



cpe:/a:adobe:adobe_reader:11.2.0::~~~**android**~~

Fig. 7. Instance of target OS information.

4.4 Installed Application Identification

In cases where only the OS information of a device is known, its application information installed in the system can be additionally identified through an analysis of the product catalogue provided by the manufacturer's database. We can automatically structure the product list of the specific application supported by the OS or the OS installed application and then identify the installed product information. For instance of Fig. 8, Internet Explorer which is a Microsoft Windows application can be identified.

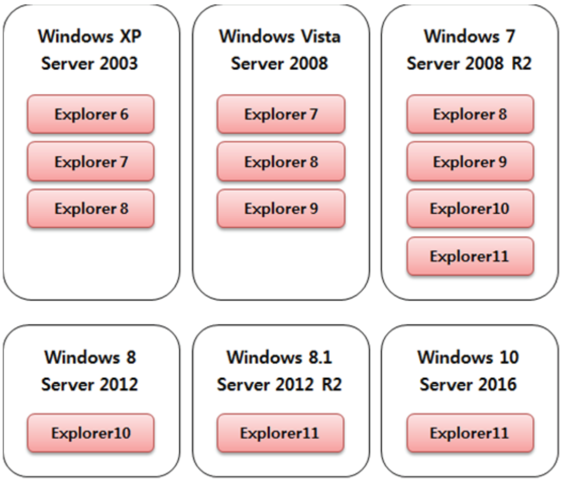


Fig. 8. Instance of installed application information.

5. Results

It was verified that how much product information in the CPE format can be identified using our proposed scheme. To do so, we analyzed the accuracy of identifying the correct hardware or software CPE information from the service banners of Internet-connected devices. This study aimed to identify as many CPE names as possible from a banner.

5.1 Experimental Data

We utilized our collection system and the Censys search engine in acquiring banner data for nine protocols. For each protocol, 10,000 banners were collected from Internet-connected hosts. From amongst all of these, the arbitrary 50 network hosts were selected for each service protocol and their service banners were used in the accuracy analysis experiment.

5.2 Experimental Results

5.2.1 Accuracy analysis

We measured the accuracy rate for identifying the appropriate CPE names in the banner. It was manually analyzed that if the HW/SW information of a device was correctly distinguished in right CPE format. Table 1 describes the results of this experiment. The average rate for the accuracy analysis was 98.9%. The accuracy rate of the proposed method was higher than ShoVAT (about 92.3%). This is because that our system can create multiple CPEs without duplication for different keywords by conducting full keyword search for the banner texts and generating all possible keyword combinations (including version information) regardless of the word location in the banner. The other existing methods, such as Shodan and Censys, have a limitation for identifying the device information since the number of their searchable devices is small (or unknown).

Table 1. Accuracy rate (%) for identifying CPE names

FTP	SMTP	SSH	IMAP	IMAPS	POP3	CWMP	HTTP	TELNET
98	100	94	100	100	100	100	98	100

The accuracy rate was high, but there was a lot of over matched CPE information. In the case of CWMP, SSH, and HTTP, the banner text was composed of similar forms and generally relevant HW/SW information. For SMTP, there were many words about the website address and a little HW/SW information. For IMAP and IMAPS, lots of banner texts were made up of commonly used words to disrupt the identification of CPEs. These words may lead to keywords being over matched. Overall, there are a number of keywords that are commonly used in the access page of the remote server, such as “server,” “login,” and “administrator.” It is very hard to distinguish these product names from the service banner. If there is a unique product name, our system is able to discover better its CPE name. In this paper, we focused on identifying a lot of CPEs from the service banners of an open port. Although there are many over matched CPE names, it is reasonable for analyzing the HW/SW information in order to identify a vulnerable device.

5.2.2 OS information analysis

A lot of service ports provide OS information in the service banner. Thus, we analyzed the service banner, including OS information, which is most common OS, such as Linux, Ubuntu, and Debian. While there are service banners that have keywords associated with the OS, it was inspected that how many CPE names could be identified. We selected 50 random sample banners with OS information and manually analyzed their software information. In this experiment, the accuracy rate of the entire target OS was approximately 100% because those OS names are distinguishable from general sentences in processed banners. Comparatively, these banners are made up of useful words for identifying the CPE name than those that have application information. In addition, we analyzed the service banner, including web server information, such as Apache, Nginx, and Lighttpd, and the accuracy rate was also high (about 100%).

5.2.3 Execution time

The device information analysis time of our proposed method was measured for each protocol and the average execution time was 1.1 seconds (Table 2). In the case of POP3 and SSH, the execution times were faster than the other protocols. In contrast, it took a little more time for FTP to complete the analysis work. The entire analysis time for current Internet-connected devices was about 5 hours in the simulation regarding the number of open ports and is presented in Table 3.

Table 2. Execution time (second) per 10,000 banners

FTP	SMTP	SSH	IMAP	IMAPS	POP3	CWMP	HTTP	TELNET
2.2	0.99	0.46	0.73	0.69	0.3	1.61	0.93	1.61

Table 3. Total analysis time (hour) for Internet-connected devices

FTP	SMTP	SSH	IMAP	IMAPS	POP3	CWMP	HTTP	TELNET
0.41	0.17	0.19	0.09	0.08	0.04	1.87	1.88	0.33

6. Discussion

6.1 Coverage

Our prototype system allows for the sorting of identified CPE names. In other words, the CPEs are sorted in descending order and the CPE with the highest accuracy rate is represented at the top of the whole CPE list. It is efficient for discovering the best CPE name as CPEs have duplicate names but individual vendors can be identified. This system also provides matching information, which shows how a CPE name and the searched keyword in the banner are related.

Moreover, our system includes a vulnerability analysis feature that associates a device with the corresponding vulnerability information (i.e., CVE identifier number). If the CPE name is identified, its related vulnerabilities can be automatically listed. Thus, the accuracy rate of CPE identification means that the accuracy rate of a device's vulnerability identification.

6.2 Limitations

Unless the vendor or product name is unique, it is hard to distinguish the CPE name from service information. For instance, there are short names with less than three characters or that consist of commonly used words. There are even some cases where the word spacing in the banner itself is not correct. In the case of the CPE name for an application, it is more difficult to discover than OS identification because there are a huge number of applications and they have various names.

To cope with this problem, we aimed to discover as many CPEs as possible from service banners and created the CPE name list in matching probability order. It needs more efforts to analyze a lot of patterns for banner texts by each service protocol in order to achieve greater accuracy. It is also important to analyze banner texts with a single approach in order to conduct keyword searches that are independent against the service protocol type.

7. Conclusion

The device management of a network host should be taken into account in order to prevent hackers from exploiting known vulnerabilities so as to conduct a large-scale cyberattack. To deal with this, we have proposed an automatic analysis method of identifying device information and its known CVE entries through conducting the keyword analysis for banner texts based on the CPE dictionary. The hardware and software information of an Internet-connected device can be obtained in the CPE format, and also the target OS or the installed application information can be identified to find out as many vulnerabilities as possible. It was demonstrated that our proposed scheme can provide a proper CPE list of the device information with the matching process. In the future, we will expand the target of software information analysis and make further researches that discover the actual OS or application installed in a network host.

Acknowledgement

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (No. 2016-0-00193, IoT Security Vulnerabilities Search, Sharing, and Testing Technology Development).

References

[1] S. Maity and J. H. Park, "Powering IoT devices: a novel design and analysis technique," *Journal of Convergence*, vol. 7, article no. 16071001, 2016.

[2] R. Mafrur, I. G. D. Nugraha, and D. Choi, "Modeling and discovering human behavior from smartphone sensing life-log data for identification purpose," *Human-centric Computing and Information Sciences*, vol. 5, article no. 31, 2015.

[3] A. Nordrum, "Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated," 2016 [Online]. Available: https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated.

[4] J. W. Joo, J. K. Lee, and J. H. Park, "Security considerations for a connected car," *Journal of Convergence*, vol. 6, no. 2, pp. 1-9, 2015.

[5] Mirai (malware) [Online]. Available: https://en.wikipedia.org/wiki/Mirai_(malware).

[6] Z. Wikholm "When vulnerabilities travel downstream," 2016 [Online]. Available: https://www.flashpoint-intel.com/blog/cybercrime/when-vulnerabilities-travel-downstream/

[7] CVE details: the ultimate security vulnerability datasource [Online]. Available: https://www.cvedetails.com.

[8] National Vulnerability Database [Online]. Available: https://nvd.nist.gov.

[9] Shodan (website) [Online]. Available: https://en.wikipedia.org/wiki/Shodan_(website).

[10] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by Internet-wide scanning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, Denver, CO, 2015, pp. 542-553.

[11] Banner grabbing [Online]. Available: https://en.wikipedia.org/wiki/Banner_grabbing.

[12] Official Common Platform Enumeration (CPE) dictionary [Online]. Available: https://nvd.nist.gov/cpe.cfm.

[13] B. A. Cjeoles, D. Waltermire, and K. Scarfone, "Common Platform Enumeration: Naming Specification Version 2.3," National Institute of Standard and Technology (NIST) Interagency Report No. 7695, 2011.

[14] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: fast Internet-wide scanning and its security applications," in *Proceedings of the 22nd USENIX Security Symposium*, Washington, DC, 2013, pp. 605-619.

[15] G. F. Lyon, *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Sunnyvale, CA: Insecure, 2009.

[16] Nessus (software) [Online]. Available: https://en.wikipedia.org/wiki/Nessus_(software).

[17] B. Genge and C. Enachescu, "ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services," *Security and Communication Networks*, vol. 9, no. 15, pp. 2696-2714, 2016.

[18] S. Na, T. Kim, and H. Kim, "A study on the service identification of Internet-connected devices using common platform enumeration," in *Advanced Multimedia and Ubiquitous Engineering*. Singapore: Springer, 2017, pp. 237-241.

Sarang Na https://orcid.org/0000-0003-2238-4394

She received B.S. and M.S. degrees in Computer Science and Engineering from Sejong University, Seoul, Korea, in 2011 and 2013, respectively. Currently, she is a researcher in Security Technology R&D Team at KISA and a PhD candidate in Graduate School of Information at Yonsei University, Seoul, Korea. Her research interests include mobile security, IoT security, vulnerability information analysis, and machine learning.

Taeeun Kim https://orcid.org/0000-0002-2558-8029

He received B.S. degree in Computer Engineering from Cheonan University and M.S. degree in Computer Engineering from Soongsil University, Korea, in 2005 and 2007, respectively. Currently, he is a deputy general researcher in Security Technology R&D Team at KISA and a PhD candidate in Computer Engineering at Soongsil University, since 2007. His research interests include network security, mobile authentication, etc.

Hwankuk Kim https://orcid.org/0000-0002-4449-5821

He received B.S. and M.S. degrees in Computer Engineering from Hankuk Aviation University, Korea, in 1998 and 2001, respectively and his Ph.D. in the graduate school of Information Security at Korea University, in 2017. He was as a researcher at ETRI until 2006. Currently, he is a team manager in Security Technology R&D Team at KISA. His research interests include ISMS, IoT vulnerability analysis, wireless network security and its application, etc.