J Ambient Intell Human Comput
DOI 10.1007/s12652-017-0581-6

ORIGINAL RESEARCH

Management platform of threats information in IoT environment

Eunhye Ko[1] · Taeeun Kim[1] · Hwankuk Kim[1]

Received: 6 March 2017 / Accepted: 11 September 2017

Abstract Recently, the use of IoT devices in living environments has increased due to the development of various wireless communication technologies. As the number of types of IoT devices has grown exponentially, many kinds of insecure operating systems and open source software programs are being used. As they run with security vulnerabilities, IoT devices using such OS and software can be targeted by malicious attackers. In addition, due to the operating characteristics of IoT devices, it is difficult to apply security patches immediately when vulnerabilities are found. Accordingly, there is an increasing need for managing and sharing cyber security threat information in order to prevent security threats and accidents. This paper suggests a platform structure and application method for collecting, analyzing and sharing vulnerability information about IoT devices.

CCS Concepts: Security and privacy→System security→Vulnerability management.

✉ Hwankuk Kim
rinyfeel@kisa.or.kr

Eunhye Ko
keh@kisa.or.kr

Taeeun Kim
tekim31@kisa.or.kr

[1] Korea Internet and Security Agency, 9, Jinheung-gil, Naju-si, Jeollanam-do 58324, Korea

Keywords Vulnerability information · Structured threat information expression (STIX) · Internet wide scan · IoT devices vulnerability correlation mapping · Management platform of threats information

1 Introduction

Given the recent activation of Internet of Things (IoT) services in various fields, the number of devices connected to the Internet such as smart home appliances, CCTVs, and wearable devices is increasing rapidly. According to a survey by Gartner in 2013, it is estimated that more than 20.8 billion IoT devices will be connected to the Internet by 2020 (Maity and Park 2016; Middleton et al. 2013). In contrast with the increasing supply of IoT, the level of security is very low, and cyber-attacks exploiting vulnerabilities of IoT devices are increasing (Berhanu et al. 2013).

According to Cisco's survey about devices with security vulnerabilities in 2016, network products such as routers and switches have an average of 28 vulnerabilities per device. In addition, 23% of devices connected to the Internet were operating with vulnerabilities noticed 5 or 6 years ago, and even 10% of those devices had vulnerabilities identified more than 10 years ago (Joo et al. 2015). Because most of all users don't access them directly after installation and activation, IoT devices including these network products are not being managed for security vulnerabilities.

Common features of unmanaged IoT devices are as follows. First, immediate security patches are difficult when vulnerabilities are found. Because it is inconvenient to update the embedded operating system (OS) or firmware. Second, they are operated with vulnerabilities as they use old wireless communication technologies, OSs, and open source software. In such an operating environment, there is a

Published online: 23 September 2017

need for a technology that can rapidly check multiple devices to find known vulnerabilities. To prevent cyber-attacks, it is necessary to compensate for the vulnerability of devices by sharing analysis information in a standardized way.

This paper analyzes examples of cyber-attacks that exploit vulnerable IoT devices and proposes a platform for collecting, analyzing, and sharing vulnerability information about these devices. In Sect. 2, the demands of the proposed technology are deduced by analyzing examples of abusing the vulnerability of IoT devices, and in Sect. 3, the technological aspect of analyzing device vulnerability information is described. In Sect. 4, the paper proposes a platform to collect, analyze, and share vulnerability information about IoT devices. In Sect. 5, the performance of the proposed technology is analyzed, and in Sect. 6 the results of this study are described.

2 Security threats in IoT environment

Compared to the wide-spread of services with IoT devices, the IoT environment is lacking management and technology for security. Due to this characteristic, there are many threats in IoT environment.

The threats could be classified into three categories according to the location where it positioned. The first type of threats is in sensor and device. Due to the IoT devices are low specification, security technology is difficult to apply. Also, due to the characteristics of the IoT environment, security patches and monitoring are difficult. The second type is in network. Wireless networks are difficult to maintain level of security, because it is needed to interconnect with other networks. The last one is in platform and service. Platforms and services usually use open source, so that they always exposed to threats.

As mentioned above, an attacker can easily bypass this authentication system and exploit the devices to launch distributed denial of service (DDoS) attacks by infecting them with malicious codes. In Sect. 2, the threats in IoT environment and some attacks are shown. Also, the necessity of technology preventing security threats by searching for the existence of vulnerabilities is described, as in Fig. 1 shown.

2.1 Weak authentication system of routers

Lately there was a massive infection of malicious code, that infection was abused the threat of IoT devices.

The routers of Deutsche Telekom, a German Internet service provider (ISP), were infected with a malicious code, causing about 900,000 users to suffer problems with their Internet connection. The attack attempted to infect the routers of the entire line with the malicious code but it failed. In the process of infection, the service was disrupted. The malicious code that caused the disruption was an upgraded version of "Mirai". The infection process of this malware was designed to utilize a weak authentication system with default logins and passwords. Then it propagated itself by searching and infecting network devices with open ports used for managing the firmware of routers and diagnosing the devices.

2.2 Infection with weak wireless protocol

The wireless network for IoT devices could be vulnerable, because it is needed to interconnect with heterogeneous devices. For this reason, weak communication protocols were used. Some attacks abusing vulnerabilities of weak protocol especially for managing devices.

Customer-premises equipment WAN management protocol (CWMP) is to monitor and configure routers. Because this protocol is for managing devices, it allows access and control from remote sites. This property makes easy to form a large botnet composed of IoT devices.

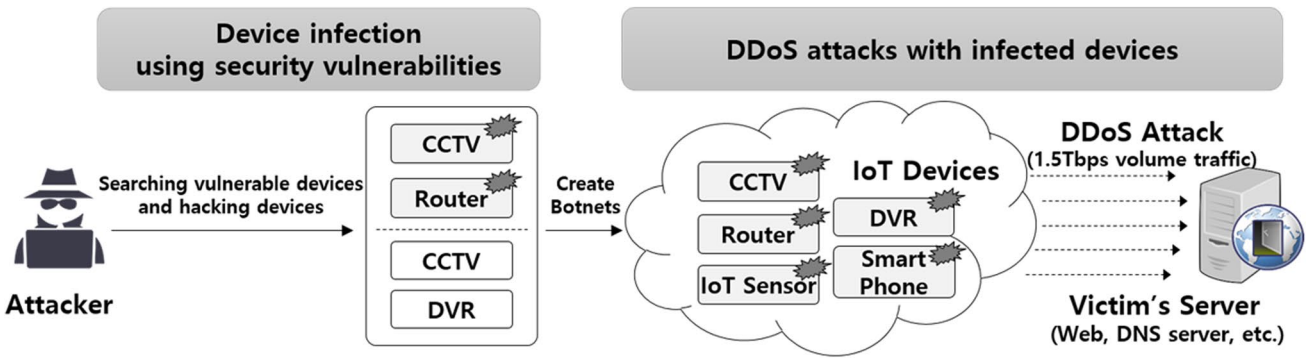

Fig. 1 Description of DDoS attack with vulnerable IoT devices

2.3 DDoS attacks through a botnet composed of IoT devices

A DDoS attack was launched against the servers of Oles Van Herman (OVH), the world's third largest web hosting company in France. This attack used 145,000 camera digital video recorders (DVR) to generate and transmit 1–30 Mbps traffic per IP on average (a total of 1.5 Tbps). The attack was executed after forming a botnet using a large number of CCTVs that were hacked through vulnerabilities.

Meanwhile, there is a large-scale DDoS attack on Dyn, a DNS provider of the United States. It was caused by IoT devices with weak authentication and that devices were infected with malicious programs. The DDoS attack made stop providing Internet service, one of the most important functions of DNS provider, and it resulted in the malfunctioning of more than 1200 large sites including Twitter and Netflix. The attack came from a botnet of IoT devices infected with a malicious code called "Mirai".

Most IoT devices that require frequent access by users, such as CCTVs, do not have security settings or use default passwords for the convenience of users. An attacker can easily bypass this authentication system and exploit the devices to launch DDoS attacks by infecting them with malicious codes.

2.4 The need for technology to prevent security threats

The need for technology to prevent security threats can be confirmed by observing the two aforementioned examples of exploiting the vulnerabilities of IoT devices (Becsi et al. 2015). First, IoT devices do not have enough security functions or lack management for ID/PW. Second, as malicious codes such as "Mirai" evolve, setting ID/PW cannot defend against attacks that exploit the vulnerabilities of the firmware itself.

Therefore, it is necessary to have a system and platform that can search device information, analyze vulnerability information, and share threat information so that devices can be managed safely against security threats (Cisco 2016; Stock et al. 2016).

3 Related work

3.1 Scan technology for Internet devices

The current network scan technologies identify the type of OS by checking the IP of the internal network and check for vulnerabilities by collecting information about the type and version of the service through scanning ports. To identify vulnerabilities, a handshake scan is performed using tools such as NMap, Nessus, etc. These scan technologies perform the function of checking for vulnerabilities by executing attack methods. But given the increasing need for technology that quickly and remotely collects information about devices connected to the Internet, passive scan technologies have recently been undergoing development.

Passive scan technologies collect information about devices by sending and receiving normal communication messages without performing attack methods. In addition, the devices targeted for collecting information are not those in the internal network, but all devices connected to the Internet. These technologies aim to quickly collect information such as the service banner and traffic header. As shown in Table 1, these scan technologies have different characteristics and methods to analysis.

John Matherly developed the Shodan search engine to search for information about devices connected to the Internet through a passive scan. Shodan scans open ports such as HTTP, FTP, and TELNET through the handshake process (Brown et al. 2015; Lee et al. 2017; Serrano et al. 2014). The device information is identified by analyzing with keywords contained in the banner. By supporting various protocols, Shodan collects the largest amount of information for any individual device. The engine can also search for the existence of vulnerabilities such as Heartbleed and Poodle by collecting information about whether or not to use SSL cryptographic algorithm as well as the version information.

Table 1 Comparison of network scan techniques

	Handshake scan (fuzzing)	Passive scan
Characteristic	Authorized user→credential scan (default PW, authority, fuzzing, etc.)	Unauthorized user→non credential scan (normal messages like system banner)
Scan target	Devices in internal network	Devices connected to the Internet
Vulnerability analysis	DeviAnalysis for known/unknown vulnerabilities (network/service/code-level analysis)	Analysis for known vulnerabilities (after building DB with information about devices)
Method of analysis	Dynamic/static fuzzing analysis	Utilizing vulnerability database
Related tools	Defensics, Nessus	Shodan/ShoVAT, Zmap/Censys

Meanwhile, Durumeric developed Censys, a search engine that can quickly scan devices connected to the Internet (Bodenheim et al. 2014). Censys was developed based on ZMap and ZGrab, which are open source. It collects port information for each of the 12 major protocols such as HTTP and POP3. Censys also provides device information such as banner and protocol header, as well as vulnerability information about the use of SSL cryptographic algorithm as Shodan does.

Differences between Shodan and Censys include the update interval for scanned device information and the time required for scanning. Shodan has an update interval of 1 month as it collects port information more than Censys, and Censys updates its major port information in 2 weeks. Censys, on the other hand, can check the "alive" condition of devices in 1:09:45 s when scanning devices in all IPv4 address bands, about 4.3 billion using a single probe (Huh and Seo 2016).

It is possible to obtain OS and application information through the banner and protocol header provided by Shodan and Censys. But such information simply provided keywords from the banner information. Therefore, additional analysis is required to obtain common platform enumeration (CPE) information for analyzing the correlation with vulnerability information (Genge et al. 2015).

3.2 Technology of sharing information on security threats

As the number of search engines like Shodan and Censys that can scan device information spread throughout the Internet increases, there is a growing need to share information to neutralize security threats and prevent accidents. To quickly and automatically share the information on cyber threats, several organizations have developed and are using standards of information. According to the type of information, these standards can be divided into two groups. The first group is used to express threat information, and the other is for describing intrusion indicator.

Threat information is comprehensive analysis information related to the attack or threat that includes strategies or tactics that are used, and motivation for the attack. On the other hand, information about intrusion indicator includes hash information and the registry of the files related to accidents. The most frequently used standards for sharing information today are Structured Threat Information eXpression (STIX) and Open Indicator Of Compromise (OpenIOC).

STIX is a standard for expressing threat information such as vulnerabilities, incidents, and related events, while OpenIOC is a standard for describing intrusion indicators such as detailed information about files and traffic (Durumeric et al. 2015; Genge et al. 2015).

OpenIOC can express intrusion indicators such as the traffic information for tracking, the hash value of the files, the rules for firewalls, intrusion detection system (IDS), and intrusion protection system (IPS). However, it focuses on describing observation information and thus has difficulties in describing detailed information about threats. Meanwhile, STIX is a standard developed by the US Department of Homeland Security (DHS) in conjunction with MITRE to build an efficient and secure information-sharing system for responding to cyber threats. As shown in Table 2, STIX can be used to structure and describe information about the accident, vulnerability, observed event and etc. As shown in Table 2, STIX consists of eight components for describing all kinds of information. Trusted Automated eXchange of Indicator Information (TAXII) is an automatic transmission standard for sharing the cyber threat information described by STIX in real time (Barnum 2012). TAXII provides services such as Push, Pull, Discovery, and Feed Management. By using each service, it can request and transmit information between producers and consumers of information.

4 Technology for searching vulnerability information

In this chapter, a proposal is made for a platform for managing threat information. This platform can collect, analyze

Table 2 Elements of STIX

Element	Description
Observable	What has been or might be seen in cyberspace
Indicator	Patterns for what might be seen and what they mean if they are
Incident	Instances of specific malicious actions
TTP	Attack patterns, malware, exploits, kill chains, tools, infrastructure, victim targeting, and other methods used by an attacker
Exploit target	Vulnerabilities, weaknesses, or configurations that might be exploited
Couse of action	Actions that may be taken in response to an attack or as a preventative measure
Campaign	Sets of incidents and/or TTPs with a shared intent
Threat actor	Identification and/or characterization of the attacker

and share information of device and vulnerability to eliminate security threats for IoT devices connected to the Internet. The proposed platform is designed to prevent vulnerable devices from suffering cyber threats and accidents by combining the individual technologies mentioned above (Ring 2014; Oehmen et al. 2015).

4.1 Composition of the proposed technology

The platform for searching security vulnerability information for IoT devices consists of a collection system for device information, an analysis system for vulnerability information, and a sharing system for threat information, as shown in Fig. 2.

In general, operating the platform consists of three stages. The collection system scans the devices connected to the Internet and collects detailed information about them. Next, the analysis system matches and analyzes vulnerability information about scanned devices received from the collection system. Finally, the sharing system shares the analyzed information about device vulnerability with users and institutions that need such information (Ionita et al. 2016). It also manages information about changes in history through a database that stores the history of device information collected on an IP basis.

Each system operates asynchronously on a separate server based on its own operating cycle.

4.2 Technology of collecting device information

To collect the information about devices connected to the Internet, a module that performs three functions was constructed. And each of functions makes to collect information of device, as shown in Fig. 3.

a. First is the "IP alive scan" module, which creates a list of IP addresses to be scanned, generates scan packets, and sends them to (or receives them from) the corresponding IP. This module requires the core functions of generating an IPv4-based address list and managing black/white lists. It also requires the function of managing large-capacity packets because devices must be scanned at high speed. The technology of generating IP addresses is very important in Internet-based scan technology. When network scan traffic is generated sequentially, the scan function is not available due to detection by security devices such as firewalls, IDS, and IPS. Therefore, it is necessary to have a technology that randomly generates lists of scan addresses. To develop the proposed technology, technologies for generating lists of IP addresses such as sequential generation, BGP table reference, and random algorithms were compared in terms of randomness and coverage rate. Randomness indicates the ability to scan without being detected by security devices while the IP coverage rate indicates the percentage of all IPv4 addresses covered by the generated list of IP addresses.

As shown in Fig. 4, the use of random algorithms for randomness does not result in high IP coverage rate depending on the nature of the random function. On the other hand, the sequential generation method is not random, and the method of referring to the BGP table requires a great deal of additional calculations to analyze and parse the relevant data. Therefore, to guarantee both randomness and IP coverage rate, the proposed technology generates a random IP list in which IP addresses are converted into decimal numbers and then divided and circulated. The algorithm complexity of the proposed method is O(n), and it shows an IP coverage rate of 100% because it creates a list of IP addresses in the same way as sequential generation.

b. Second is the "handshake scan" module, which collects service (port) information about device usage through

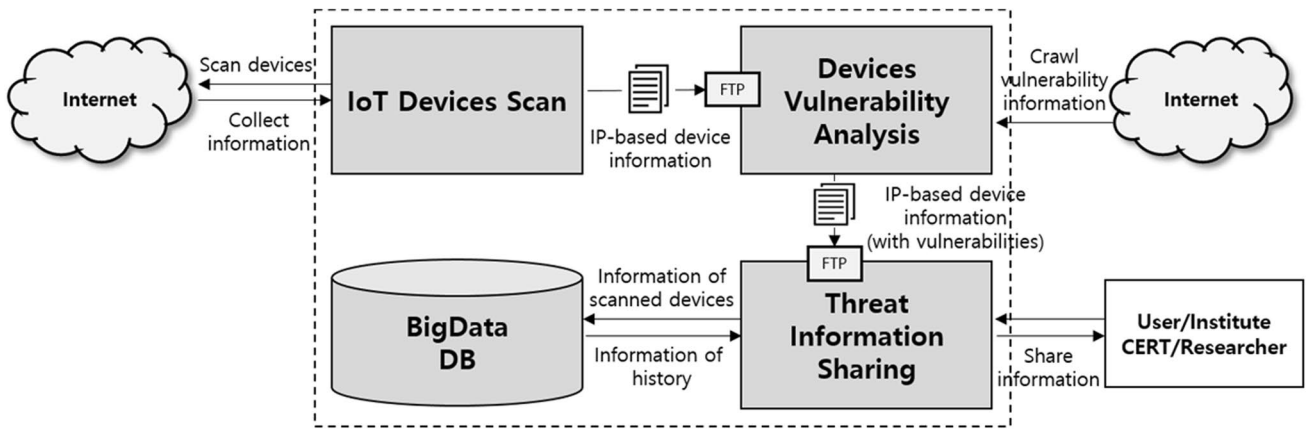

Fig. 2 Composition of the proposed platform

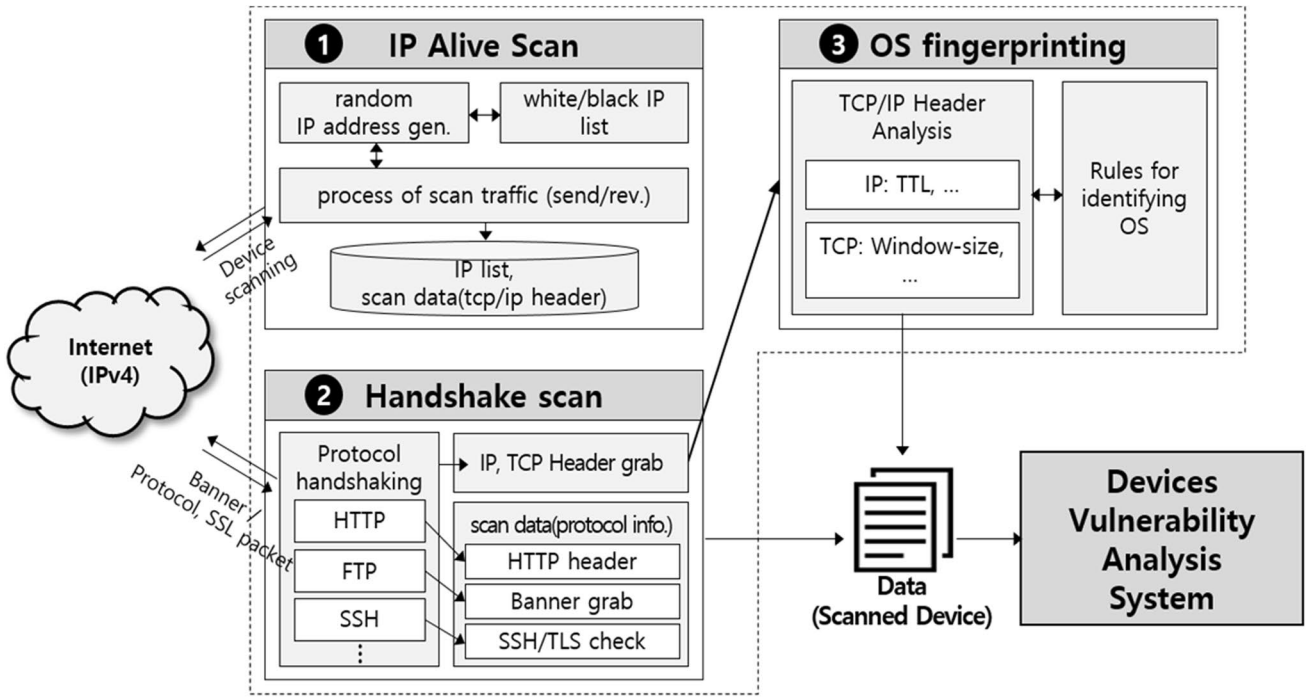

Fig. 3 The process of collecting device information

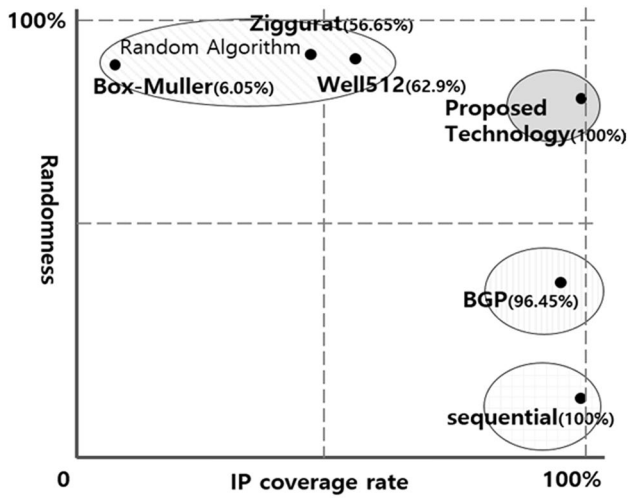

Fig. 4 Comparison among technologies for generating lists of IP addresses

the IP list of alive states. The handshake scan module scans information about major ports such as FTP, Telnet, SSH, and HTTP that are used for communication in the device. Information collected from the major ports generates information by going through the process of extracting scan information such as the system connection banner, encrypted communication information, packet header, and HTTP header/body information.

c. Third is the "OS fingerprinting" module for extracting the OS information. It uses the previous method of generating TCP/IP-based fingerprints and comparing them with the OS matching rule. A total of 77 OSs can be identified by comparing the OS matching rule with information from nine fields related to the TCP/IP packet, such as TTL, IPID, total_length, window_size, MSS, timestamps, sackOK, don't_fragment, nop, and window_scaled. In other scan tools, the name of the OS contained in the banner is parsed to identify the OS. If the name of the OS is not found in the device information, the device is not shown separately.

The collected information of the device connected to the Internet is generated as JSON files, which are periodically updated in the network file system (NFS) of the system for analyzing vulnerability information about devices. The operation time of the three modules for collecting device information may change depending on the addition of handshake scan modules. Currently, the new data set is updated on a weekly cycle.

4.3 Technology of analyzing vulnerability information about device

The analysis system analyzes vulnerability information about devices, which receives device information based

on the IP address from the collection system. As in Fig. 5 shown, it is composed of two modules for analyzing device information and finding vulnerabilities.

a. First is the "device data analysis" module, which extracts information necessary for analyzing vulnerability from the device information received from the collection system. Device data include all kinds of identifying information such as banner, packet header, and HTML. Such identification information can be extracted and classified by keyword units to identify information such as product, OS, and application. The extracted information is matched with a CPE, which consists of the names of IT products and platforms in the standard format, and then tagged with a CPE. An analysis is also conducted on the basic information about the device, such as the network and location information included in the IP information (Ussath et al. 2016).
b. Second is the "vulnerability analysis" module, which collects public vulnerability information and uses it to analyze vulnerability information about the device. This module mainly serves to collect and classify public vulnerability information such as common vulnerabilities and exposures (CVE) and to carry out correlation mapping between device information and vulnerability information based on CPE (Genge and Enăchescu 2015). Currently, the collected vulnerability information includes CVE, CPE, common weakness enumeration (CWE) and common vulnerability scoring system (CVSS) information provided by National Vulnerability Database (NVD), as well as Microsoft (MS) security patch information. This information includes CPE related to the vulnerability information. In this module, CPE information included in the vulnerability information is mapped with CPE tag in the device information from the collection system. Through the mapping process, it is possible to obtain the information in the CPE–CVE list contained in the device.

Information generated through the collection system and the analysis system as explained above is created in JSON format and is updated in the NFS of the sharing system.

4.4 Technology for sharing information on security threats

Information generated by the two systems above is processed in STIX standard format through the sharing system, as shown in Fig. 6. Then, the information is shared with external users or organizations via web interface, search API, TAXII protocol, etc. Through the sharing system, information included in the device as well as known vulnerabilities can be searched for. In addition, information about the change of device information can be produced by conducting

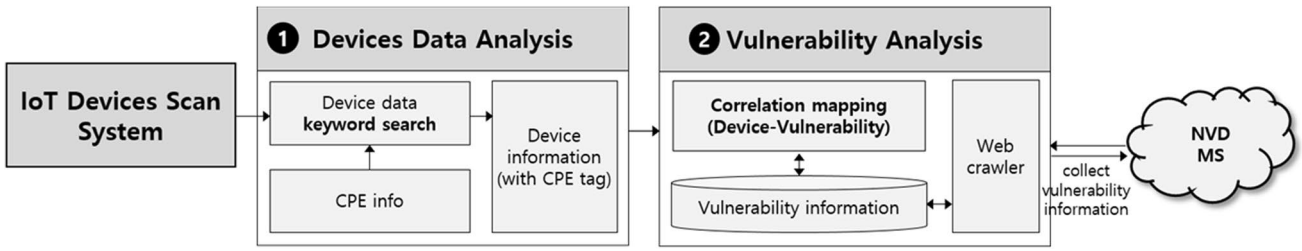

Fig. 5 Technology for analyzing vulnerability information about devices

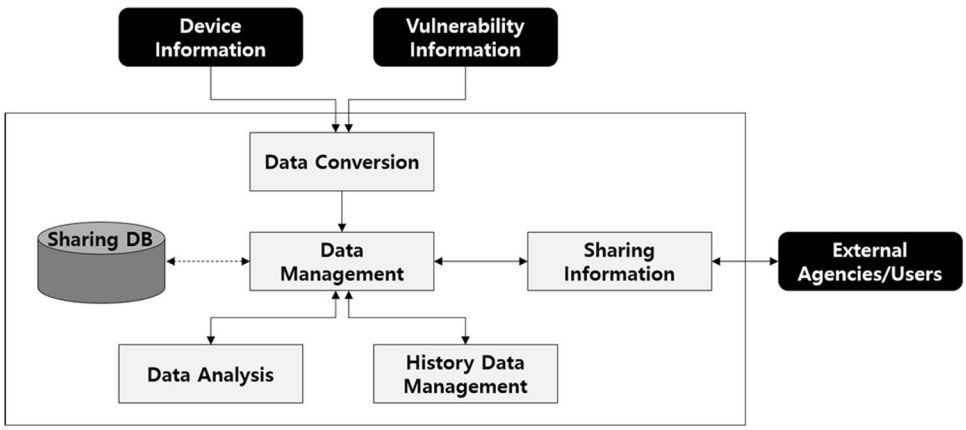

Fig. 6 Concept of vulnerability information sharing system

correlation analysis on the history information. For the security manager, it is important to acquire vulnerability information about the device.

To enable information search, the sharing system converts information about device vulnerability into the STIX format, stores and manages. In addition, based on the update cycle of information, previous data is backed up in the cloud server to store the history. Statistical information such as IoT devices with a large number of vulnerabilities and a history of changes in device information are generated by this system.

5 Analyzing the performance of the proposed technology

5.1 Performance of collecting device information

To scan the entire 4.3 billion IPv4 address and collect information about devices, high-speed traffic processing is required. This technology focus on the public IPv4 address, so that the number of subject IP is 3.702 billion which is excluding private and reserved IPv4 address spaces.

The server was set up in the cloud environment to operate the collection system and measure the performance of processing traffic (Keegan et al. 2016). Cloud server was used to deploy this collection system. The server was supported on the Linux OS and has 16 GB memory and an SSD 200G. It also uses a 10G network card to support the scanning process.

In the virtual server in the Amazon cloud mentioned above, IP alive scan was performed on 18 ports (including HTTP and FTP) for the entire IP. Then, the scan speed was measured for each execution time and traffic volume. The scan speed was expressed in throughput packets per minute (TPM) and compared with the speed described in a research paper from ZMap.

Table 3 Public performance of ZMap (IP Alive Scan)

Protocol	TPM	Alive scan	
		Duration	Number of IP
HTTP	54,752,940	1:08:05	3.702 billion
FTP	54,339,480	1:08:13	
CCTV	55,077,240	1:07:28	
SSH	54,151,380	1:08:42	
TELNET	54,594,240	1:07:49	
SMTP	54,077,040	1:08:38	
POP3	53,802,960	1:08:51	
IMAP	53,470,380	1:09:17	
HTTPS	54,186,000	1:08:16	
POP3S	54,304,080	1:08:18	
IMAPS	54,027,060	1:08:37	

Table 4 Public performance of ZMap (IP Alive scan)

Type of scan	Coverage	Time for scan
ZMap 1probe	0.987	1:09:45
ZMap 2 probes	1.000	2:12:35
NMap 1 probe	0.814	62.5 days
NMap 2 probes	0.978	116.3 days

As shown in Table 3, the duration of alive scan took longer than one hour for each protocol. On average, it took 1 h 8 min 23 s.

The collection system scans the ports of basic services (Connolly et al. 2014). In the table above, 11 protocol could be found and some of protocols such as HTTP require more than one port scan. In the future, the target ports of collection system will be expanded to collect more information.

Scanning of device information was performed in a common commercial network environment. A performance of about 55 million TPM was measured for an average of 1 h and 9 min for 3.702 billion accessible IP addresses. This can be interpreted as a performance similar to the "one-probe" condition announced in ZMap as in Table 4 shown (Apoorva et al. 2017; Bodenheim et al. 2014; Vijayarajan et al. 2016).

5.2 Performance of analyzing the vulnerability information about device

To analyze the vulnerability information about devices, it is necessary to extract a large amount of CPE information from the scan information of the devices. Since the CVE information is mapped with the extracted CPE information, the analysis rate of vulnerability information about the devices becomes higher as more CPE information is extracted.

The result of the analysis above shows how much CPE information was extracted to analyze the vulnerability information in each port from the information of 10,000 sampling devices and how many matched with CVE information. On average, 83.9% of CPE information could be found in the port information of the sampled devices, and this makes it possible to analyze vulnerability information.

This analysis rate means the extraction rate of CPE information for the each scan data of the protocol.

$$Analysis\ rate\ (\%) = \frac{the\ number\ of\ extracted\ CPE\ data}{the\ number\ of\ sampling\ devices}. \quad (1)$$

Table 5 Analysis rate of vulnerability (10,000 sampling devices)

Protocol	FTP	SMTP	SSH	IMAP	IMAPS	POP3	CWMP	HTTP	TELNET
Analysis rate (%)	95.8	72.9	78.4	96.7	97.1	94	50.3	87.9	82.4

The extracting CPE information process is based on a banner grab, so that the difference in the analysis rate of vulnerability is caused. According to this analysis, the extracted CPE information is the largest in the IMAPS scan information and the smallest in the CWMP scan information, as shown in Table 5.

6 Conclusion

Recently, the rapid spread of IoT services has caused an increase in the number of devices connected to the Internet. However, it is not easy to directly manage the security of IoT devices (including routers and CCTVs) due to the characteristics of their usage. Under this environment, security management such as periodic vulnerability inspections and security patches for each device is insufficient. This could cause these devices to become the targets of attackers who exploit such vulnerabilities (Arora et al. 2006).

In this paper, a platform is proposed to prevent cyber-attacks using the information of vulnerable devices connected to the Internet. Recent attacks on the IoT environment are increasing, especially caused by miss configuration or lack of management, so that it is important to promptly respond. Previous engines that search for device information, support only keyword searches after collecting device information. However, the proposed technology provides public vulnerability information and patch information as well, it makes quick response through checking vulnerability of devices. This proposed platform could be effective in prevention when it shares the threat information with network service providers or government.

In the future, additional research will be conducted to improve the traffic processing algorithm for collecting device information at high speed and to expand the scope of identifiable vulnerabilities by also collecting unstructured information about security vulnerabilities.

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

References

Apoorva M, Eswarawaka R, Reddy PVB (2017) A latest comprehensive study on structured threat information expression (STIX) and trusted automated exchange of indicator information (TAXII). In: Proceedings of the 5th international conference on frontiers in intelligent computing: theory and applications, pp 477–482. doi:10.1007/978-981-10-3156-4_49

Arora A, Nandkumar A, Telang R (2006) Does information security attack frequency increase with vulnerability disclosure? An empirical analysis. Inf Syst Front 8(5):350–362. doi:10.1007/s10796-006-9012-5

Barnum S (2012) Standardizing cyber threat intelligence information with the structured threat information expression (STIX™). MITRE Corporation. http://stixproject.github.io/getting-started/whitepaper. Accessed 3 Mar 2017

Becsi T, Aradi S, Gaspar P (2015) Security issues and vulnerabilities in connected car systems. In: 2015 international conference on models and technologies for intelligent transportation systems, pp 477–482. doi:10.1109/mtits.2015.7223297

Berhanu Y, Abie H, Hamdi M (2013) A testbed for adaptive security for IoT in eHealth. In: Proceedings of the international workshop on adaptive security. doi:10.1145/2523501.2523506

Bodenheim R, Butts J, Dunlap S, Mullins B (2014) Evaluation of the ability of the Shodan search engine to identify Internet-facing industrial control devices. Int J Crit Infrastruct Prot 7(2):114–123. doi:10.1016/j.ijcip.2014.03.001

Brown S, Gommers J, Serrano O (2015) From cyber security information sharing to threat management. In: Proceedings of the 2nd ACM workshop on information sharing and collaborative security, pp 43–49. doi:10.1145/2808128.2808133

Cisco (2016) Cisco 2016 midyear cybersecyrity report of cisco. http://www.cisco.com/c/dam/m/en_ca/never-better/assets/files/midyear-security-report-2016.pdf. Accessed 3 Mar 2017

Connolly J, Davidson M, Schmidt C (2014) The trusted automated exchange of indicator information (TAXII). The MITRE Corporation. http://taxiiproject.github.io/getting-started/whitepaper. Accessed 3 Mar 2017

Durumeric Z, Adrian D, Mirian A, Bailey M, Halderman JA (2015) A search engine backed by internet-wide scanning. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, pp 542–553. doi:10.1145/2810103.2813703

Genge B, Enăchescu C (2015) ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services. Secur Commun Netw 9(15):2696–2714. doi:10.1002/sec.1262

Genge B, Haller P, Enăchescu C (2015) Beyond internet scanning: banner processing for passive software vulnerability assessment. Int J Inf Secur Sci 4(3):81–91. http://www.semanticscholar.org/paper/Beyond-Internet-Scanning-Banner-Processing-for-Pas-Genge-Haller/07b987672346b9583af0d994a658a9dd938ecacb. Accessed 3 Mar 2017

Huh JH, Seo K (2016) Design and test bed experiments of server operation system using virtualization technology. Hum Centric Comput Inf Sci. doi:10.1186/s13673-016-0060-7

Ionita MG, Patriciu VV (2016) Secure threat information exchange across the internet of things for cyber defense in a fog computing environment. Inf Econ 20(3/2016):16–27. doi:10.12948/issn14531305/20.3.2016.02

Joo JW, Lee JK, Park JH (2015) Security considerations for a connected car. J Converg 6(2):1–9. http://www.manuscriptlink.com/journals/joc/digitalLibrary/2015/6/2/2892

Keegan N, Ji SY, Chaudhary A, Concolato C, Yu B, Jeong DH (2016) A survey of cloud-based network intrusion detection analysis. Hum Centric Comput Inf Sci. doi:10.1186/s13673-016-0076-z

Lee S, Shin SH, Roh B (2017) Abnormal behavior-based detection of Shodan and Censys-like scanning. In: 2017 ninth international conference on ubiquitous and future networks, pp 1048–1052. doi:10.1109/icufn.2017.7993960

Maity S, Park JH (2016) Powering IoT devices: a novel design and analysis technique. J Converg 7:1–18. http://www.manuscriptlink.com/journals/joc/digitalLibrary/2016/7/0/3571. Accessed 3 Mar 2017

Middleton P, Kjeldsen P, Tully J (2013) Forecast: the internet of things, Worldwide, 2013. Report of Gartner. http://www.gartner.com/doc/2625419/

Oehmen C, Peterson E, Cox BA (2015) Behavior models to express and share threat information. IT Prof 17(5):12–14. doi:10.1109/mitp.2015.93

Ring T (2014) Threat intelligence: why people don't share. Comput Fraud Secur 2014(3):5–9. 10.1016/S1361-3723(14)70469-5

Serrano O, Dandurand L, Brown S (2014) On the design of a cyber security data sharing system. In: Proceedings of the 2014 ACM workshop on information sharing and collaborative security, pp 61–69. doi:10.1145/2663876.2663882

Stock B, Pellegrino G, Rossow C, Johns M, Backes M (2016) POSTER: mapping the landscape of large-scale vulnerability notifications. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp 1787–1789. doi:10.1145/2976749.2989057

Ussath M, Jaeger D, Cheng F, Meinel C (2016) Pushing the limits of cyber threat intelligence: extending STIX to support complex patterns. Information Technology, New Generations, pp 213–225. doi:10.1007/978-3-319-32467-8_20

Vijayarajan V, Dinakaran M, Tejaswin P, Lohani M (2016) A generic framework for ontology-based information retrieval and image retrieval in web data. Hum Centric Comput Inf Sci. doi:10.1186/s13673-016-0074-1