System Hardening and Security Monitoring for IoT Devices to Mitigate IoT Security Vulnerabilities and Threats

Seul-Ki Choi[1], Chung-Huang Yang[2], and Jin Kwak[3]
[1] ISAA Lab., Department of Computer Engineering, Ajou University, Suwon, Republic of Korea
[e-mail: skchoi.isaa@gmail.com]
[2] National Kaohsiung Normal University, Kaohsiung, Taiwan
[e-mail: chyang@nknu.edu.tw]
[3] Department of Cyber Security, Ajou University, Suwon, Republic of Korea
[e-mail: security@ajou.ac.kr]
*Corresponding author: Jin Kwak

Received September 30, 2017; revised January 5, 2018; accepted January 22, 2018; published February 28, 2018

Abstract

The advent of the Internet of Things (IoT) technology, which brings many benefits to our lives, has resulted in numerous IoT devices in many parts of our living environment. However, to adapt to the rapid changes in the IoT market, numerous IoT devices were widely deployed without implementing security by design at the time of development. As a result, malicious attackers have targeted IoT devices, and IoT devices lacking security features have been compromised by attackers, resulting in many security incidents. In particular, an attacker can take control of an IoT device, such as Mirai Botnet, that has insufficient security features. The IoT device can be used to paralyze numerous websites by performing a DDoS attack against a DNS service provider. Therefore, this study proposes a scheme to minimize security vulnerabilities and threats in IoT devices to improve the security of the IoT service environment.

Keywords: IoT security, system hardening, security monitoring, security threat

---

A preliminary version of this paper was presented at APIC-IST 2017, and was selected as an outstanding paper.

http://doi.org/10.3837/tiis.2018.02.022 **ISSN : 1976-7277**

1. Introduction

With the recent rapid development of hardware technology and network technology, the Internet of Things (IoT) technology is being applied to every part of our lives [1][2]. The areas where this IoT technology is applied include a home environment called Smart Home and wider spaces such as a Smart Factory and a Smart City [3][4]. Furthermore, in the transportation sector, intelligent sensors are installed in vehicles and roads, and are all connected through networks, which are used for various services. IoT technology is being used as a basic technology for the Cooperative-Intelligent Transport System (C-ITS), which is called a next-generation intelligent transportation system [5]. The IoT technology is applied to various fields, and many services utilizing IoT technology are emerging.

However, IoT devices have been widely deployed with weak security features or a lack of security [6] owing to efforts to quickly provide IoT services according to the rapid growth and diverse needs of the market. These features have made IoT devices a good target for attackers with malicious intentions, and in many cases, exploits using IoT devices have been occurring.

One of the most typical accident cases was the Mirai Botnet incident in October 2016 [7]. Numerous IoT devices were compromised by attackers, and through an IoT device, the attackers performed a DDoS attack against Dyn, a DNS provider. This attack caused delays of access or network paralysis on many websites such as Twitter and Netflix. Furthermore, several IoT devices were infected with malicious code as the source code of the malicious code used in the accident was released online.

Furthermore, IoT devices that remotely control the opening and closing of windows in the Smart Home and environment could be threatened because attackers use the compromised devices to unlawfully break into houses. For example, the control authority of IP CCTV or IP Webcam that operates for personal security, if compromised by attackers, can lead to threats such as privacy leakage.

IoT devices that are widely deployed and used throughout our lives can be physically dangerous if captured by malicious attackers. The compromised IoT devices can be further exploited as tools for performing secondary attacks.

However, IoT devices, which can be a significant problem if dominated by attackers, have been widely deployed with insufficient security or a lack of security, without considering security by design at the time of development to adapt to rapid changes in the market. Owing to this, IoT devices have been easily vulnerable to attacks.

Therefore, this study proposes a scheme that utilizes system hardening and security monitoring technology to minimize security vulnerabilities and threats by deploying basic security features on IoT devices that do not implement security by design.

We also implemented a prototype of the service that can easily check the activity of malware existing in an IoT device, as well as accessing the IoT device, in this study.

The service implemented in this study can contribute to the overall security of IoT devices through system hardening and security monitoring of IoT devices.

2. Related Work

2.1 Vulnerability and Security Threat Analysis for IoT Devices

This section analyzes the types of vulnerabilities in IoT devices defined by the Open Web Application Security Project (OWASP) and representative examples of security threats that are easily found in IoT devices [8].

2.1.1 IoT Vulnerability Project

The table below shows the IoT vulnerabilities defined by the OWASP [8].

Table 1. IoT Vulnerabilities

Vulnerability	Attack Surface
Username Enumeration	-Administrative Interface -Device Web Interface -Cloud Interface -Mobile Application
Weak Passwords	-Administrative Interface -Device Web Interface -Cloud Interface -Mobile Application
Account Lockout	-Administrative Interface -Device Web Interface -Cloud Interface -Mobile Application
Unencrypted Services	-Device Network Services
Two-Factor Authentication	-Administrative Interface -Cloud Web Interface -Mobile Application
Poorly Implemented Encryption	-Device Network Services
Update Sent Without Encryption	-Update Mechanism
Update Location Writable	-Update Mechanism
Denial of Service	-Device Network Services
Removal of Storage Media	-Device Physical Interfaces
No Manual Update Mechanism	-Update Mechanism
Missing Update Mechanism	-Update Mechanism
Firmware Version Display and/or Last Update Date	-Device Firmware
Firmware and Storage Extraction	-JTAG/SWD interface -In-Situ dumping -Intercepting a OTA update -Downloading from the manufacturer’s webpage -eMMC tapping -Unsoldering the SPI Flash/eMMC chip and reading it in an adapter
Manipulating the Code Execution Flow of the Device	-JTAG/SWD interface -Side channel attacks such as glitching
Obtaining Console Access	-Serial interfaces (SPI/UART)
Insecure 3rd-Party Components	-Software

2.1.2 Example of Security Threat to IoT Devices

• Lack of authentication

IoT services operated on the web can experience security threats in which an unauthorized third party can easily access corresponding IoT services and devices because the authentication function for users is weak or absent. In particular, services such as SHODAN, which can retrieve information about devices connected to the Internet, provide a large amount of information on network devices such as Network Attached Storage (NAS), routers, and IoT devices such as IP CCTV in addition to servers [9].

Some IoT devices provide services (HTTP, SSH, FTP, Telnet, etc.) that collect information about IoT devices connected to the Internet and operate in the IoT device, and lack an authentication function. Thus, these IoT devices can be vulnerable to security threats such as unauthorized access by a third party to the IoT services and devices.

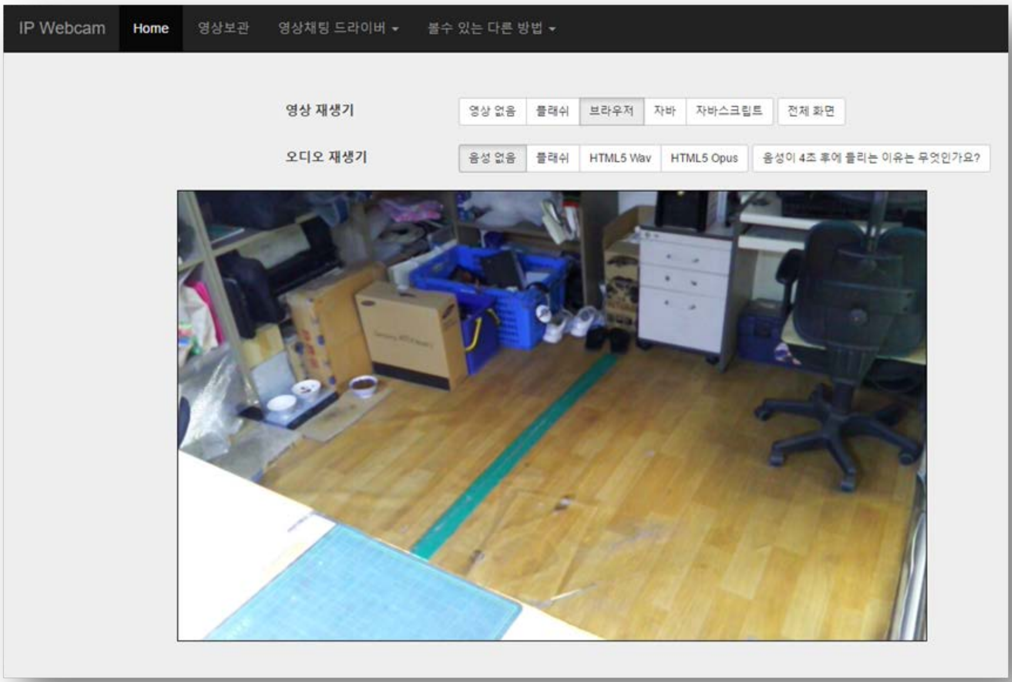


Fig. 1. Example of lack of authentication

• Console access

IoT device manufacturers may leave a physical access path for troubleshooting their products. However, if a physical access path that was left for normal purposes such as troubleshooting is identified by an attacker, the access path can be exploited as an attack path.

Fig. 2 shows that the path for UART communication was discovered by partially disassembling an IoT device. Such an access path allows for direct access to the operating system of the IoT device, which is vulnerable to various attacks such as tampering with firmware or inserting malicious code.

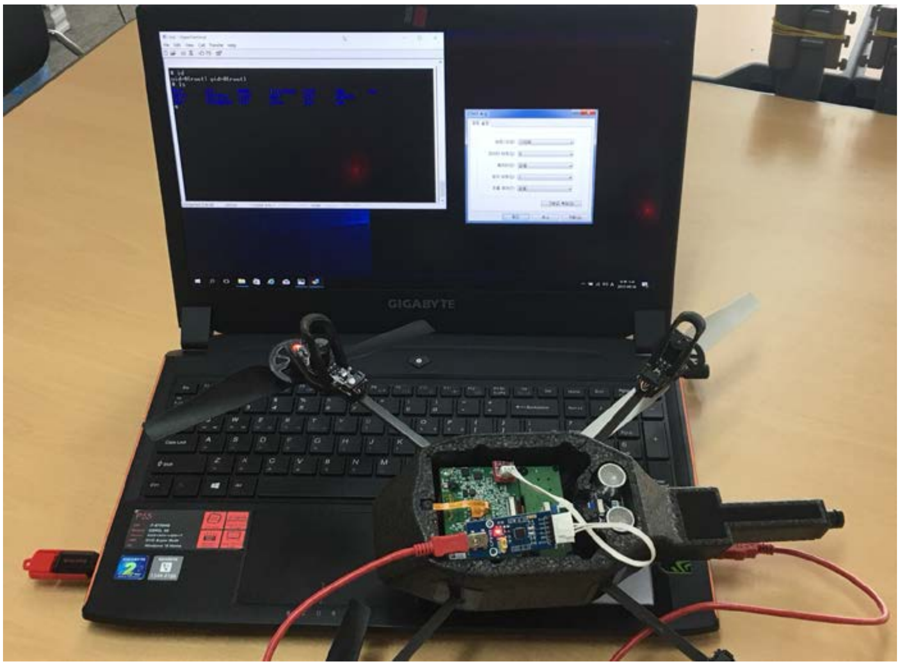
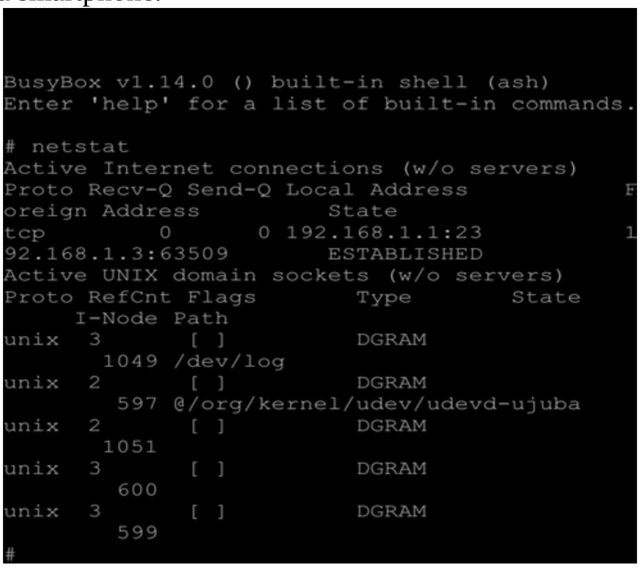


Fig. 2. Example of console access

• Internal access via vulnerable service (Telnet)

Unauthorized access by third parties may be possible owing to the opening of unnecessary or vulnerable service ports in addition to access ports for providing IoT services. In particular, IoT devices that use the Telnet service for direct access to an embedded operating system are on the market. Because the Telnet service does not support cryptographic communication, it is possible for an attacker to easily retrieve not only the result values of commands and the commands themselves through sniffing of the communication packet but also the ID and password used during the Telnet login.

Fig. 3 shows the successful result of unauthorized access to the Telnet service operated by an IoT device using a smartphone.



```
BusyBox v1.14.0 () built-in shell (ash)
Enter 'help' for a list of built-in commands.

# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           F
oreign Address         State
tcp        0      0 192.168.1.1:23          1
92.168.1.3:63509       ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State
     I-Node Path
unix  3      [ ]         DGRAM
       1049 /dev/log
unix  2      [ ]         DGRAM
        597 @/org/kernel/udev/udevd-ujuba
unix  2      [ ]         DGRAM
       1051
unix  3      [ ]         DGRAM
        600
unix  3      [ ]         DGRAM
        599
#
```

Fig. 3. Example of Internal access via Telnet

2.1.3 IoT Vulnerability Report

Fig. 4 shows the number of vulnerabilities reported from 2012 to 2016 by category [10].

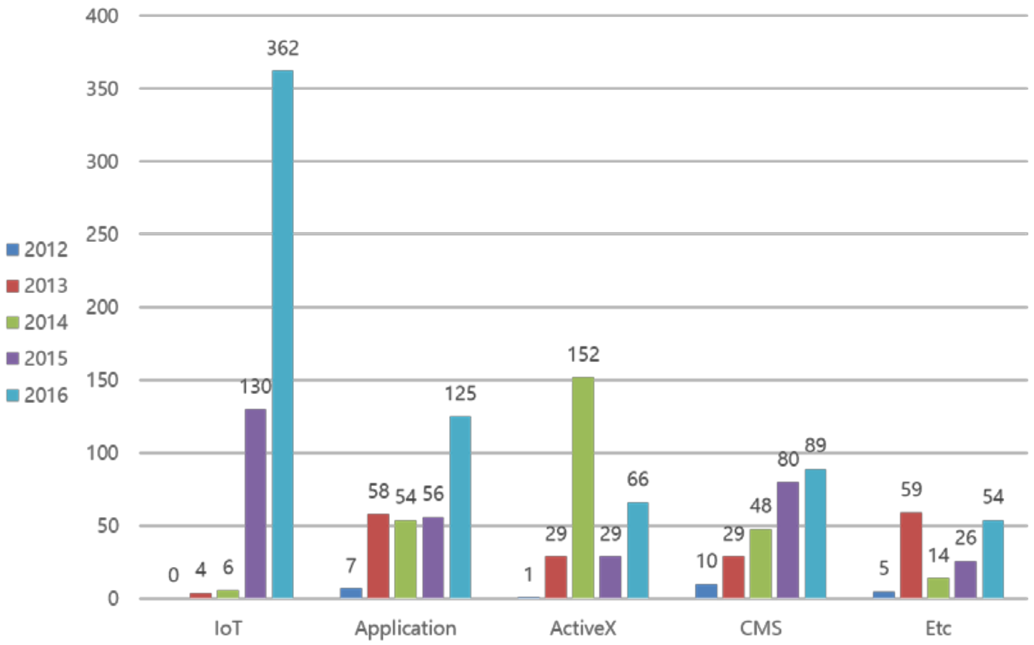


Fig. 4. Trend of vulnerability report

Fig. 4 shows that the number of vulnerability reports on the IoT has increased sharply since 2015. Furthermore, **Fig. 5** shows the types of IoT devices that were identified in a total of 502 IoT vulnerability reports [10].

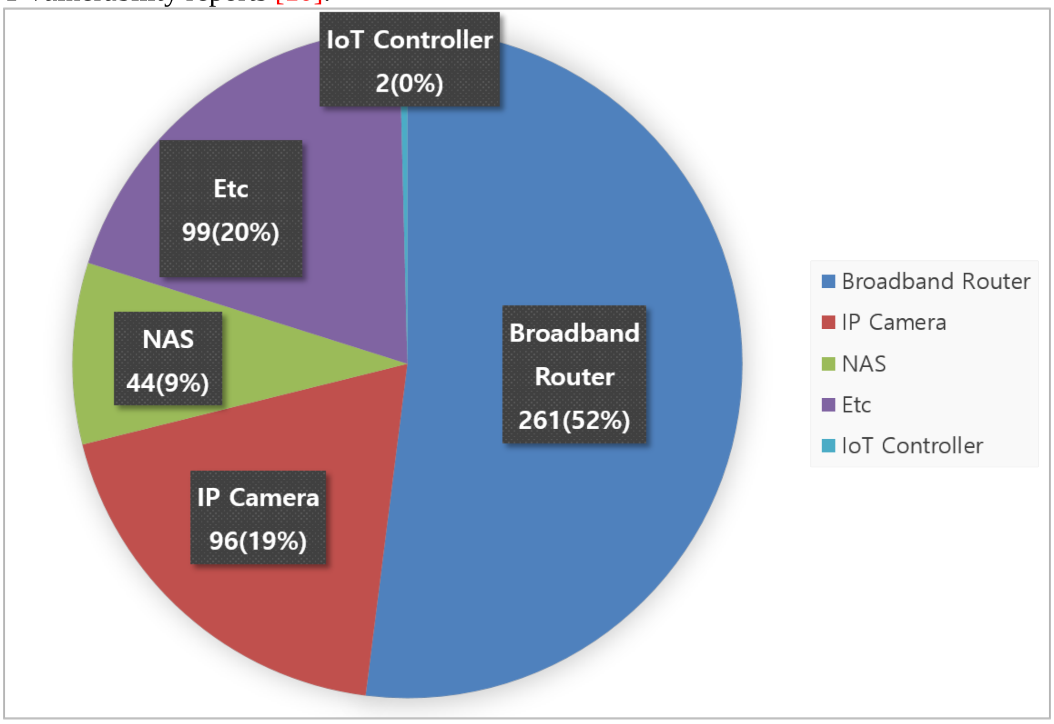


Fig. 5. Types of IoT vulnerability reports

A total of 80% of IoT devices such as broadband routers, IP cameras, and NASs are vulnerable to security threats because IoT devices have a performance level that can be operated with an operating system rather than acting as a simple sensor.

2.2 Linux System-Hardening Checklist

System hardening refers to a technique that minimizes security vulnerabilities and threats by setting various functions in the target system [11]. The systems are mainly connected to the network, and are primarily used as a method for protecting servers with frequent external access. **Table 2** is a checklist for hardening to minimize security vulnerabilities and threats based on Red Hat Enterprise Linux 7 [12].

Table 2. System Hardening Checklist

Category	Hardening Checklist
Preparation and Physical Security	-If machine is a new install, protect it from hostile network traffic until the operating system is installed and hardened. -Set a BIOS/firmware password. -Configure the device boot order to prevent unauthorized booting from alternate media. -Use the latest version of RHEL possible.
Filesystem Configuration	-Create a separate partition with the nodev, nosuid, and noexec options set for /tmp. -Create separate partitions for /var, /var/log, /var/log/audit, and /home. -Bind mount /var/tmp to /tmp. -Set nodev option to /home. -Set nodev, nosuid, and noexec options on /dev/shm. -Set sticky bit on all world-writable directories.
System Updates	-Register with Red Hat Satellite Server so that the system can receive patch updates. -Install the Red Hat GPG key and enable gpgcheck.
Secure Boot Settings	-Set user/group owner to root, and permissions to read and write for root only, on /boot/grub2/grub.cfg. -Set boot loader password. -Remove the X Window system. -Disable X Font Server.
Process Hardening	-Restrict core dumps. -Enable Randomized Virtual Memory Region Placement.
OS Hardening	-Remove legacy services. -Disable any services and applications started by xinetd or inetd that are not being utilized. -Remove xinetd, if possible. -Disable legacy services. -Disable or remove server services that are not going to be utilized. -Set Daemon umask.
Network Security and Firewall Configuration	-Limit connections to services running on the host to authorized users of the service via firewalls and other access control technologies. -Disable IP forwarding. -Disable send packet redirects. -Disable source routed packet acceptance. -Disable ICMP redirect acceptance.

	-Enable Ignore Broadcast Requests. -Enable Bad Error Message Protection. -Enable TCP/SYN cookies.
Remote Administration via SSH	-Disable SSH Root login. -Set SSH PermitEmptyPasswords to No.
System Integrity and Intrusion Detection	-Install and configure AIDE. -Configure SELinux. -Install and configure OSSec HIDS.
Logging	-Configure Network Time Protocol (NTP). -Enable system accounting (auditd). -Install and configure rsyslog. -All administrator or root access must be logged. -Configure log shipping to separate device/service.
Files/Directory Permission/Access	-Integrity checking of system accounts, group memberships, and their associated privileges should be enabled and tested.
PAM Configuration	-Ensure that the configuration files for PAM, /etc/pam.d/* are secure. -Upgrade password hashing algorithm to SHA-512. -Set password creation requirements. -Restrict root login to system console.
Warning Banners	-If network or physical access services are running, ensure the university warning banner is displayed. -If the system allows logins via a graphical user interface, ensure the university warning banner is displayed prior to login.
Anti-Virus Considerations	-Install and enable anti-virus software. -Configure to update signature daily on AV.

3. Proposed System

3.1 Motivation and Purpose

This study analyzed the causes of IoT security incidents as follows:
• No security function for quick response to IoT service market
• Easy acquisition of information on IoT devices connected to the Internet

This study aimed to provide security technology that can be easily applied to IoT devices in use without the implementation of security by design at the time of design and sales.

Furthermore, IoT devices other than designated applications do not interact directly with users unless abnormal cases occur, and IoT devices do not operate except with regard to applications specified in the system [9]. In consideration of such characteristics of IoT devices, this study used technologies, namely system hardening and security monitoring to prevent or detect activities other than normal operation.

3.2 Mountable Device Type

IoT devices such as home gateways and auto thermostats can independently perform various functions or can collect and process information from subsensor data to provide IoT services, which are capable of collecting and processing data. IoT devices with such performance are connected to external networks and have become good targets for attackers, because these devices can be easily exploited owing to their performance conditions.

Thus, the system and security functions proposed in this study are applied to IoT devices that are easily selected as attack targets by attackers. For example, IoT devices are loaded with lightweight versions of Linux- and Unix-based operating systems. **Table 3** lists the representative types of devices for each of the seven IoT industry sectors, and the main applications of the proposed technique are shown in bold and are underlined [13].

Table 3. Seventh IoT Industry Field and Major Device Type

7th Industry Fields	Device Type (Processor Type)
Smart Home	Smart Plug (Atmega128 (8 bits)), Power Sensor (ARM9 (32 bits), Smart Light bulb (Atmega128 (8 bits)), Smart Electronics (8–32 bits), **<u>Home Gateway (Cortex A9 (32 bits))</u>**, **<u>Auto Thermostat (Cortex A8 (32 bits))</u>**
Medical	Wearable Healthcare Device (Cortex M (32 bits)), Clothing Healthcare (MSP430 (16 bits)), Smart sneakers (MSP430 (16 bits)), **<u>Smart Watch (Cortex A9 (32 bits)</u>**
Transportation	**<u>Automotive sensors and ECU Device (Cortex A9 (32 bits))</u>**, ARM7 (32 bits))
Environment/Disaster	Gas Sensor (Atmega128 (8 bits)), Image Processing Module (Atmega128 (8 bits))
Manufacturing	Factory-Things (Atmega128 (8 bits)), Control/Sensing Module (Atmega128 (8 bits)
Construction	Crack/Vibration Sensor Module (Atmega128 (8 bits))
Energy	Smart Meter (ARM 32 (32 bits)), PIC 32 (32 bits), Distribution Switch Control (32 bits), **<u>RTU (Cortex A9 (32 bits))</u>**

3.3 System Hardening and Security Monitoring of IoT Devices

3.3.1 System Hardening

IoT devices can be accessed by authorized users and by an unspecified number of users through external networks. Thus, it is necessary to ensure the security of IoT devices by minimizing security vulnerabilities and threats from various approaches that occur out of the scope of this study. Such deployment schemes should be applied to the performance of the target IoT devices, as shown in the Linux system-hardening checklist provided in Section 2.2 of this paper.

3.3.2 Security Monitoring

The primary function of security monitoring presented in this study is to continuously monitor the system hardening status of IoT devices. Furthermore, the security monitoring feature continuously monitors the logs generated from the logging function activated within the IoT devices to detect earlier anomalous signs, thus minimizing security vulnerabilities and threats. For example, the security monitoring feature can detect anomalous indications, such as persistent SSH access requests from unauthorized external IPs through continuous internal log analyses, in order to perform various response plans such as notifying the IoT device manager or blocking the corresponding IP.

As mentioned above, the security monitoring feature continuously monitors logs recorded in the system hardening status and inside IoT devices. Among the events occurring in IoT devices, events and logs that can be used to improve security are defined by the OWASP, as shown in **Table 4** [14].

Table 4. IoT Logging Events

Event Category	Events
Request Exceptions	-Attempt to Invoke Unsupported HTTP Method -Unexpected Quantity of Characters in Parameter -Unexpected Type of Characters in Parameter
Authentication Exceptions	-Multiple Failed Passwords -High Rate of Login Attempts -Additional POST Variable -Deviation from Normal GEO Location
Session Exceptions	-Modifying the Existing Cookie -Substituting Another User's Valid SessionID or Cookie -Source Location Changes During Session
Access Control Exceptions	-Modifying URL Argument Within a GET for Direct Object Access Attempt -Modifying Parameter Within a POST for Direct Object Access Attempt -Forced Browsing Attempt
Ecosystem Membership Exceptions	-Traffic Seen from Disenrolled System -Traffic Seen from Unenrolled System -Failed Attempt to Enroll in Ecosystem -Multiple Attempts to Enroll in Ecosystem
Device Access Events	-Device Case Tampering Detected -Device Logic Board Tampering Detected
Administrative Mode Events	-Device Entered Administrative Mode -Device Accessed Using Default Administrative Credentials
Input Exceptions	-Double Encoded Character -Unexpected Encoding Used
Command Injection Exceptions	-Blacklist Inspection for Common SQL Injection Values -Abnormal Quantity of Returned Records
Honey Trap Exceptions	-Honey Trap Resource Requested -Honey Trap Data Used
Reputation Exceptions	-Suspicious or Disallowed User Source Location

3.3.3 Overall Operation Process

Fig. 6 shows the overall operation process of the proposed system. After the IoT device is turned on, the device scans the system hardening status defined by the user or the administrator, as well as the target binary and data, to check whether tampering occurred. The device continuously monitors logs of various events (process activity, network activity, etc.) that occur continuously inside the IoT device. If an anomalous sign occurs during this process, the device follows the corresponding policy defined by the user and the administrator, by performing a notification function.

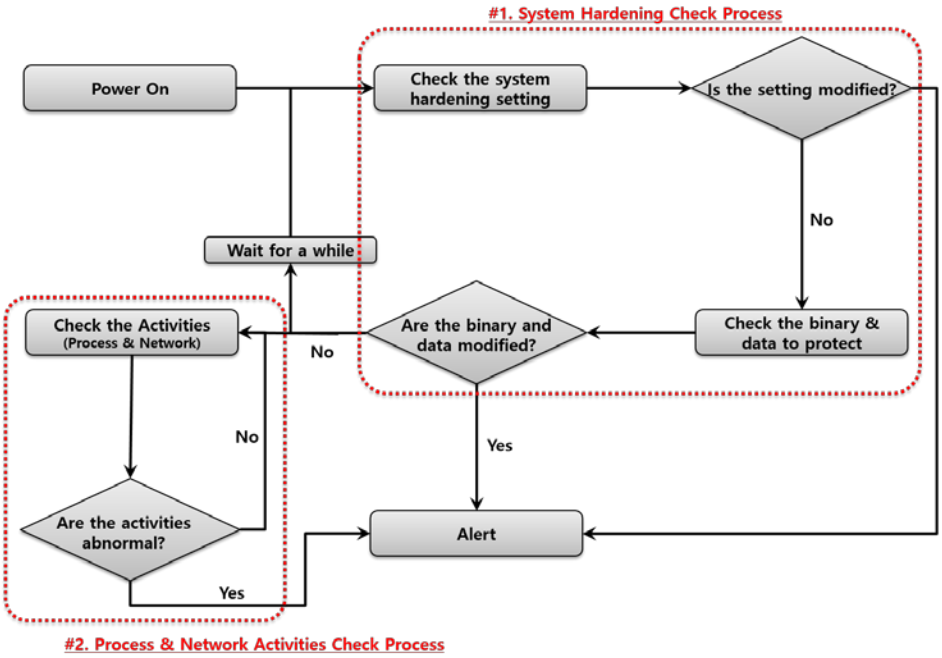


Fig. 6. Overall operation process

4. Experiment and Conclusion

This study developed a prototype by selecting some functions of the proposed system to verify whether the proposed system can operate on IoT devices. As shown in **Fig. 7**, the study further developed a web service that can easily monitor the status of several IoT devices. This study anticipates that the proposed techniques could be useful in managing numerous IoT devices such as in Smart Factory.

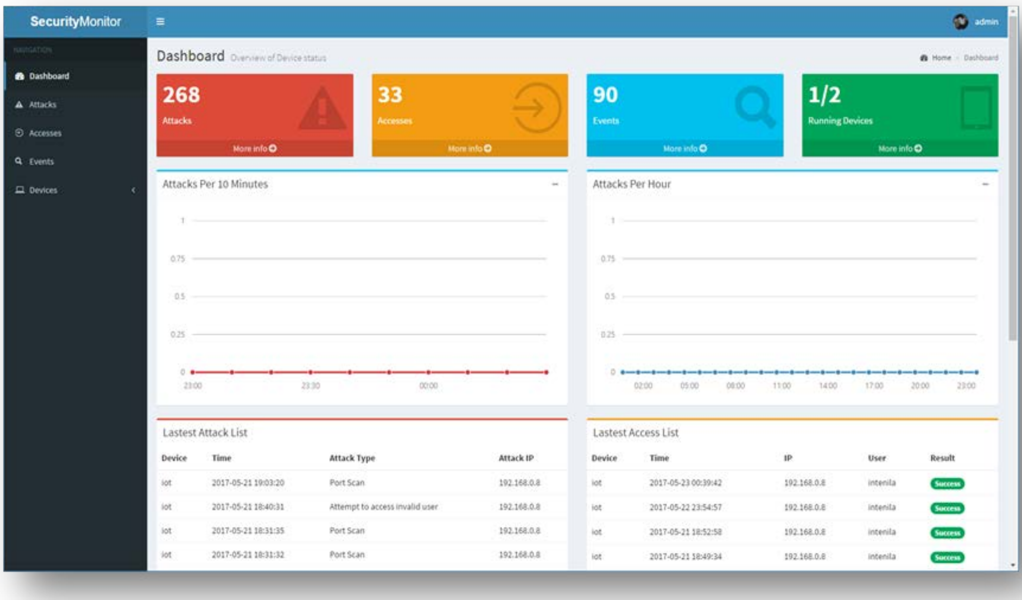


Fig. 7. IoT device management web service

The IoT market is changing rapidly, and IoT devices are being widely adopted in various environments. However, because IoT devices are distributed without implementing security by design from the development stage, existing IoT devices have been abused through various cyber threats. Therefore, this study proposed a system and its operation method that can easily apply security functions to IoT devices. The study further verified the usability of the proposed techniques by developing a prototype.

Future studies will investigate a scheme to use BusyBox in application to various processes, and develop strategies to lighten and optimize binaries that implement the above security technology, resulting in wider applications of the results in this study.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2017R1E1A1A01075110) and by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2017-2015-0-00403) supervised by the IITP(Institute for Information & communications Technology Promotion) and by the Ajou University research fund.

References

[1] Somia Sahraoui and Azeddine Bilami, "Asymmetric End-to-End Security for Human-to-Thing Communications in the Internet of Things," in *Proc. of IoT'16 Proceedings of the 6th International Conference on the Internet of Things*, pp.131-139, November 07-09, 2016. Article (CrossRef Link)

[2] Meesun Kim, Hyun Ahn and Kwanghoon Pio Kim, "Process-Aware Internet of Things: A Conceptual Extension of the Internet of Things Framework and Architecture," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 8, August 31, 2016. Article (CrossRef Link)

[3] Vu-Anh-Quang Nguyen, "Study on realtime control system in IoT based smart factory: Interference awareness, architectural elements, and its application," in *Proc. of Information Science and Technology (ICIST), 2017 Seventh International Conference on*, April 16-19, 2017. Article (CrossRef Link)

[4] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-khah and P. Siano, "Iot-based Smart Cities: a Survey," in *Proc. of Environment and Electrical Engineering (EEEIC), 2016 IEEE 16th International Conference on*, June 7-10, 2016. Article (CrossRef Link)

[5] Jorge Alfonso, Nuria Sánchez, José Manuel Menéndez and Emilio Cacheiro, "Cooperative ITS communications architecture: the FOTsis project approach and beyond," *IET Intelligent Transport System*, vol. 9, issue. 6, pp.591–598, August 06, 2015. Article (CrossRef Link)

[6] Elisa Bertino, Nayeem Islam, "Botnets and Internet of Things Security," *Computer*, vol. 50, issue. 2, pp. 76-79, 2017. Article (CrossRef Link)

[7] James A. Jerkins, "Motivating a market or regulatory solution to IoT insecurity with the Mirai botnet code," in *Proc. of Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual*, January 09-11, 2017. Article (CrossRef Link)

[8] OWASP, "IoT Vulnerabilities Project," Article (CrossRef Link)

[9] Ryan Williams, Emma McMahon, Sagar Samtani, Mark Patton and Hsinchun Chen, "Identifying Vulnerabilities of Consumer Internet of Things (IoT) Devices: A Scalable Approach," in *Proc. of Intelligence and Security Informatics (ISI), 2017 IEEE International Conference on*, July 2017 Article (CrossRef Link)

[10] Korea Internet & Security Agency, "SW New Vulnerability Reporting Award Status and Key Vulnerabilities," March 27, 2017. Article (CrossRef Link)

[11] National Institute of Standard and Technology, "Guide to General Server Security," *Special Publication*, 800-123, July 25, 2008. Article (CrossRef Link)

[12] Information Security Office, "Red Hat Enterprice Linux 7 Hardening Checklist," *The University of Texas at Austin*, Article (CrossRef Link)
[13] Korea Internet & Security Agency, "Guide to Using Cryptography Authentication Technology in Internet (IoT) Environment," April, 2016. Article (CrossRef Link)
[14] OWASP, "IoT Logging Events," Article (CrossRef Link)
[15] Thuy T.T. Nguyen and Grenville Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Communications Surveys and Tutorials*, vol. 10, issue 4, pp. 56-76, November, 2008. Article (CrossRef Link)

Seul-Ki Choi received Korea B.S. and M.S degrees in Department of Information Security Engineering from Soonchunhyang University. He is currently pursuing the Ph.D. degree in Department of Computer Engineering with Ajou University, Korea. His research interests include IoT Security, Vulnerability & Malware analysis and Cryptographic protocols.

Chung-Huang Yang received B.S. Degree from the National Cheng-Kung University at Taiwan in 1981 and M.S. and Ph.D. degrees from the University of Louisiana at Lafayette (formerly University of Southwestern Louisiana), USA in 1986 and 1990, respectively. He is currently a Professor in the Department of Software Engineering and Management at the National Kaohsiung Normal University, Taiwan. His research interests include Mobile Device Security, Digital Forensics, and Efficient Implementations of Cryptosystems.

Jin Kwak is a professor at Dept. Of Cyber Security in Ajou University, Korea. He received the Ph.D. degree from SKKU, Korea. His research interests include Cryptographic protocols, Applied security mechanisms for Cloud and Big Data system and so on.