IoT Vulnerability Analysis and Its Security Controls

K. Nakao[1]*, K. Yoshioka[1] and D. Inoue[2]

In our previous work, we analysed IoT devices and investigated the threat of vulnerable IoT devices compromised by malware by means of the proposed IoTPOT (honey) and IoTBOX (sandbox). In this paper, based on the previous work mentioned above, on vulnerable IoT devices which are owned by individuals without any management, we propose two types (approaches) of security controls for the less-controlled IoT devices. Recognising the current situation wheremany vulnerable IoT devicesare already infected by several types of malware, the first approach proposes a security solution to remove malware from infected IoT devices, or to stop the activation of malware (deletion of registry, exe, or scheduler). In the second approach, in order to develop general security controls which are commonly applicable, the development of a security function for updating software/firmware modules located in IoT devices is proposed. This security solution provides an initial secure software/firmware update procedure based on the secure software update procedure for ECUs (Electronic Control Units) in an ITS (Intelligent Transportation System) which has been developed in an ITU-T (International Standardization Body) as proposed by authors of this paper. Finally, a list of research topics for the IoT environment is provided forfuture collaborative research.

INTRODUCTION

In our previous paper [1], we IoT devices were analysed andthreats due to malwares of IoT devices were investigated. First analysing Telnet-based scans in darknet (using unused IP addresses), we recognised that the attacks (scans) on Telnet had dramatically increased since 2014. Moreover, by grabbing Telnet banners and web contents of the attackers, amajority of the attacks were indeed from IoT devices. Motivated by this, we proposed IoTPOT, a novel honeypot to emulate Telnet services of various IoT devices to analyse ongoing attacks in depth with backend high-interaction virtual environments called IoTBOX (sandbox analysis for IoT devices) for different CPU architectures. Over 39 days of experimental operation, we observed 76,605 download attempts of malware binaries from 16,934 visiting IPs. We also confirmed that none of these binaries could have been captured by existing honeypots that handled Telnet protocol such as honey and telnet password honeypot because they were not able to handle different incoming commands sent by the attackers.

Finally in our previous work, combining the observations results of IoTPOT and the sandbox analysis by IoTBOX, we confirmed that i) there were at least four distinct malware families spreading via Telnet, ii) their common behavior was performing DDoS and further propagation over Telnet, iii) some families evolved quickly, updating frequently and shipping binaries for a variety of CPU architectures, even in the limited observation period of 39 days.

In this paper, based on theprevious work on the vulnerable IoT devices which were owned by individuals without any management, we propose a method to implement security controls for the less-controlled IoT devices. The security controls should be provided fromthree different angles. 1) Security guidelines should be provided to improve IoT device owners' awareness such as promoting the use of appropriate IDs and Passwords, 2) Proper shipping of IoT devices by IoT vendors for the the initial setting of a more secure use of the Internet (e.g. close port 23) and 3) Removing malwares from infected IoT, or stopping the activation of malwares (deletion of registry, exe, or scheduler).

[1]Authors are with NICT (4-2-1 Nukui-Kitamachi, Koganei, Tokyo, Japan) and Yokohama National University (79-5 Tokiwadai, Hodogaya-ku Yokohama, Japan)
[2]Author is with NICT
*Corresponding author (K. Nakao, e-mail: ko-nakao@nict.go.jp)

It isimportant to consider a time-line of IoT devices, that is, a) most of the vulnerable IoT devices are already in the market and in use (already shipped), b) IoT devices are going to be shipped by IoT vendors, and c) future types of IoT devices of which we are still notable to foresee the characteristics. The above security control 1) could be applicable for all a)-c), however, security control 2) is for b) and security control 3) is basically for a).

Furthermore, in connection with the security control 3), the implementation of an appropriate software/firmware update function was significantly important for IoT devices in all time-lines (a)-c)). In this paper, consideration is given to this update function for IoT software/firmware based on the ITS secure update procedure. Finally, a list of research topics for the IoT environment is provided for future collaborative research.

METHODS

According to the current use-cases of IoT devices, we have recognised two major uses of IoT devices as follows:

Use-case-1: IoT devices are used and well controlled under the IoT based "services" such as lighting, parking, home-networking and so on. In this case, the owner of the IoT devices is the service provider and security controls should be considered by the provider;

Use-case-2: IoT devices are purchased by individuals for their own purposes such as health-care, home-network, security-monitoring, and so on. In this case, the device owner is the individual who has basic responsibility for the security.

In this paper, we basically focus on the **Use-case-2**. In addition to these use-cases above, we need to consider the time-line of IoT devices in use as follows:

Time-line-a: IoT devices that are already in the market and in use (already shipped) and there are many vulnerable devices observed based on our previous findings [1];

Time-line-b: IoT devices that are going to be shipped by IoT vendors and in this case, there islittle room to implement security controls before shipping;

Time-line-c: the new types of IoT devices which are expected to be available in about3 years. In this case, we will not be able to anticipate how to use the IoT devices.

Based on the above use-cases of IoT devices (Use-case-1 and 2) and the Time-lines (a-c), the following two practical research approaches can be identified in this paper:

Approach-1: By means of using our previous work on IoTPOT and IoTBOX, we firstly conduct the processes of "Monitoring IoT devices" and "Analysing IoT behaviors". These two processes are covered by the previous work [1]. The next process can be identified as the "Execution of IoT security controls" for vulnerable IoT devices and the last process is "Sharing knowledge of IoT intelligence" to be utilised for future security management. In this paper, we concentrate on the third process of the "Execution of IoT security controls" for IoT vulnerable devices for Use-case-2 and Time-line-a.

Approach-2: In order to develop general security controls which are applicable to all in common, one example could be the development of a security function for updating software/firmware modules located in the IoT devices. This approach could be applicable for Use-case-2 (even for Use-case-1) and for Time-line-b and c. In this paper, the security function of updating software/firmware for IoT devices is considered based on a similar function developed for the ITS (Intelligent Transportation System) environment.

RESULTS AND DISCUSSION

Approach-1

Recognising the current status wheremany vulnerable IoT devices are already infected by several types of malwares, methods to remove malwares from infected IoT devices, or to stop the activation of malwares (deletion of registry, exe, or scheduler) areconsiderable solutions in this approach.

More specifically, after identifying the“infected IoT device” by means of IoTPOT (Honey) and getting its IoT finger-print about the infected IoT, the IoT finger-print information can then be forwarded from IoTPOT (Honey) to IoT devices vendors or IoT integrated maintenance centers as shown in Figure 1.

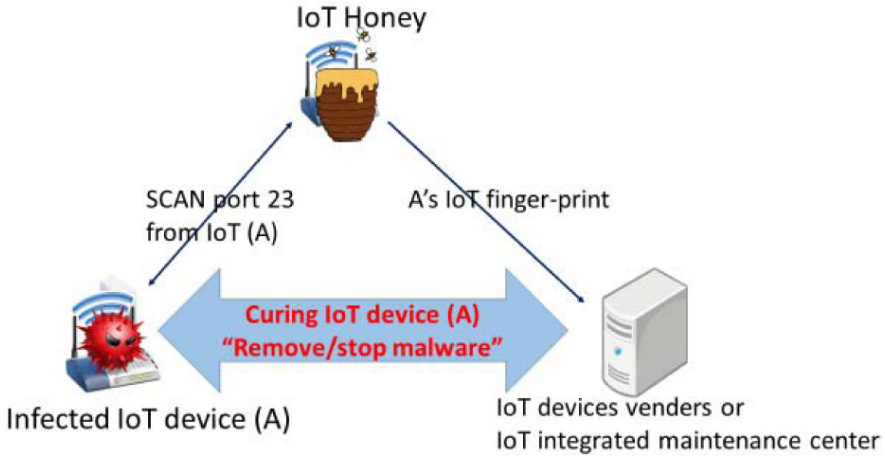

Figure 1. Scheme for curing IoT devices

In the scheme illustrated in Figure 1, we basically consider a new entity of IoT devices vendors or IoT integrated maintenance centers. Because of the “illegal access protection law” established in many countries, the IoTPOT never directly accesses the infected IoT devices withoutpermission obtained from the owner of the IoT devices. It is sometimes hard to obtain the permission of the IoT device owner who is an individual, because the device owner sometimes has very poor awareness of the IoT device and poor security knowledge. IoT devices vendors can cure the infected IoT devices for the purpose of maintenance of their own products (IoT devices). Furthermore, if the device vendors donot have the capability to cure the devices (e.g. because of the expense), then the IoT integrated maintenance centers can be another solution to totally cover many types of IoT devices for the purpose of curing the infected devices under contract with the IoT device vendors.

Considering the above scheme, we have started thetesting to cure infected devices fromremote in our own experimental environment equipped with several types of IoT devices that are the same products observed by IoTPOT. According to current experimental results, it was difficultto remove malwares in the infected IoT devices without having an agent software like anti-virus-software, but it was possible tostop activating the target malwares by deleting the registry, the exe. file, or its scheduler and so on. As it was not feasible for IoT devices vendors to deploy anti-virus-software in the IoT devices,therefore, curing the IoT devices from the vendors or IoT integrated maintenance centers was the feasible solution for this security control.

Approach-2

It is remarkable that the number of IoT devices have beendramatically increasing and thus, hundreds of security threats have been detected every day including vulnerability identification for general ICT environments. It isanticipated that the next market target of cyber-attacks (security threats) could be IoT environments. Considering the above circumstances, the function of secure remote updating software and/or firmware inside IoT devices should be a major consideration in IoT markets.

In the ITS (Intelligent Transportation System), the above secure remote updating function for ECUs (Electronic Control Unit) in the vehicle which were similar to IoT devices in terms of ITS environments have been understudy and are being standardised on an international level.

In the context of the remote updating function at the ITU-T (International Telecommunication Union – Technology), the following scopes of standards (Recommendation) have been identified [4]:

In the context of updates of software modules in the electric devices of vehicles in the intelligent transportation system (ITS) communication environment, this Recommendation aims to provide a procedure of secure software updating for ITS communication devices for the application layer in order to prevent threats such as tampering of and malicious intrusion to communication devices on vehicles. This includes a basic model of software update, its threat and risk analysis, security requirements and controls for software updates and a specification of abstract data format of the update software module.

The procedure is intended to be applied to communication devices on ITS vehicles under vehicle-to-infrastructure (V2I) communication by means of the Internet and/or ITS dedicated networks. The procedure can be practically utilised by car manufacturers and ITS-related industries as a set of standard secure procedures and security controls.

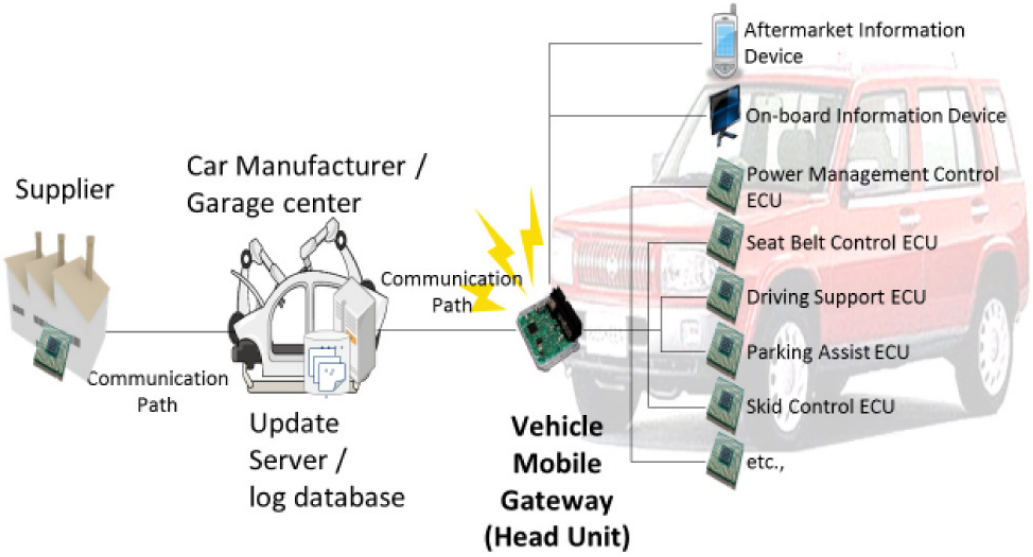

Figure 2. Basic components for secure software updates

In this paper, as an initial consideration, we tried to apply the secure procedure developed by ITU-T for ECUs in the ITS environment for the IoT software/firmware updating function. As shown in Figure 2, the basic components for the secure software update of ECUs in vehicles were “update server including log DB”, “Vehicle Mobile Gateway (VMG, called Head Unit)” and a series of ECUs. Update information stored in the update service was provided by the “Supplier of ECUs”.

In the case of software/firmware updates for IoT devices, “IoT integrated maintenance centers and IoT devices vendors” would have similar roles for the “Update service/Suppler of ECUs” in order to cure IoT devices in Approach-1. “IoT devices” would be the same target component as for “ECUs”. At this point, it was not clear whether the Gateway component for the IoT updating function was needed. Which was similar to “VMG” in the case of ITS.

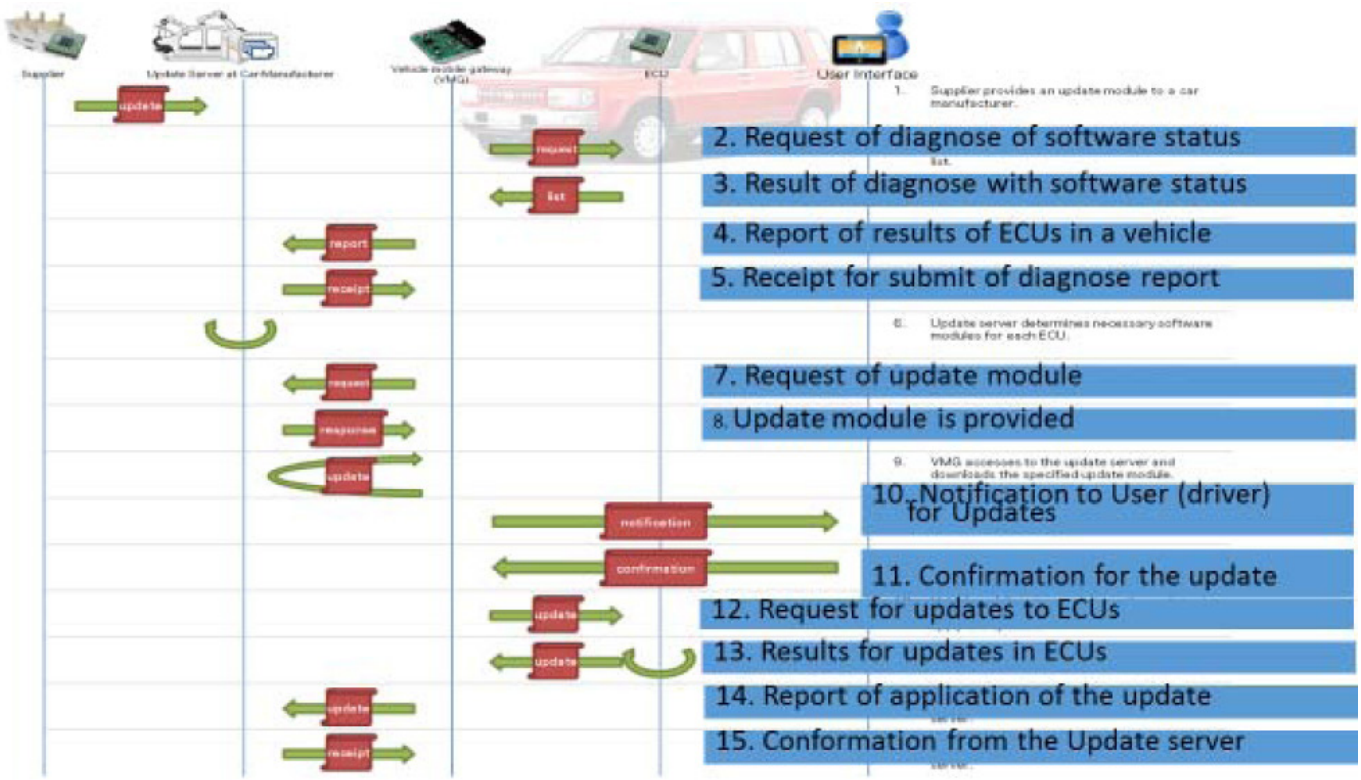

Figure 3. Software update procedure for ITS

Before considering the IoT software/firmware update function, the ITS software update procedure (function) needs to be learned in Figure 3 with the following descriptions of each step:

1. At the first step of the process, an update module is provided by an automotive component supplier, which occurs asynchronously with the following steps.
2. As the initiation of the update procedure starts, a vehicle mobile gateway (VMG) requests ECUs to submit their software list.
3. An ECU checks its software status, generates a list of software modules and reports it to the VMG.
4. The VMG submits the collected list to the update server to check whether any update for the vehicle exists.
5. The update server sends back a receipt of the submitted list to the VMG.
6. According to the list, the update server inspects the status of the installed software of the vehicle and determines the necessary software updates for the ECUs.
7. Since this inspection may take a long time, VMG periodically checks the necessity of the updates for the vehicle.
8. If there is any update, the update server sends an access uniform resource locators (URLs) for the updates; otherwise, it sends back only an acknowledgement message.
9. If there is any update for the vehicle, the VMG connects to the update server to download the update modules for the vehicle.
10. Before applying the updates to the ECUs, the VMG notifies the driver to confirm the application of the updates.
11. The driver confirms and accepts to apply the updates.
12. VMG delivers the update files to the corresponding ECUs and requests them to apply the updates (See 6.2.3).
13. Each ECU applies the update and reports the application result to the vehicle mobile gateway.
14. The vehicle mobile gateway submits a report of the application results to the update server.

15. Finally the update server sends back a
    - receipt of the update information. If the
    - application of the update has failed or
    - some remaining update is found, the update
    - server retries the procedure from step 6 to
    - 14 until the application has succeeded.

As the basic assumption for investigating the IoT software update procedure (function), the “IoT update handler” in IoT devices should be basically implemented by IoT devices vendors in order to provide update functionsfor the IoT devices. Based on the above software update procedure for the ITS environment, we propose the following software update procedure which can be simplified and adaptable for IoT devices as follows:

1. At the first step of the process, an update module should be provided by an IoT devices vendor, and be stored in the IoT integrated maintenance center/IoT devices vendors. The update occurs asynchronously with the following steps.
2. We can eliminate step-2 of the ITS for IoT update.
3. We can eliminate step-3 of the ITS for IoT update.
4. The IoT device submits a status information concerning the software/firmware implemented in the IoT device to the “IoT integrated maintenance center/IoT devices vendors”asking to check whether any update for the IoT device exists.
5. The “IoT integrated maintenance center/IoT devices vendors” sends back a receipt of the submitted status information to the IoT device.
6. According to the status information, the “IoT integrated maintenance center/IoT devices vendor” inspects the status of the installed software of the IoT device and determines the necessary software updates for the IoT device.
7. We can eliminatestep-7 of the ITS for the IoT update.
8. If there is any update, the “IoT integrated maintenance center/IoT devices vendor” sends an access uniform resource locators (URLs) for the updates; otherwise, it sends back only an acknowledgement message.
9. If there is any update for the IoT device, the IoT device connects to the “IoT integrated maintenance center/ IoT devices vendor” to download the update modules for the IoT device.
10. We can eliminate step-10 of the ITS for the IoT update.
11. We can eliminate step-11 of the ITS for the IoT update.
12. We can eliminate step-12 of the ITS for the IoT update.
13. The IoT device applies the update and reports the application result to the “IoT integrated maintenance center/IoT devices vendor”.
14. We can eliminate step-14 of the ITS for the IoT update.
15. Finally the “IoT integrated maintenance center/IoT devices vendor” sends back a receipt of the update information. If the application of the update has failed or some remaining update is found, the “IoT integrated maintenance center/IoT devices vendor” retries the procedure from step 6 to 13 until the application has succeeded. It should be noted that number of retries should be defined in the policy statement provided by the IoT device vendor.

DISCUSSION AND FUTURE RESEARCH TOPICS

Approaches-1 and -2 only provide security controls for reducing the impact against infected malware in IoT devices and for providing initial update solution for the IoT device software/firmware update. However, the two approaches do not cover the rest of security issues for IoT environments.

The following further studies are required in connection with the two approaches in this paper:

A) Cyber-security information captured by our IoTPOT (honey) should be correctly and appropriately shared with the right stakeholders including researchers for active collaboration on IoT vulnerability analysis;

B) Remote curing method should be further investigated including evaluation under the text-bed environment;

C) IoT software/firmware update function and procedures should be further practically evaluated through experimental environments;

D) IoT security guidelines should be developed and standardised for IoT device owners, IoT service providers and IoT device vendors.

Furthermore, in addition to the above issues, the following research topics should be shared and investigated among researchers and experts:

E) Developing a generic IoT system model and reference architecture and investigating the management/ measurement of IoT security including the IoT risk assessment method;

F) Another detection method of malwares, malfunctions and/or intrusions for IoT devices (rather than using IoTPOT);

G) Study of a light-weight crypto mechanism for data confidentiality of IoT communications;

H) Appropriate Authentication and Access control utilised for IoT environments in a light-weight manner;

I) Incident handling schemes for IoT environments including threats information sharing;

J) Depending on the generic IoT model, the role of the Gateway function should be investigated including Gateway security;

K) Issues related to Privacy and Big Data under the IoT environment should be studied;

L) A secure design of application for IoT systems should be also investigated.

The research topics listed above are the initial candidates of research for IoT devices, IoT systems and IoT environments in order to kick-off the research discussion regarding IoT issues. The method ofuse of IoT devices and IoT system may differ in different regions such as in EU, US and Asia, however, the above research topics can be generally applicable for many regions with cross-region collaboration.

REFERENCES

Pa Pa, YM, Suzuki, S, Yoshioka, K, Matsumoto, T, Kasama, T and Rossow, C 2015, 'IoTPOT: Analysing the Rise of IoT Compromises', 9th USENIX Workshop on Offensive Technologies (USENIX WOOT 2015)

Yoshioka, K 2016 IoT Security - Research Center for Information and Physical Security, Yokohama National University, viewed 1 May 2016, <http://ipsr.ynu.ac.jp/iot/index.html>

Eto, M, Inoue, D, Song, J, Nakazato, Ohtaka, K and Nakao, K 2011, 'nicter: a large-scale network incident analysis system: case studies for understanding threat landscape', BADGERS 11 Proc. First Workshop Build. Anal. Datasets Gather. Exp. Returns Secure

Eto, M and Nakao, K 2016 'Secure software update capability for intelligent transportation system communication devices' ITU-T draft Recommendation