Design and Realization of the Intelligent System for Identification of the IoT Devices

J. Horalek, F. Holik
University of Pardubice, Faculty of Electrical Engineering and Informatics, Pardubice, Czech Republi.c
josef.horalek@upce.cz

***Abstract*—This paper presents a unique solution for identification and verification of IoT devices. The solution is based on actual security threats in the IoT design. For this reason, the unique identification chip DS2401 is used and devices are monitored from the centralized system. The paper introduces this system for authentication and authorization of IoT devices. These functions use the unique identification number. The whole system, including the management of identities and the user server, is realized with Raspberry Pi devices.**

***Index Terms*—Internet of Things; IoT Security; Authorization and Authentication; DS2401; IoT Identification.**

I. INTRODUCTION

A new era of the Internet of Things (IoT) brings a lot of advantages into the traditional networks, although it comes with some disadvantages. More and more devices can now be connected to the Internet and they can be therefore remotely controlled and managed. Moreover, these devices can communicate together to achieve smart functionality. These devices include smart refrigerators, TVs, and all kinds of smart sensors. Obviously, data from these devices contain very sensitive personal information about their users.

The IoT field is a very dynamic field and it is getting into all aspects of life. The number of active IoT devices is rising exponentially and it corresponds to the Gartner estimation of 6 billion connected IoT devices in 2016 [1]. Such a progressive increase will result in a large number of new types of attacks. Moreover, considering their constant usage, the level of security risks is higher than the other types of devices connected to the Internet. An attacker does not need to gain access to the system, but he/she can just simply use the read access to eavesdrop transmitted messages. A lot of devices are collecting some forms of personal information such as name, address, date of birth, or credit card numbers. This information, in combination with data from other sensor devices, can allow an attacker to predict users' behavior and life habits. This knowledge can then be misused by social engineering methods of attacks. The security risk is further increased with the IoT devices connecting into the cloud services and using mobile applications. Many devices, connected to home networks, are sending data in unencrypted form [2]. This can be easily misused if the home network is not properly configured, or if it is using non-sufficient security mechanisms [3]. Hewlett-Packard conducted an analysis [3] of the vulnerability of IoT devices with the following results:

- 90% of all IoT devices contain at least one personal information;
- 70% of all IoT devices were using unencrypted network services;
- 80% of IoT devices using cloud and mobile application components did not specify sufficient password requirements (complexity and length);
- 70% of IoT devices allowed an attacker to identify valid user account via account enumeration,
- 60% of devices using a user interface were vulnerable to large number of errors such as XSS (Cross-site scripting) and weak authentication.

An attacker can use the vulnerabilities of IoT devices, such as weak passwords, unsecured password recovery procedures, or wrong setting of authentication mechanism for gaining access to the device. Most of the devices and their cloud and mobile application components do not require sufficient password complexity and length. This means that most of these devices are using trivial passwords, which can be predicted easily. The last alarming fact concerning security is a mechanism of software updates of IoT devices. More than 60% of these devices are downloading such updates without using encrypted communication. The update files are typically not protected also. This makes it easy for an attacker to capture this communication, and to potentially manipulate with the update file. This file, containing modified software, can then be used by IoT devices without any suspicions.

II. RISKS SECURITY

Security in the IT field is a very broad topic and it covers many different subsections. In the case of IoT, the following security fields have to be considered: data, end devices, central devices, servers, databases, web services, communication, and networking infrastructure. The IoT field is very heterogeneous both in the possibilities of usage and the technologies used [4]. This results in complicated process of securing the IoT devices and dangerous impact of attacks. In the IoT deployed within smart homes, the security requirements are different than the production networks, or health care. An attacker attacking a smart home can, with a slight exaggeration, controls lighting, heating, and other appliances, which do not typically pose a significant threat (in the first phase). On the other hand, a similar attack in a production or health care network can have fatal consequences. The security is therefore a very important field for the IoT and it is influencing the usability and future development of IoT. Security requirements on IoT devices should, in the first place, include identification mechanisms and functions to ensure the integrity of users' data, users' privacy, authentication, and system trustfulness. Data security can then be divided into two layers: storage and communication. The current research in the IoT is mainly focused on the communication layer security [5-7]. A lot of

standard communication protocols are providing security on the higher level. The main problem of these protocols is the required hardware and software, which allow effective and secure running of the security algorithms. The IoT architecture is typical for its composition from several separated and isolated layers. Different rules and systems are used for setting up the communication between these layers. Each layer is defined by its functionality and type of devices used on this specific layer. Different models of architectures exist within the IoT field, but one of the mostly used, as stated in [3,8], is the four-layer architecture from Cisco, which is shown in Figure 1.

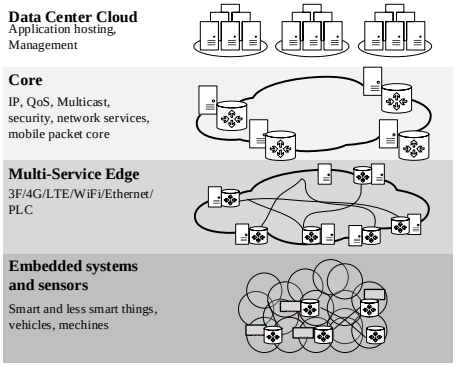

Figure 1: Cisco IoT platform architecture

The IoT architecture is specific for combining digital and physical worlds. This has severe consequences on security, such as making eavesdropping type of attacks much more serious. Consequently, well-known attacks can become new security threats on new devices that use new protocols and procedures. Many of the closed operating systems (SCADA, Modbus, CIP) are oriented as IP-based systems, making them much more prone to security issues. The IoT can be misused by different categories of security threats including:

- Common worms and viruses from the ICT world.
- Script kiddies and other attackers using: unsecured web cameras, data theft and intrusion into a smart home central system.
- Cyber terrorism – attacks on nuclear power plants (virus Stuxnet [9]), electrical grid networks, monitoring systems of critical infrastructure (railway, transport).

Although threats in the IoT can be the same as in the Internet [10-12], their impact can be completely different. For this reason, it is necessary to conduct an effective analysis of threats and their solutions in the IoT area. One of the most basic principles for ensuring security in the IoT is to identify a mechanism that can verify it. Many IoT devices have insufficient computation performance and low memory to support standard authentication protocols. Most commonly used authentication protocols, including AES (Advanced Encryption Suite) and RSA (Rivest-Shamir-Adleman), use strong encryption methods and are therefore computationally too demanding for the IoT devices. For this reason, authentication and authorization of the IoT devices must be conducted using much more efficient methods. Another security elements are placement and level of data protection, strong identification, improvement of other networking protocols (DNS, DNSEC, DHCP), and acceptance of latency-tolerant protocols. Many of the IoT use cases are considering encryption. However, IoT end devices and sensors are typically used within a long timeframe (20 years); therefore, it is important to consider long-term security of the encryption protocol. Identity management can increase security of the IoT by combining different authentication methods of users and devices. Protection of privacy and privacy compliance are connected to each other and the extent of connection is regulated by appropriate countries and regulations. With the rapid development of these technologies, users have to be aware of the influence of these issues on their lives.

A. Security Recommendations

The IoT security is a very important and challenging area for both developers and attackers. The following part describes the basic requirements on IoT security. These requirements are based on conducted analysis of security threats and attacks. The requirements can be met only if the security is included in the initial development phase. Today, there is no general solution that is able to solve all the IoT security problems [13]. Security requirements have to be considered with security failures, risk levels, and various types of attacks. Further, the cost of implementing proper security mechanisms has to be considered. The basic functions that should be considered are [14,15]:

- Secure launch – this can be achieved by cryptographically signed code from the vendor. The mechanism has to include hardware support of code verification for the code validity check. This will ensure that the firmware is not modified and is therefore secure.
- Secured updates – similarly to the secure launch, secured updates can be achieved with the signed code, eliminating the option of code tampering.
- Data security – it can be achieved by preventing unauthorized access to the device and by encrypting storage and communication.
- Verification – unauthorized access to the devices can be mitigated by methods consisting identity verification with strong password authentication, or using of protocols like X.509 or 802.11.X.
- Protection from attacks – a critical part of this layer of security is a firewall. A firewall can allow communication only for known and trusted hosts, while simultaneously block attackers before an attack is launched. It is necessary to ensure protection from common attacks (packet flood attacks, buffer overflow, and exploits – misuse of protocols' bugs).
- Detection and monitoring – current embedded devices can be attacked without notice of legitimate users. An attacker can send millions of login attempts without any record of such an activity. This can present a critical security flaw if the devices do not require strong authentication passwords (as mentioned in the HP research [3]). Embedded devices therefore have to be able to detect and notify invalid login attempts and other threats.
- Integrated security message – integration of a system for security control, which can set different security policies is important. By modifying these policies in real-time, security threats can be mitigated.

A key factor for the IoT security is authentication and authorization – or with additional accounting the AAA architecture. These methods are used for identification of IoT devices, for the following verification of their privileges, and

finally, for monitoring.

a. Authentication in the IoT

Authentication is used to uniquely distinguish and identify subjects within IoT network. These subjects can be embedded devices, sensors, actuators, or endpoints. Verification of device trustworthiness is done via authentication. The way of storing and presenting identity information in the IoT can be different from the traditional networks (which use usernames and passwords, tokens, or biometric data). In the IoT, end devices, in most cases, do not need any human interaction. It is highly encouraged that such devices can identify themselves. For such a task, radiofrequency identification (RFID), shared secret key, X.509 protocol, certificates, or hardware based authentication methods (for example based on MAC address) can be used [16].

b. Authorization in the IoT

Authorization is a process of verification access privileges of IoT devices into the IoT infrastructure. This process typically follows successful authentication and based on the identity, certain privileges are granted. The main task of authorization is therefore to verify, if the device has the privilege to conduct a specific action (such as inserting a new entry into the database). A challenge in this area is to create an architecture that is able to handle millions of devices with different level of trustworthiness [17].

c. Accounting in the IoT

This part of the AAA architecture is not dealing with security, but with log-in. Accounting is used for managing networking services and resources used by end devices. This information can be important for management, planning, and for preventing certain types of attacks. It is, for example, possible to track, which device tried to verify itself, when, and the result. For the AAA purposes, architecture with traditional computer networks, TACACS+ or 802.1x with RADIUS can be used in the IoT.

III. The Intelligent System of IoT Authentication

Our solution of the intelligent system of IoT authentication was tested in the environment of a home network, but the logical model can be implemented in industrial IoT networks as well. The main reasons for implementing in a home network were the availability of technologies and the fact that such environment is more vulnerable to security threats. The industrial IoT networks are managed by IT specialists using company's security policies, processes and procedures, which make these networks more secure.

The system handles device authentication and identification. This is done by a single centralized entity. The system also contains a web server, database, and uses communication filtering. The advantage of this implementation is embedding it in the usage of available technologies, hence it is applicable to wide public use. As a communication network, Wi-Fi was chosen due to its typical usage in most home networks. If necessary, the communication network can be adjusted to Zigbee or LoRa without the need to modify the system (only a change of the specific hardware is required). IPv4 was used for addressing this issue due to its wide usage in home networks. IPv6 however, can be used and it is fully supported by the system. The communication topology is shown in Figure 2.

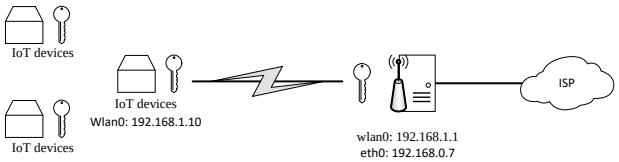

Figure 2: Communication topology

A. The Logical Model of the Authentication System

The main idea of the logical model is to assign a unique identification number to every element of the IoT network and to use this number for verification. This can be achieved with the integrated circuit DS2401. To mitigate a risk of eavesdropping and MitM attacks, the whole communication has to be encrypted. This means that firstly, a secure channel between an IoT device and the central element has to be established. Only after a successful verification, full communication can be enabled. OpenSSL library and TLS 1.2 cryptographic protocol are used for the encryption. In this case, two key pairs are used. Each of the server and a client has their own key pair. For the encrypted communication, a client does not need to be verified. Authentication of the IoT device over the unique number is conducted in the following five steps (shown in Figure 3):

- Before the authentication process can start, a secured communication using SSL/TLS is established. The secured communication could be established using certificates, but such a solution would require a unique certificate for each device. Our solution, on the other hand, assigns the same certificate for every device. This certificate only confirms that the device belongs to the network. Verification itself is then realized over the unique number.
- Authentication is initiated by the client sending an authentication request ("auth").
- The server processes the request and replies with its ID. The client receives the ID and verifies, if the server is the valid one. If the verification is successful, the client sends its ID to the server.
- The server verifies the client's ID in its database. If the ID exists, the server checks, if that device is allowed to communicate. If it is, the device is verified with the 0x123 message; otherwise, the communication is rejected with the 0x987 message.
- After the successful verification, the client can use the encrypted communication with the server on port 1111.

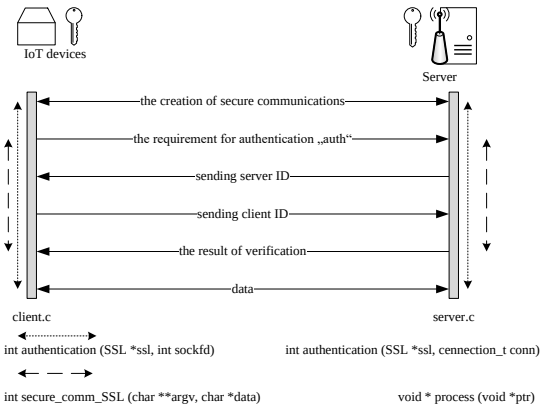

Figure 3: The authentication process

B. The Implementation of the Authentication System

As already mentioned, the IoT devices have relatively low computational performance. This is reflected in our authentication system. For this reason, we have chosen adequate devices for the system's implementation. The control entity was built using Raspberry Pi 2 platform. This device is responsible for controlling the authentication, providing a web user interface, and maintaining a database (with IoT devices, their status, and a list of users allowed to enter the control system). As a testing device requesting an authentication, older Raspberry Pi B model was used. The unique 48bit identification value was generated by the DS2401 integrated circuit. This solution has main benefits, such as its cheap cost and ease of use. The DS2401 contains the unique 64-Bit ROM ID chip for identification. This number includes 48bit serial number, 8bit CRC control checksum, and 8bit family code (this code determines the type of integrated circuit and its function). Another advantage is the use of 1-Wire bus, allowing the use of multiple DS2401 circuits on a single data line. The bus can communicate up to 16,3 kbps and reduce the control, addressing, and data transfer into a single pin connector. The power supply is provided on the data pin as well, reducing the need for external power supply [17]. For connecting the Raspberry Pi, GPIO was used. GPIO contains drivers for several interfaces and they are stored as core modules available via a *modprobe* command. This command can load appropriate modules into the linux kernel, if needed. This eliminates the need for programming a new reading method for DS2401 circuit. To set up the GPIO, the configuration file */boot/config.txt* has to be modified as follows (added lines at the end of the file):

```
#GPIO
dtoverlay=w1-gpio,gpiopin=4
```

Modules are loaded with the *sudo modprobe* command (wire, w1-gpio, and w1-smem). After the system is restarted, the name and unique ID of the connected DS2401 can be found in the file */sys/bus/devices/*. If any of the circuits are connected, a folder with a unique ID is created (using format of: 01-xxxxxxxxxxxx). The first two numbers state the family code (01 for the DS2401). After the dash, the unique 48bit code follows. This code is read by the *get_id.h* header file.

For the device management, a simple database containing the three following tables was created:

- Device – contains information about devices (identification number, MAC address, name of the device, and status – allowed / denied).
- Accounting – allows basic login functionality (status of verification, and device unique ID, name, MAC address, IP address, data, time, and status).
- Users – stores users' login credentials for application access (password is stored in SHA256 hash format).

Because the logging information has to be archived, there is no relationship between the accounting and device tables. Authentication server realized on Raspberry Pi device is providing the following services: webserver, DHCP server, database, and access point (AP). Daemon *hostapd* is used for the AP functionality (the wireless interface has to be supported by the nl80211 driver). The complete wireless setting is stored in the file: */etc/hostapd/hostapd.conf*. An additional security was achieved by using a firewall – with the iptables tool. The standard security policy of denying everything by default, was used (only the necessary communication is allowed). For the system control, web application was developed using the *Bootstrap* framework. This application allows to monitor the state of stored devices, their IDs, MAC addresses, names, and authentication state. The IoT authentication can be controlled by allowing or disabling devices access. Each device can be verified, if it is connected, by sending a ping message. The traffic of each device can be further controlled by editing traffic rules. This includes communication inside and outside the local network, and communication with the authentication server (on port 1111). Default policy is to drop everything.

IV. Discussion

The most critical features of every unique identification, as stated in [19], are uniqueness, reliability, persistence, and scalability. The existing approaches of IoT device identification are insufficient. The most common identification methods – via IPv4 or IPv6 addresses – are unable to identify specific devices within a group of cohabiting sensors, and have problems with the confluence of data from heterogeneous nodes. Although the IPv6 represents a great approach for unique identification with scalability in mind (due to its 128-bit long address), it has considerable requirements on end devices performance. This can be a problem with low-performance IoT devices like sensors. For this reason, lightweight IPv6 alternatives are being developed.

The second approach for unique identification is the Uniform Resource Name (URN) with number addressing, which can be combined with URL (easily accessible via name addressing) and URC (controllable). This approach offers a hierarchical addressing, including specific gateways. On the other hand, it requires special implementation on network nodes.

Both of these methods use only the logical addressing, which can be easily spoofed. The third approach is to use physical addresses (MAC), which can be more difficult to modify. However, they can still be spoofed relatively easily because these addresses are verified in software and sent in an unencrypted format,

Unlike the previously mentioned, our implemented approach, adds a security layer with advanced authentication. The unique authentication is implemented with the usage of the integrated circuit DS2401, and secured protocol, ensuring the confidentiality of the unique number. Moreover, our system can be combined with all the previously mentioned identification methods, or be used as a standalone solution.

V. Conclusion

This paper presents a unique solution for enhancing IoT security via identification of devices. The solution corresponds to the fact, that there is no single optimal system for securing the whole IoT. The IoT field is very complex and dynamic in its nature, so there is a demand for implementing new applications for enhancing its security. The implemented system was tested in LAN of a typical home network, but can be applied into larger scenarios, including production networks. The unique 48bit number, issued by integrated circuit DS2401, was used for device verification. This verification process can be compared to the verification over MAC address. The main advantage of our verification process, in comparison to MAC address verification, is the

fact, that the unique number will stay hidden – it is transferred via a secured channel. The communication between the devices and authentication server is realized via a socket layer. Multiple devices can be verified at once because the server is using thread programming – a new thread is created for each connection. This thread is then responsible for the device verification and communication. The process of verification is logged and stored in a database. Lastly, a responsive web interface was developed for controlling access and monitoring of the IoT devices.

ACKNOWLEDGMENT

This work and contribution is supported by the project of the student grant competition of the University of Pardubice, Faculty of Electrical Engineering and Informatics.

REFERENCES

[1] R. Van Der Meulen, "Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015," in: *Gartner*. Stamford, 2015.

[2] J. Horalek, and V. Sobeslav, "Measuring of Electric Energy Consumption in Households by Means of Arduino Platform," in: *Advanced Computer and Communication Engineering Technology*, 2016, pp. 819-830. DOI: 10.1007/978-3-319-24584-3_69

[3] Hewlett Packard Enterprise, "Internet of things research study: 2015 report," 2015 [online] Available: https://goo.gl/RRvvMh, Accessed: 2016-11-05.

[4] J. Brozek, V. Fiala, J. Fikejz, and P. Pich, "Use of industrial control unit in intelligent homes," in: ELEKTRO, 2016. Pp. 489-494. DOI: 10.1109/ELEKTRO.2016.7512124

[5] O. Vermesan, D. Ovidiu, and P. Friess, "Internet of things: converging technologies for smart environments and integrated ecosystems," 2013, ISBN 9788792982964.

[6] S. Babar, P. Mahalle, A. Stango, N. Prasad, and R. Prasad, "Proposed Security Model and Threat Taxonomy for the Internet of Things (IoT)," DOI: 10.1007/978-3-642-14478-3_42.

[7] S. Babar, A. Stango, N. Prasad, J. Sen, and R. Prasad, "Proposed embedded security framework for Internet of Things (IoT)," in: *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE)*, 2011, pp. 1-5. DOI: 10.1109/WIRELESSVITAE.2011.5940923.

[8] J. Yang, and B. Fang, "Security model and key technologies for the Internet of things," in: *The Journal of China Universities of Posts and Telecommunications*. 2011, pp. 109-112. DOI: 10.1016/S1005-8885(10)60159-8. ISSN 10058885.

[9] M. Kelley, "The Stuxnet Attack On Iran's Nuclear Plant Was 'Far More Dangerous' Than Previously Thought," 2013, [online] Available: https://goo.gl/00KK9l, Accessed: 2016-10-01

[10] R. Weber, "Internet of Things – New security and privacy challenges," in: *Computer Law & Security Review*. 2010, DOI: 10.1016/j.clsr.2009.11.008.

[11] Q. Jing, A.Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the Internet of Things: perspectives and challenges," in: *Wireless Networks*. 2014, pp. 2481-250. DOI: 10.1007/s11276-014-0761-7. ISSN 1022-0038.

[12] C. Medaglia, and A. Serbanati, "An Overview of Privacy and Security Issues in the Internet of Things," In: The Internet of Things. 2010. pp. 389. DOI: 10.1007/978-1-4419-1674-7_38.

[13] S. Sicari, A. Rizzardi, L. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," in: *Computer Networks*, 2015, pp. 146-164. DOI: 10.1016/j.comnet.2014.11.008. ISSN 13891286.

[14] L. Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," in: *IEEE Transactions on Industrial Informatics*. 2014 pp. 2233-2243. DOI: 10.1109/TII.2014.2300753. ISSN 1551-3203.

[15] J. Horalek, and V. Sobeslav, "IPv6 firewall functions analysis," in: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, pp. 219-228. DOI: 10.1007/978-3-319-45246-3_21.

[16] T. Svoboda, and J. Horalek, "Analysis of security possibilities of platforms for 3D graphics," in: *Journal of Telecommunication, Electronic and Computer Engineering*, 2016. pp. 43-47.

[17] F. Holik, J. Horalek, O. Marik, S. Neradova, and S. Zitta, "Effective penetration testing with Metasploit framework and methodologies," in: *CINTI 2014 - 15th IEEE International Symposium on Computational Intelligence and Informatics, Proceedings*, 2014, pp. 237-242. DOI: 10.1109/CINTI.2014.7028682.

[18] DS2401 Silicon Serial Number. 2015. *Maxim Integrated* [online] available: https://datasheets.maximintegrated.com/en/ds/DS2401.pdf Accessed: 2016-09-17.

[19] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions" in: *Future Generation Computer Systems*, 2013, pp. 1645-1660. DOI: http://dx.doi.org/10.1016/j.future.2013.01.010