Authentication and Authorization Rules Sharing for Internet of Things

Michal Trnka and Tomas Cerny

*Dept. of Computer Science, FEE, Czech Technical University, Technická 2,
166 27 Praha, Czech Republic
E-mail: {trnkami1; tomas.cerny}@fel.cvut.cz*

Received 31 October 2016; Accepted 19 December 2016;
Publication 29 December 2016

Abstract

Significant interest in internet of things drives both research and industry production these days. A lot of important questions has been solved but some remain opened. One of the essential unresolved issue is the identity management of particular end devices.

Having possibility to share authorization and authentication rules across network between various sensors, applications and users have clear advantages. It reduces duplication of those policies, while ensuring that they are coherent across the board. There are various proposals, methods and frameworks for identity management in normal environment. However, just few of them is usable in internet of things environment and even less have been made directly for the usage in the internet of things devices.

This paper proposes solution for management of devices for internet of things. The solution is based on central identity store. Each device has an associated account in the store that any device and application in the network can verify the device's identity against. The central element does not provide only authentication of the devices, but the devices can be associated with different roles. Those roles can be used for authorization.

Case study for the proposed framework is built on top of the current web standards as OpenID Connect, OAuth and JSON Web Token. Also, central identity store is based on well-established open source solution.

The communication scheme is very simple – after the device’s account is established, it retrieves OAuth 2.0 token and uses it to certify itself in every network connection. The token does not only contain authentication information, but also roles assigned to the device. The central element thus creates trusted environment and enables rapid response for any security events.

Keywords: Internet of Things, Security, Identity Management, Authentication, Authorization.

1 Introduction

The notion Internet of things (IoT) is currently getting a lot of attention and first real usages are taking place in real world scenarios. For instance, Gartner Inc. [6] predicts that by 2020 there will be 26 billion units installed in IoT products. Although to make IoT solutions ready for production-level quality outside of controlled lab environment multiple issues needs to be solved. Security and privacy challenges are considered to be one of the most crucial [1, 7]. Gartner company states [6]: The Internet of Things concept will take more than 10 years to reach the Plateau of Productivity – mainly due to security challenges, privacy policies, data and wireless standards, and the realization that the Internet of Things requires the build-out of a topology of services, applications and a connecting infrastructure.

The IoT builds on an idea that a single device may not provide any significant and useful functionality and therefore it cooperates with multiple other devices in its neighborhood. An individual device needs to trust other devices in order to deliver any value for a user. The trust must be established not only among devices and but also between the user and the devices.

However, device management and creation of confidential environment between them is one of the major open issues in IoT [19]. IoT can be divided into three layers – perception, transport and application. The device identity management must be implemented at least on application layer as it is responsible for all communication with end user and significant part of communication with devices, as it gathers all relevant data for user.

This paper proposes a framework that creates trusted environment of devices around centralized identity store. It allows not only authentication, but also complete device identity management. The following text describes tools for connecting devices into the network and their initial verification, processes for devices invalidation and also for their basic identity management (e.g. assigning devices with roles). The framework also allows response fast

enough to prevent any further damages in case of an attack targeting devices as described by Jing [10]. This paper brings a case study presenting a real example of the devices authentication and authorization involving proposed framework.

The rest of the paper is organized as follows. Section 2. provides related work. Next a security framework proposal is introduced. A case study is provided in Section 4. Finally, we conclude the paper.

2 Related Work

Seitz et al. [20] propose a framework with Authorization Engine as Trusted Third Party. It resolves authorization requests for constrained devices when user wants to access any of devices data or service. The Authorization Engine is exposed through REST API and therefore most of the IoT devices can access it. The overall architecture is very similar to our and the authors propose very complex framework built on top of the existing technologies. However, they deal with user-device security and this work does not address machine to machine trust.

Kothmayr et al. [13] surveys two-way authentication scheme for IoT devices using Datagram Transport Layer Security protocol. While the proposed solution describes feasibility of the protocol usage for point to point communication, it does not describe any management and administration of multiple devices.

Zhang et al. [22] describe distributed privacy-preserving access control in sensor networks. Token based approach in single owner multi-user sensor networks is used to preserve users' privacy with central token issuing authority. The work is focused on the issues with token reusability. Although, their methods are promising, the protocol itself does not solve fine grained access control policies.

Solution suggested by Pranata et al. [16] uses capability tokens for granting access to services. The token is issued by provider for a client during the initial provision of a service. It consists of a URL, user/object ID, time stamp of creation, subscription period, service ID and list of access permissions. Though, the paper does not address sharing of the access permissions across providers, how to deal with overtake of a client, and most importantly, how the avoid duplication of information for authentication and authorization of clients on service providers.

Nehme et al. [15] describes framework for data authentication in IoT. Data are treated as streams with security punctuations, which get analyzed before

the data are provided to its consumers. The work does not describe how to apply security rules, therefore MAC, DAC and RBAC can apply for such solution. Nevertheless, the work expects the data creator to set restrictions on them, which is unsuitable in many applications, when the sensor is just a slave of another hierarchically, a higher entity (for example temperature meter of home network server). Moreover, it does not address problem of security rules duplication and consistency across different devices.

Roman et al. [18] discusses possibility to use existing link-layer mechanisms link Public Key Cryptography and pre-shared keys for sensor nodes in the IoT as well as applicability of those link-layer oriented key management systems. They focus only on devices connected through wireless network and they find those methods and protocols mostly unsuitable for IoT devices (with some exceptions).

Lightweight encryption method based on XOR manipulation described by Lee et al. [14] can be used also for anti-counterfeiting, privacy and mainly for authentication. However, the proposal is very brief and mainly is done by device hardware. Which means that the IoT devices would need to be manufactured to work with this method. The consequence is that all devices in network would need to have this hardware predisposition and no other device (for example older ones) could operate in given environment, same as those devices could not operate in network of other devices.

Ye et al. [21] describes detailed authentication and access control method based on the basis of Elliptic Curve Cryptography, a lightweight encryption mechanism. The authentication must be triggered manually and on top of that their proposal contains attribute-based access control for authorization. Together it solves the resource-constrained issue at application level in IoT. Usability of that solution is limited for programmers as they do not provide any framework or proposal for one to allow building application for IoT easily based on their method.

Interesting topic of identity management, user's privacy and authentication discuss Hu et al. [9] in his paper. He describes system that can provide location in emergency situations, while under normal circumstances, the user's location information is confidential. Two most interesting parts are: user's authentication and policy subsystems. The system confirms the identity of the user through the user authentication subsystem and gets the level of the emergency through the policy subsystem. Although their idea is very interesting, they authentication system is only vaguely described. It does not address authentication of devices and therefore it could not be used for general use in IoT.

Let's summarize the current research discussed above. There are two proposals [14, 18] focusing on particular domain of IoT space. While their research is significant, it can't be used generally for any type of network and device. Another research [9] shows an interesting usage of context aware security for IoT, but only vaguely describes authentication methods. Two articles [16, 22] suggest solutions based on tokens for authorizations, unfortunately, they do not describe authentication part well enough. They lack in terms of propagation of security rules across the network. An excellent research for general means of securing of the data in the IoT environment provides [13, 20]. However, they do not address two significant use-cases – device to device and device to user communication. Other proposals, such as [13, 15, 21] provide promising methods for authentication and authorization but do not solve the issue with security rules duplication on every device or sharing the authentication and authorization policies across devices and applications.

As the paragraphs above shows, the IoT device security is a topic that currently gains a lot of attention. Nevertheless, despite tremendous amount of work done and effort put in the research a lot of questions still needs to be answered. One of the many problems, that needs to be solved, we aim to propose a solution for centralized authentication and authorization service to reduce duplication and to unify security rules across the network.

3 Security Framework Proposal

This paper proposes a framework with a central identity store, which would keep the record for every device connected to the network. It contains not only unique identifiers for devices but also supports the Role-Based Access Control (RBAC) [2] by storing the roles internally. All machines and applications in the network can use those roles for their authorization rules. Such trusted central identity provider can provide environment, in which both participants can verify identity of the second partner and also they can determine if the partner is allowed to perform the given action.

Communication consists of two participants. The first one, called provider, exposes services to others. If the provider decides that its services are secured it must register at the identity store as an identity client. The second one, consumer, uses providers functionality and initiates the communication. To initiate communication with secured service, it needs to acquire a token from the identity store that is valid for the particular provider. This enables authentication at the identity store and at the same time it prevents misuse

Figure 1 Diagram of device's possible states.

by malicious service providers. Figure 1 demonstrates possible states of the devices.

The identity store does not need to serve solely as an authentication service, it also provides additional functionality. In this case devices are provided with authorization roles. The service provider can specify roles allowed for a given action. When the consumer tries to use the service its roles are verified at the identity store. This also means that the roles are global in the specific IoT environment, which reduces efforts related to administration and the number of repeating configurations across all systems.

Figure 2 demonstrates the work-flow of devices authentication and authorization. It can be described in the following steps:

- Administrator creates an account for a device and set up its roles.
- The device is configured with credentials provided by the administrator and requests a token from the store.
- For any confidential communication the device uses the token to authenticate itself.
- Application/device receiving the communication verifies the identity and roles by given token at the central store.
- Administrator can disable or remove a device from the identity store and therefore effectively disable it for any cooperation.

Figure 2 Diagram of communication in the proposed framework.

The framework by itself consists of server part, administration application and then communication module for IoT devices. Figure 3 shows component diagram of our framework.

- IDM module – module that administers devices and their roles. It also verifies tokens.
- Device provider – allows devices to log in and refresh token.
- Administration provider – module that enables administrator to add, remove or disable devices.

Figure 3 Component schema of the framework.

- Administration application – application that provides UI for the administrator.
- Communication module – module which is embedded into the IoT device and takes care about every communication. It authenticates the device, retrieves the token and uses it for further communication.
- Sensor module – This module contains business logic of the IoT device and it is not part of the framework.

- Application – Application that uses data from sensor. It needs to verify the device's token against IDM server. This component must be programmed by end-users of the framework.

Using the central identity element in IoT promotes a trusted environment. Devices do not deal with machine to machine trust, it is enough if they establish a confidence in the identity store. Whenever there is a suspicion about a hostile takeover of any device, the device can be withdrawn all its roles. This action ensures immediate propagation through the whole network. This kind of approach applies also to less serious situations, such as device malfunction resulting with transmission of faulty data. However, using any central element in network architecture has known security threats, as for example Denial-of-Service attack, and therefore need to be sufficiently protected.

To illustrate need for such solution let's imagine following situations. In the first one we have a car equipped with location sensor and connection to the internet. Such vehicle would be able to provide its location on request. In the base configuration, the location could be used only in emergency situations. However, the user may want to change the settings. E.g. he decides to participate in any form of smart transportation. Or insurance company offers him lower rate based on his small annual mileage. Having central identity store in those cases would make everything easier. He would just allow devices with role "insurance locator" or "London smart traffic" to communicate with the vehicle.

Use case above requires trusted central identity. Which is something very difficult to get in real world. However, there is another use case situation based on smaller scale IoT network – smart home. For example, there is a family that built new house and used modern technologies in it. The house can keep constant temperature inside for whole year. Also, in order to minimize costs, it uses most energetically efficient way to preserve stable environment. Based on the temperature in different rooms it can turn on heating, air conditioning. In addition to that it can open windows to cool down the building in colder weather (for example in night it can decide to open windows instead of turning on air conditioning). This family decides to leave for longer vacation and leaves. In this moment temperature sensor in one room starts malfunctioning. It randomly gives too low or too high measurements. With central identity store, owner can disable the sensor as soon as he finds out something weird is going on and this sensor will not interfere with other devices any more.

Configuring a device in such a network does not require significant efforts. First, the device is setup provided with credentials. Before it initiates communication with its partner, it requests a token with credentials, valid solely for the given service provider, restricted to a certain period of time.

In certain cases, a time-unlimited token is viable, in others token with a short-time validity is preferred. However, once the device obtains a token, it can communicate freely with the partner. The partner can verify device identity as well as its roles, based on the presented token.

The framework supports communication over REST API [17]. This allows utilization of all the technologies and properties of the HTTP protocol [3]. At first, SSL protocol [5] is tightly integrated with the HTTP protocol (called HTTPS [3]), which provides us transportation security as added identity confidentiality in the network. The advantage brought by this approach is that firewalls rarely block communication on ports 80/443. There might exist potentially more suitable protocol than HTTP(S), which is tied to REST architecture, however none of them is so widely used and adapted as HTTP(S).

4 Case Study

Based on the framework proposal described in the previous section we form a prototype that builds on existing solutions (as suggested by Finkelstein [4]) integrated together to provide the expected functionality. Building on top of existing solutions allows us leveraging user efforts and simplify the transition to the real usage. Furthermore, using existing infrastructure does not constrain us with other issues to address, and mostly it brings a verification to the proposals applicability and integrability with existing production-level tools. Moreover, it ensures that current state of art is sufficient for extension and no further crucial technologies needs to be developed as a replacement.

Roman et al. [19] states that also traditional Web 2.0 SSO such as OpenID and Shibboleth could also be used in this situation, although it should be noted that they were not designed to fulfill certain IoT requirements such as identity disclosure. Therefore, we have opted to try out existing technologies to determine if (and how) they are sufficient for usage in the IoT ecosystem.

Small scale simulation of IoT was created for the purpose of that paper. Figure 4 shows scheme of our case study application. It consists of those major elements:

- Central identity store – Keycloak¹ was chosen as it provides Single-Sign-On and identity management for web applications and mainly for RestFul web services. For our purposes was leveraged mainly support of Oauth 2 [8] and OpenID Connect² JSON Web Token [11] standards.

¹<http://keycloak.jboss.org/>

²<http://openid.net/connect/>

Figure 4 Scheme of our case study application.

- Two sensors – in particular movement sensor HC-SR501³ and temperature sensor DS18B20⁴ were used. Both of the sensors provide digital output and therefore can be used without any analog-to-digital converter. However, sensors still need to be connected to some device with computational capabilities to transmit the data over the internet. In this case, Raspberry Pi is used to host sensors services.
- An application using data from sensors – simple application with RESTful interface⁵. It gathers data from sensors and exposes them to users via JavaScript web front end.

Central identity store is deployed as a standalone application. It consists of two roles *temperatureSensor* and *movementSensor*. Next, an account for every device was created and assigned appropriate roles. The password and username of the given account must be provided to the particular sensor. Authentication token is set to never expire. This allows to a device to use it as long as needed without need to refresh it periodically. OAuth 2 protocol is used for bearer token [12] acquisition and the token issued follows JSON Web Token (JWT) standard [11]. The advantage of using JWT tokens is that they may contain additional information, such as user roles, etc. Therefore, the communication with the central identity store can be reduced to a single call aggregating multiple information and thus improving the performance.

The sensors by itself do not possess any computational power and therefore they need a device to control and observe them. In our case study, they are directly wired onto to the bus of Raspberry Pi computer. Small script written

³https://github.com/slowercz/rpi_HCSR501/

⁴https://github.com/slowercz/rpi_DS18B20/

⁵https://github.com/slowercz/keycloak_iot

in JavaScript on top of Node.js framework performs all the logic related to the sensors. The script differs for various types of sensors and needs to be initialized with credentials for the particular sensor. The communication process is following:

- Acquire token with username/password.
- Every second send information to the central application, with token for authentication and authorization.
- If token becomes invalid, attempt to acquire new one.

All the communication between sensors, the central identity store and the central application are made through RESTful interfaces. As the sensors are managed by JavaScript service additional mocked sensors can be deployed into the case study environment, not impacting the scalability of the infrastructure.

The central application receives data from sensors and display them at a web page. In order to do so the application consists of two parts – backend and frontend. The backend part uses Java EE and it leverages the Keycloak adapter to make integration with central identity server easier. It provides RESTful interface for gathering data from sensors and also for exposing the gathered information. The frontend part of the application is also connected to this RESTful interface.

The case study demonstrates that using proposed scheme is possible and it enables the expected advantages, such as broad machine to machine trust and rapid incident reaction. However, it also shows limitations that should be addressed. First, there is a need to distribute credentials for every sensor, store it at the device and use it for obtaining a token. Second, for every sensor an administrator needs to manually create an account, set up its roles and propagate identification and password to sensor.

4.1 Performance Evaluation

The performance overhead of our case study is very low. Our measurement show that it takes from 115 ms to 130 ms (with mean time 123 ms) to retrieve or refresh token. This was measured in the control Node.js program controlling the sensor. Validity of the token can be determined by system administrator and can vary from single request up to unlimited. Illustration of the amount of the that device spends managing security token can be seen in the Table 1. As we can see, the overhead would become significant only if the data were sent from the sensor every second with a single usage token.

Table 1 Overhead of sensor communication

Token Validity	Percentage of Device Time in the Worst Scenario
1 second	13%
1 minute	0.21667%
5 minutes	0.04333%
1 hour	0.00361%
1 day	0.00015%

The volume of the data sent exhibits at the first sight significant increase. The data are transferred from sensor using HTTP GET method. It does not contain any body and therefore length of the request is small. For example, temperature sensor's requests are only 189 bytes large. Out of the total, there is 152 bytes for URL address, 6 bytes for the data itself and 31 for various symbols needed in HTTP request, such as headers. With security added, the HTTP request changes the size to 1398 bytes. The token by itself is 1185 bytes long. It may look like a huge overhead, however, 1kB of data added is insignificant increase with current Internet technologies.

4.2 Validity Threats to the Case Study

The measurement for our case study was performed in very small environment. The application ran on the same computer as was used for user's connection and validation. The sensor network consisted of 2 real sensors – one motion sensor and one temperature meter. Both of them were connected to Raspberry Pi that administered both of them. We did not have sufficient resources to simulate large scale IoT environment. In such case the performance is questionable. Based on the performance of the Keycloak we strongly believe that thousands of sensors should be manageable. However, we can't say now where the limits of the solution are. Whether it is tens of thousands of sensors, hundreds of thousands or even millions.

Based on the same issue with small testing network we did not try more than two roles for authorization. There is no doubt that device and central store can manage significantly more than any device would ever need. Nevertheless, there is still need to administer them. We used RBAC system, which can become hard to maintain with increasing number of roles. We assume that more than 100 roles would become hard to manage. However, there is currently no research stating how many roles would be needed for IoT environment.

5 Conclusion

This paper addresses IoT device management with the main focus on their authentication and partially authorization. A framework proposal is shown including a case study of experimental implementation of the suggested solution.

Currently there exists various articles, works and frameworks for authentication and authorization in IoT environment. Although, none of them provides all of the needed requirements that are – security rules reusability, authentication and authorization both device-device, user-device and application-device and mainly to be currently ready for use. This is why we have created our proposal and verified it on small scale case study.

The key component of the framework is the central identity store. It enables administration of all devices from a single place and promotes trusted environment. All devices have an account associated with roles. This enables not only authentication but also basic authorization. The authentication follows OAuth 2 protocol and all the communication uses JWT token. In case of any security incident, the given device can be disabled by administrator at a single place and consequently the device will not be able to participate in any further communication.

However, the presented solution exposes few limitations. It uses only basic RBAC authorization and it solves security only on application layer. It does not address network security as well as it cannot protect devices from physical manipulation. This paper does not provide any performance evaluation at production environment, which remains for future work.

Acknowledgment

Research described in the paper was supported by the Grant Agency of the Czech Technical University in Prague, under grant No. SGS16/234/OHK3/3T/13.

References

- [1] Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Comput. Netw.* 54, 2787–2805.
- [2] Ferraiolo, D., and Kuhn, R. (1992). “Role-based access control,” in *Proceedings of the 15th National Computer Security Conference*, Baltimore, MD, 554–556.

- [3] Fielding, R., and Reschke, V., (2014). *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. RFC 7231 (Proposed Standard)*.
- [4] Finkelsteiin, A., and Kramer, J. (2000). "Software engineering: a roadmap," in *Proceedings of the Conference on The Future of Software Engineering, ICSE '00*, (New York, NY:ACM), 3–22.
- [5] Freier, A., Karlton, P., and Kocher, P. (2011) *The Secure Sockets Layer (SSL) Protocol Version 3.0. RFC 6101 (Historic)*.
- [6] Gartner Inc. (2013). *Hype Cycle for the Internet of Things*. Stamford, CT: Gartner.
- [7] Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): a vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* 29, 1645–1660.
- [8] Hardt, D. (2012). *The OAuth 2.0 Authorization Framework. RFC 6749 (Proposed Standard)*.
- [9] Hu, C., Zhang, J., and Wen, Q. (2011). "An identity-based personal location system with protected privacy in iot," in *Proceedings of the 4th IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, Shenzhen, 192–195.
- [10] Jing, Q., Vasilakos, A. V., Wan, J., Lu, J., and Qiu, D. (2014). Security of the internet of things: perspectives and challenges. *Wireless Netw.* 20, 2481–2501.
- [11] Jones, M., Bradley, J., and Sakimura, N. (2015). *JSON Web Token (JWT). RFC 7519 (Proposed Standard) RFC 7797*.
- [12] Jones, M., and Hardt, D. (2012). *The OAuth 2.0 Authorization Framework: Bearer Token Usage. RFC 6750 (Proposed Standard)*.
- [13] Kothmayr, T., Schmitt, C., Hu, W., Michael B., and Carle, G. (2013). {DTLS} based security and two-way authentication for the internet of things. *Ad Hoc Netw.* 11, 2710–2723.
- [14] Lee, J. Y., Lin, W. C., and Huang, Y. H. (2014). "A lightweight authentication protocol for internet of things," in *Proceedings of the 2014 International Symposium on Next-Generation Electronics (ISNE)*, Taoyuan, 1–2.
- [15] Nehme, R. V., Rundensteiner, E. A., and Bertino, E. (2008). "A security punctuation framework for enforcing access control on streaming data," in *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, (Washington, DC: IEEE Computer Society), 406–415.
- [16] Pranata, H., Athauda, R., and Skinner, G. (2012). "Securing and governing access in ad-hoc networks of internet of things," in *Proceedings of the IASTED International Conference on Engineering and Applied Science*, EAS 84–90.

- [17] Richardson, L., and Ruby, S. (2007). *Restful Web Services*. O'Reilly, first edition.
- [18] Roman, R., Alcaraz, C., Lopez, J., and Sklavos, N. (2011). Key management systems for sensor networks in the context of the internet of things. *Comput. Electr. Eng.* 37, 147–159.
- [19] Roman, R., Zhou, J., and Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things. *Comput. Netw.* 57, 2266–2279.
- [20] Seitz, L., Selander, G., and Gehrmann, C. (2013). “Authorization framework for the internet-of-things,” in *Proceedings of the 2013 IEEE 14th International Symposium and Workshops on the World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 1–6.
- [21] Ye, N., Zhu, Y., Wang, R.-C., Malekian, R., and Qiao-min, L. (2014). An efficient authentication and access control scheme for perception layer of internet of things. *Appl. Math. Inf. Sci.* 8, 1617.
- [22] Zhang, R., Zhang, Y., and Ren, K. (2012). Distributed privacy-preserving access control in sensor networks. *IEEE Trans. Parallel Distrib. Syst.* 23, 1427–1438.

Biographies

M. Trnka received master degree in computer science from Faculty of Electrical Engineering of Czech Technical University in Prague and he is currently Ph.D. student in Czech Technical University in Prague. His area of interests is software engineering and especially in context aware applications security. Lately he focused on using context information to enhance security rules as well as methods to obtain context from trusted sources, like Internet of Things devices.

T. Cerny received his Bachelors and Engineers degrees from the Faculty of Electrical Engineering of Czech Technical University (FEE, CTU) in Prague, Czech Republic. Next, he received his M.S. degree from Baylor University and in 2016 finished Ph.D. in Information Science and Computer Engineering. Since 2009, he works as an Assistant Professor at FEE, CTU. His area of research is software engineering, separation of concerns, model-driven development, enterprise application development and networking.