POSTER: IoT Application-Centric Access Control (ACAC)

Mohammed Al-Shaboti
alshaboti.it@gmail.com
School of Engineering and Computer Science, Victoria University of Wellington
Wellington, New Zealand

Ian Welch
Ian.Welch@ecs.vuw.ac.nz
School of Engineering and Computer Science, Victoria University of Wellington
Wellington, New Zealand

Aaron Chen
aaron.chen@ecs.vuw.ac.nz
School of Engineering and Computer Science , Victoria University of Wellington
Wellington, New Zealand

ABSTRACT

As smart environments become more common, IoT applications can automate more complex and dynamic activities. Users can define their activities as abstract workflows and suitable devices will be selected dynamically to execute them based on user quality of experience (QoE) requirements. However, many of such applications violate the principle of least privilege in terms of the allowed interactions between the IoT devices. We propose an Application-Centric Access Control (ACAC) framework to enable least privilege network access control for dynamic workflows while considering users' QoE. ACAC enables automatic derivation of an access control policy for an IoT application and allow this to be adjusted dynamically as new devices come and go in order to maintain user QoE.

KEYWORDS

Internet of Things (IoT), network access control, IoT applications

ACM Reference Format:
Mohammed Al-Shaboti, Ian Welch, and Aaron Chen. 2019. POSTER: IoT Application-Centric Access Control (ACAC). In *ACM Asia Conference on Computer and Communications Security (AsiaCCS '19), July 9–12, 2019, Auckland, New Zealand.* , 3 pages. https://doi.org/10.1145/3321705.3331008

1 INTRODUCTION

Recent years have witnessed a rapid increase in Internet of Things (IoT) applications in smart environments. In 2022 according to Gartner, a typical smart home network will have up to 500 smart devices [9]. Many IoT cloud-based frameworks, such as Microsoft Flow [8], have been proposed to manage IoT applications effectively. With the help of such frameworks, users can automate their activities easily as *workflows*. For example, a workflow involving alarm and coffee maker can be set up to prepare coffee at a coffee time automatically. Various IoT applications rely upon local cross-device communications [13].

Existing research has studied how to define user's activity as *abstract workflows* that capture task and structure level, leaving resource selection to service discovery that determine a set of specific resources (i.e. IoT devices) for task execution, called *concrete/execution workflow* [10, 11]. This provides ability to the users to define workflows in a flexible way, and managing dynamic workflows easier. Workflows can change dynamically due to many factors such as user mobility and Quality of Experience (QoE) [3, 4, 7]. For the same example, a workflow may prepare coffee using different coffee maker each time based on user location to improve user's QoE. Driven by this understanding, many researchers have investigated dynamic access control for IoT [5, 6]. However, there is a lack of research on IoT dynamic access control that supports dynamic workflows while considering QoE.

In this paper, we will study how to support dynamic workflows through dynamic network access control policy that follows the principle of least privilege (PoLP) and satisfies QoE requirements?. Access Control List (ACL) approach is a common way for access control, especially in the IoT network [5]. The capability-based model assigns rights to subjects a prior to its current context [5], however in ACL model access is granted based on attributes/context that might be static or change dynamically. Moreover, an ACL model can be enforced at the network level, this has the benefit that there is no need to install special libraries on clients such as IoT devices [2]. This is important in the context of IoT where it may be impossible to modify any program code shippes as part of the device. We argue that this makes applying an ACL model a practical choice for many smart homes.

In previous work [1] we have studied automatic ACL generation for user-defined workflows to maximise user preferences in a static environment. In this paper, our research focuses on proposing an IoT security architecture that enables dynamic workflow access control without seriously affecting users' QoE. The proposed architecture called IoT Application-Centric Access Control (ACAC) which implies access control decision rely on the application workflow dynamic state.

The dynamicity of workflows significantly increase the difficulty of generating access control policy. To illustrate these challenges let us consider the example in Figure 1 where two activities modelled as two workflows must be automated; workflow W_1 allows a TV to audio stream through a speaker, and workflow W_2 enables an alarm clock to trigger a coffee maker. A user is located in a smart environment that contains several devices including TV tv_1, speaker (s_1, s_2), alarm a_1, and coffee maker c_1.

At time t the activity workflow W_1 is realised by TV tv_1 and speaker s_1, both located in the living room. Accordingly, the ACL policy P_1 has been generated and enforced to provide connections between tv_1 and s_1, and block other connections (Dynamic Access Control Policy (DACP) is presented in Section 2.3). Then, at time $t + \Delta$ the second workflow W_2 is introduced to automate coffee preparation. This time, the alarm a_1 (in the living room) and the

AsiaCCS '19, July 9–12, 2019, Auckland, New Zealand

ACM ISBN 978-1-4503-6752-3/19/07.
https://doi.org/10.1145/3321705.3331008

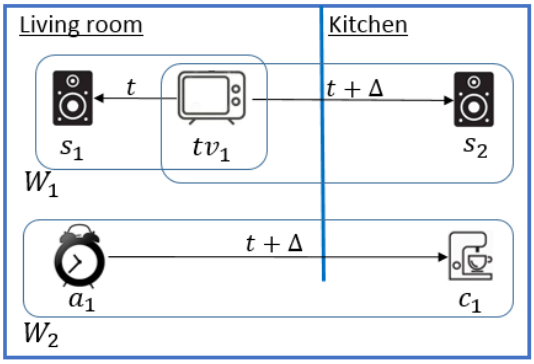

Figure 1: An example of Dynamic workflows: In Workflow W_1 TV tv_1 audio streaming changed to kitchen speaker s_2 at time t_Δ where a user will drink coffee prepared by coffee maker c_1 in workflow W_2.

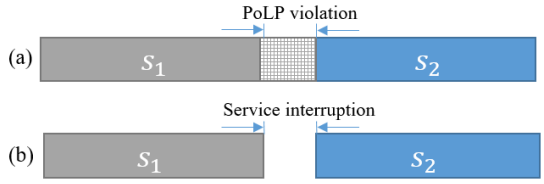

Figure 2: TV tv_1 stream handover access control from s_1 to s_2.

coffee maker c_1 (in the kitchen) are selected for this task. Accordingly, the ACL policy P_2 has been generated and enforced to provide connections between a_1 and c_1 and block other connections.

However, upon setting up the new workflow W_2, due to users' mobility that leads to changed preference, TV tv_1 of workflow W_1 needs to stream to the speaker in the kitchen (i.e. s_2), instead of s_1 in the living room, see Figure 1.

The challenge is how to dynamically update workflow W_1 policy P_1 that obeys the PoLP and minimising the impact on QoE. In particular, P_1 policy should block connection $conn_1$ (i.e. tv_1 to s_1) and allow connection $conn_2$ (i.e. tv_1 to s_2). However, this is not a trivial problem, because there is a trade off between the PoLP and minimising the interruption during streaming handover. For example, if $conn_2$ is allowed before the handover, then this violates the PoLP (tv_1 has access to s_1 and s_2 while it only requires one of them), see Figure 2(a). On the other hand, if $conn_1$ is blocked before the handover, this streaming service will be interrupted (i.e. tv_1 can't access s_1), as demonstrated in Figure 2(b). Therefore, for an access control mechanism to support such dynamic workflows it needs to be aware of workflows changes in the application level.

Motivated by existing technologies that a) enable dynamic workflow service discovery that consider user QoE [3] and b) support dynamic access control which enforces the PoLP [5, 6]. We introduce the following research question what is the proper way to enforce dynamic access control that obey the PoLP meanwhile not affecting user's QoE?. We investigate this question by designing application-centric access control (ACAC) framework for dynamic

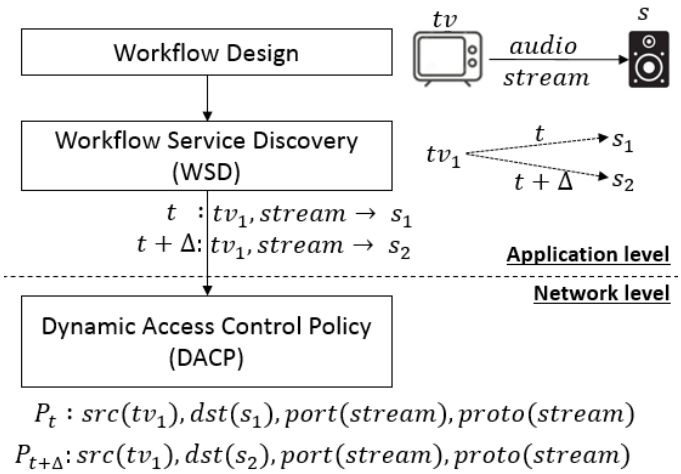

Figure 3: ACAC framework architecture.

IoT applications that obeys the PoLP meanwhile considering users QoE. The proposed ACAC framework introduces an interface between application level and network access control level, see Figure 3. This channel of interaction enables DACP to enforce access controls while considering to minimise any interruption on ongoing services on the application level.

2 THE ACAC FRAMEWORK

The proposed ACAC framework is designed to enable the least privilege network access control for dynamic workflows while considering users' QoE. This is possible by allowing notification to flow from application to access control policy generator. The three main components of the ACAC framework, as depicted in Figure 3, are: workflow design, workflow service discover (WSD), and Dynamic Access control policy (DACP).

2.1 Workflow Design

IoT application level *workflow design* is where users define their activity/task as *abstract workflows*. It provides only service semantic information on how the workflow is composed without any execution details. Workflow design main goal is to decouple the activity from the underlying devices that it is going to run it. Hence, devices can be selected/changed dynamically to build *execution workflow* to satisfy user's requirements (e.g. QoE and preferences). For example, a user defines *abstract workflow* for audio streaming activity as TV tv audio streams to a speaker s, and left the selection of the suitable TV and speaker devices to the workflow service discovery.

2.2 Workflow Service Discovery (WSD)

The workflow service discovery (WSD) manage and maintain *workflow execution* to meet users' QoE and preferences. WSD discovers and automatically select a suitable collection of IoT devices to provide required services for a new workflow [10]. The devices that execute the workflow are selected among available devices in the network to maximise user QoE and preferences. Moreover, WSD dynamically changes/switches workflows' services from one device

Algorithm 1: Policy Generation Algorithm

Input : Workflow changes W_c, Service requirements Net_{req}, Network devices Net_{dev}, ACL policy acl

Output: New/updated ACL policy acl

foreach *change* $c_i \in W_c$ **do**
  $srcIP \leftarrow Net_{dev}(c_i.FROM, IP)$;
  $dstIP \leftarrow Net_{dev}(c_i.TO, IP)$;
  $dstP \leftarrow Net_{req}(c_i.TO, c_i.SERVICE, PORT)$;
  $proto \leftarrow Net_{req}(c_i.TO, c_i.SERVICE, PROTO)$;
  $rule \leftarrow (\{src\_ip = srcIP, dst\_ip = dstIP, dst\_port = dstP, tp\_proto = proto\}, allow)$;
  if $c_i.ACTION == REMOVE$ **then**
    $acl- = rule$;
  else
    $acl+ = rule$;
  end
end

to another to meet users' QoE and preferences [3, 4]. Workflow service changes can be triggered by environmental changes such as user mobility [4], devices availability or other factors. For example, WSD defines *execution workflow* at time t by selecting $tv_1 \rightarrow_1$ then at $t + \Delta$ changes s_1 to s_2 to maintain user QoE, as shown in Figure 3.

2.3 Dynamic Access Control Policy (DACP)

The DACP generates ACL policy rules to support workflow communications. We consider WSD as a blackbox that can notify DACP about network changes on the workflows. Therefore, DACP can generate and updates ACL rules dynamically as a react for workflows changes or prior to expected changes by WSD. Reactive approach DACP generates and installs ACL policy as a reaction for changing in a workflows. The reactive approach is well known in access control research and usually used with IDS [12]. The notifications contain information regarding the changes in the workflow such as *(FROM, TO, SERVICE, REMOVE/ADD)*. For the aforementioned example, upon changing W_1, WSD sends *(tv_1, s_1, stream, REMOVE), (tv_1, s_2, stream, ADD)*. Then access control policy generator will generate the corresponding rules of W_1 to block tv_1 from accessing s_1 and allow it to access s_1.

On the other hand, proactive approach generates required access rules for the expected changing in workflows. It requires early notification from the WSD about any expected changing in a workflow connections. WSD sends notifications as it does in the reactive approach. However, this time is before the change take place (i.e. before handover). Once the notifications received, corresponding rules will be added for new connections, however, old connections will not be removed until handover is done. The proactive approach will handle the aforementioned example as following; first, it receives WSD early notification about workflow changes and install rules for *ADD* connections only (e.g.$tv_1 \rightarrow s_2$). Once the handover complete WSD sends another notification about that, and accordingly, the workflow connections *REMOVE* (e.g.$tv_1 \rightarrow s_1$) will be removed.

ACL rules are consist of a) *match*: represented by four-tuple *(ip_src, ip_dest, port_dest, tp_proto)* and b) allow *action*, and the default is deny. ACL rule's *match* and *action* are filled by DACP with values corresponding to the flow changes. For example, if WSD sends *(tv_1, s_1, stream, REMOVE), (tv_1, s_2, stream, ADD)*, then DACP compiles these changes into rules using network device information Net_{dev} and network service requirements Net_{req} as shown in Algorithm 1. The generated rules can be enforced using Software Defined Network (SDN) as it supports dynamic rule enforcement into switches [12].

3 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed ACAC, a framework supporting dynamic IoT application network access control. ACAC address the problem of enforcing least privilege on IoT applications while supporting dynamic access control policies. We investigate reactive and proactive approaches to dynamically generate access rules for workflows and the effect on the user QoE. We observe the following future research directions: a) how do we reduce handover time while providing an acceptable user experience but not violating the principle of least privilege; b) how to infer workflow changes based upon changes in user behaviour or in the environment.

REFERENCES

[1] Mohammed Al-Shaboti, Chen Aaron, and Welch Ian. 2019. Automatic Device Selection and Access Policy Generation based on User Preference for IoT Activity Workflow. In *eprint arXiv:1904.06495*.

[2] Mohammed Al-Shaboti, Ian Welch, Aaron Chen, and Muhammed Adeel Mahmood. 2018. Towards secure smart home iot: Manufacturer and user network access control framework. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 892–899.

[3] Kyeong-Deok Baek and In-Young Ko. 2017. Spatially cohesive service discovery and dynamic service handover for distributed IoT environments. In *International Conference on Web Engineering*. Springer, 60–78.

[4] Wei Bao, Dong Yuan, Zhengjie Yang, Shen Wang, Wei Li, Bing Bing Zhou, and Albert Y Zomaya. 2017. Follow me fog: toward seamless handover timing schemes in a fog computing environment. *IEEE Communications Magazine* 55, 11 (2017), 72–78.

[5] Paolo Bellavista and A Montanari. 2017. Context awareness for adaptive access control management in IoT environments. *Secur. Priv. Cyber-Phys. Syst.: Found. Princ. Appl* 2, 5 (2017), 157–178.

[6] Suman Sankar Bhunia and Mohan Gurusamy. 2017. Dynamic attack detection and mitigation in IoT using SDN. In *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, 1–6.

[7] Jae-Hyun Cho, Han-Gyu Ko, and In-Young Ko. 2016. Adaptive service selection according to the service density in multiple QoS aspects. *IEEE Transactions on Services Computing* 9, 6 (2016), 883–894.

[8] Microsoft flow team. [n. d.]. Microsoft Flow. https://flow.microsoft.com/en-us/ (visited on 18/10/2018).

[9] Gartner. 2014. Gartner Says a Typical Family Home Could Contain More Than 500 Smart Devices by 2022. https://www.gartner.com/en/newsroom/press-releases/2014-09-08-gartner-says-a-typical-family-home-could-contain-more-than-500-smart-devices-by-2022 (visited on 8/4/2019).

[10] In-Young Ko, Han-Gyu Ko, Angel Jimenez Molina, and Jung-Hyun Kwon. 2016. SoIoT: Toward a user-centric IoT-based service framework. *ACM Transactions on Internet Technology (TOIT)* 16, 2 (2016), 8.

[11] Dimosthenis Kyriazis, Konstantinos Tserpes, Andreas Menychtas, Antonis Litke, and Theodora Varvarigou. 2008. An innovative workflow mapping mechanism for Grids in the frame of Quality of Service. *Future Generation Computer Systems* 24, 6 (2008), 498–511.

[12] Hongda Li, Feng Wei, and Hongxin Hu. 2019. Enabling Dynamic Network Access Control with Anomaly-based IDS and SDN. In *Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*. ACM, 13–16.

[13] Rida Zojaj Naeem, Saman Bashir, Muhammad Faisal Amjad, Haider Abbas, and Hammad Afzal. 2019. Fog computing in internet of things: Practical applications and future directions. *Peer-to-Peer Networking and Applications* (2019), 1–27.