Towards Policy Enforcement Point as a Service (PEPS)

Towards Policy Enforcement Point as a Service (PEPS)

Arash Shaghaghi[1,2], Mohamed Ali (Dali) Kaafar[2], Sandra Scott-Hayward[3], Salil S. Kanhere[1] and Sanjay Jha[1]
[1]School of Computer Science and Engineering, UNSW Australia, Sydney, Australia
[2]Data61, CSIRO, Australia
[3]Centre for Secure Information Technologies (CSIT), Queen's University Belfast, Northern Ireland
Contact: a.shaghaghi@unsw.edu.au

***Abstract*—In this paper, we coin the term Policy Enforcement as a Service (PEPS), which enables the provision of innovative inter-layer and inter-domain Access Control. We leverage the architecture of Software-Defined-Network (SDN) to introduce a common network-level enforcement point, which is made available to a range of access control systems. With our PEPS model, it is possible to have a 'defense in depth' protection model and drop unsuccessful access requests before engaging the data provider (e.g. a database system). Moreover, the current implementation of access control within the 'trusted' perimeter of an organization is no longer a restriction so that the potential for novel, distributed and cooperative security services can be realized. We conduct an analysis of the security requirements and technical challenges for implementing Policy Enforcement as a Service. To illustrate the benefits of our proposal in practice, we include a report on our prototype PEPS-enabled location-based access control.**

I. Introduction

With Software-Defined-Network (SDN), the separation of control and data plane and programmability in the network enable provision of enhanced security systems. A diverse set of proposals have emerged that exploit the architecture of SDN, and specifically the network-wide view of SDN controllers, to implement reactive monitoring and automated response systems. Recently, an emerging body of literature is shaped around the idea of using SDN to introduce innovative security services. We follow the latter approach and leverage the capabilities of SDN in moving towards a new model of access control enforcement, which could potentially open the door to a range of new types of security services.

Access control systems limit the operations of legitimate users [19]. The main components of an access control system include Policy Decision Point (PDP), Policy Repository (PR) and Policy Enforcement Point (PEP). Accordingly, an authorization flow involves retrieving the user access request by PDP, inquiry the PR for matching policies and enforcing the decision by PEP. Figure 1 illustrates a *typical* access control process flow between a Database Management System (DBMS), as the Data Provider (DP), and a user at a remote network, as the Data Requestor (DR). An access request by a DR is sent from the DR network to the DP network, where the DBMS makes the access decisions and enforces them. In other words, with this setup, an access request reaches DR at application-layer and only then is decided about. Hence, an attacker is allowed to engage the system and its hosting network and possibly execute certain types of attacks such as Denial of Service (DoS) or port scanning.

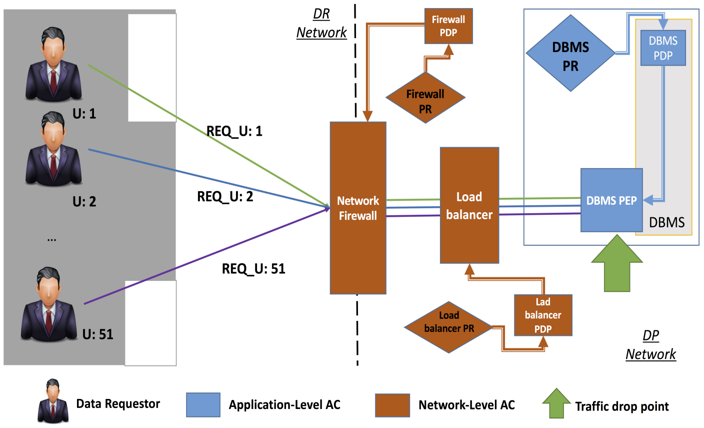


Fig. 1: A typical access control process flow between a Data Provider (DP) and Data Requestor (DR) located in separate networks.

In this paper, we propose to leverage the capabilities brought by SDN to introduce programmable network-level policy enforcement points, which application-layer services may subscribe to. The extra enforcement points serve to create a 'defense in depth' [3, p. 308] model of protection and improve the protection of services hosted in enterprise-like networks. PEPS enables applications such as DBMS to enforce dynamic access control policies both at a lower-level (i.e. network-level enforcement rather than application-level) and closer to the DR's network (i.e. inter-domain enforcement). In effect, PEPS enables authorized system resources to push pre-approved policies to a purpose-built SDN application, which enforces these policies at the level of SDN switches. We coin Policy Enforcement Point as a Service (PEPS) for this model of enforcement.

Referring to Figure 1, with PEPS, instead of waiting for the requests to reach the DBMS's PEP, the DBMS may instruct the network to drop requests originated from a specific network address for a certain period. Similarly, for Quality of Service (QoS) purposes the DBMS may instruct the firewall to adjust traffic volume forwarded to it. Moreover, if the two network-domains were to collaborate, the DBMS may push dynamic and pre-approved policies to the DR's network and block unauthorized access requests either pro-actively or reactively. For example, access requests from 'non-secure' areas of a building destined to the DBMS may be dropped as early as entering the DR's network. We remind that in defense in depth model of protection, the outer-layer defenses may be less reliable than the inner-layers. Hence, if, for any reason, the DR's network fails to ensure to the remote policies, the standard DR's PEP is still in effect.

The resulting protection with PEPS is significantly different and novel compared to status-quo. In fact, from an access control viewpoint, the extra enforcement points at SDN's data plane, facilitates moving towards distributed and cooperative enforcement of access control for application and services. PEPS also motivates a new line of thought in access control, which is deploying verifiable protection points beyond the trusted perimeter of an organization.

The rest of this paper is structured as follows. In Section §II we briefly revise background information on Access Control and SDN security. Thereafter, in §III, we elaborate on our motivation and preliminary technical requirements for implementing PEPS. In §IV, we report on our prototype implementation of a PEPS-enabled location-based access control (LBAC) system. The advantages of our LBAC compared to state-of-the-art is discussed to motivate further investigation of various applications of PEPS. We conclude this paper specifying our work-in-progress and outlining suggestions for future work.

II. Background

A. Access Control

Every user's attempt to interact with protected resources is mediated by access control - the oldest information security mechanisms. During the last decade, an increasing number of major data leakage incidents are associated with the failure of access control [8]. Security researchers [17], [25], [27], associate this to the incompatibility of currently implementable access control with today's requirements. Hence, an increasing number of researchers are investigating innovative proposals to change this condition [9]. One of the promising directions is the interaction of access control with other security services. For example, Crampton et al. propose integrating intrusion detection systems with access control systems [6].

Distributed access control is a fairly recent trend in access control. For example, in [24], authors propose having multiple principals defining the policies for PDP. Nevertheless, the enforcement is through a single trusted reference monitor. Digital Rights Management (DRM) [22] is another example, which is constituted of distributed enforcement. With DRM, the client-side enforcement is, in fact, an extra point of enforcement that facilitates a more granular control over information. DRM is well-recongized and appreciated by industry, and its architecture has been inspiring for our work.

B. Software-Defined-Network Security

SDN Security literature may be split into two main categories, securing the Software-Defined-Network itself or leveraging the capabilities of this technology for security services. In [20], Scott-Hayward et al. provide a categorization of the security issues associated with the SDN framework, and detail the body of literature focussed on solutions to these threats. The security requirements of PEPS defined in §3.3 rely on such solutions.

On the other hand, SDN facilitates the provision of reactive and automated monitoring, analysis and response systems. The key SDN characteristics contributing here are the network-wide view for centralized monitoring [2] and the programmability of SDN to redirect selected network traffic through middleboxes (see [4], [10], and [18] for examples). Along with the improvement of traditional security solutions via SDN, novel security services are also built on top of SDN. For example, [11] uses SDN to develop an architecture that enables residential internet customization, which could be used to secure household appliances. [15] and [21] also introduce innovative services.

Recently, a few number of solutions extend the Authentication, Authorization, and Accounting (AAA) functionality using the SDN controller and focus on identity management and authentication mechanisms (e.g. [14] and [7], [23]). Our PEPS model is a network-level access control implementation deployed at the SDN data plane.

III. Policy Enforcement as a Service

A. Motivation

Every organization has a number of systems equipped with their own access control mechanism, e.g. file systems, firewalls, location-detection, etc. The access control component of these systems operates independently. Hence, if any of these PEP fail then unauthorized access to data is inevitable. As mentioned in II-A, distributed reference monitors have been previously investigated in the literature. However, to the best of our knowledge, the idea of having a cooperation among PEP has not yet been explored. Recalling that in most cases access requests to data, or resources, are mediated through the network we believe it is possible to place a shared enforcement point for all services to use. However, unlike firewalls, this component has to adhere to dynamic policies and requirements of application-layer systems.

Moreover, letting applications such as DBMS instruct the network may result in better and more dynamic network management. For example, assume at time t of day d the network infrastructure hosting the DBMS is congested and can only handle 50 concurrent connections to DBMS due to the global QoS requirements. Accordingly, the DBMS administrator defines a policy to drop connection requests beyond 50 and instructs the DBMS'PEP to limit the total number of requests from a single source to 10. The issue with this arrangement is that the UNSW network Admin has to trust the DB Admin and the DBMS access control for this as such temporary policies are application-dependent and are unknown to the network components such as a firewall. Furthermore, with application-level access control traffic still reaches the network and attacks such as DoS may still target the network hosting the DBMS.

Thirdly, dropping traffic associated with unauthorized requests closer to the source would enable saving significant traffic from flowing over the networks or Internet.

B. Proposed Approach

We propose designing a shareable enforcement point at network-level, which is made available to application-layer access control systems. The shareable enforcement point is made available as a service and application or services need to subscribe to use it. We coin the term ‘Policy Enforcement as a Service’, or PEPS, for this security service.

Relying on traditional networks and deploying middle-boxes for PEPS would be challenging. Specifically, policy conflict resolution and performance management will be inefficient and troublesome. However, the SDN architecture is well-suited for such requirements since the controller composes policies received by various applications and there is an on-going effort to optimize this process with respect to dynamic and reactive policies.

In SDN, the control plane entails both PR and PDP and the data plane is equivalent to PEP in access control. In essence, the SDN controller takes as input an extra set of policy for PEPS, which may be defined by local or remote application-layer access control systems. We design an SDN application responsible to retrieve these policies and submitting them to the network operating system.

C. Assumptions

We require the following assumptions to hold:

- The SDN controller and external SDN applications are assumed to be secure and able to communicate securely (e.g. using TLS).
- The SDN data plane is not compromised.
- The east and west bound communication link between controllers in different networking domains is secure.

As mentioned in II-B, there is an over-expanding body of literature exploring the security of SDN both at data plane and control plane. Similar to various proposals that leverage SDN to introduce novel services and applications (see §II-B), we focus on our proposed system assuming the underlying platform is reasonably reliable and secure.

D. Security Requirements

A PEPS solution should be designed and implemented such that a malicious subscriber, whether in the same perimeter or not, cannot:

- Violate the policy specifications of the service provider through the remote policies.
- Violate the policy specifications of other services, which use the enforcement point, whether in the same perimeter or not.
- Affect the performance of the SDN controller itself. For example, causing a DoS attack with constant update of the remote policies.

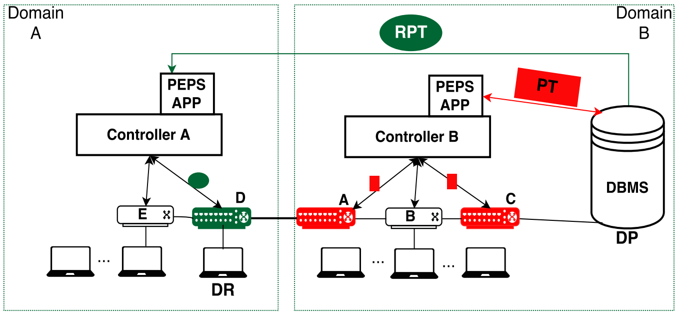


Fig. 2: Abstract representation of Policy Transfer (PT) and Remote Policy Transfer (RPT) in SDN networks deploying PEPS. Switches in red and gree colour are effected by PT and RPT, respectively.

E. Main Components and Requirements

Figure 2 shows the main components required in an SDN network deploying PEPS. **Policy Transfer** is the standard protocol used to define policies at application-layer (e.g. by DBMS) for network-level SDN application. Similarly, **Remote Policy Transfer** is used to translate application-layer policies for a remotely located SDN network deploying PEPS. RPT is securely exchanged over east and westbound link between controllers and PT is exchanged over a secure connection.

Conflicting policies will result in one or more of the threats mentioned in III-D. Therefore, we have to ensure the following three requirements are met:

Requirement 1: Let P' be the set of policies for controller C_1, which is in domain D_1 and governs over the set of switches S. We define P_r as the PT for C_1 and say: P_i is a valid PT for P' if and only if $P = \{P' \cup P_r\}$ does not violate the original policy specification P'.

Requirement 2: ensures the remote policies do not conflict with original policy specification. Therefore, we just replace PT with RPT in Requirement 1.

Policy composition and conflict detection is an ongoing challenge in Software-Defined-Network [12]. In order to prevent adding further complications to this domain with PEPS, it is best to restrict the capabilities of RPT at this time. We postulate to restrict a PEPS service subscriber only to submit RPT that relate to flow destined directly towards it (e.g. DB in Domain B may only set RPT at domain A for traffic flowing towards it’s own domain). Moreover, the priority of rules set after conversion of RPT should always be set below any matching policy set locally. Accordingly, we define Requirement 3:

Requirement 3: Let P' be the set of policies for controller C_1, which is in domain D_1 and governs over the set of switches S and *has been defined locally*. We define P_r as the remote policy for C_1, which is generated according to RPT. Then, having $\exists P_{r_i}$ that $\perp P'_i$ results in P'_i OVERRIDES P_{r_i} in the final policy set $P = \{P' \cup P_r\}$.

F. Practical Considerations

Multi-Table Pipeline: the data plane of SDN supports Flow Table Pipeline (FTP) - introduced with OpenFlow specification

V1.1 to improve the flow processing performance [1]. The pipeline consists of multiple flow tables. The incoming packet is first matched with the first flow table, where the specified actions could direct the packet to another flow table for further processing of the packet. With this redirection mechanism, the SDN control plane could build a logical single source directed acyclic graph on the FTP for processing.

To implement non-conflicting remote policies we propose customized use of FTP. All flow rules resulting from PT or RPT should be added to the last flow table. This flow table is directly managed by our purpose built PEPS APP. The incoming flow to the switch is first-matched against all but the last flow table (i.e. rules required by local policies are first processed), and if a flow is still allowed, then it is passed to the final flow table for processing. In other words,

Let FTP be a set of flow tables $\{FT_1, FT_2, ..., FT_n\}$, FT_i for $i < n$ generated according to the set of policies P' for Controller C_1, FT_n set according to remote policy P_r for C_1. Then, an incoming packet $Pckt$ is MATCHED against FT_i for $i < n-1$. The resulting $Pckt'$ is then MATCHED against FT_n.

This simplifies conflict resolution between local and remote policies when using FTP.

Multiple PEPS SDN Application Instances: PEPS APP is installed on networks deploying PEPS model of enforcement. This application is responsible to retrieve PT and RPT and to convert them into flow-table rules for submission to the controller. PEPS should be securely connected to application-layer services sending PT or RPT. Moreover, we must ensure PEPS has minimum impact on the controller performance. Network-Function-Virtualization (NFV) may be used to improve the PEPS performance.

IV. PEPS in Practice

We now report on our prototype implementation of a PEPS-enabled location-based access control. This section aims to highlight the advantages of PEPS in practice and motivate future work.

Location-based access controls rely on user's location as one of the attributes when making access decisions. There are simple solutions to retrieve user's location. For example, it is possible to retrieve user's location using the device integrated peripherals such as GPS device. However, proof of presence is a challenging aspect of location-based services, especially for an indoor environment. As thoroughly discussed in [16], proof of presence schemes can be categorized into beaconing-based, context-based and distance-bounding based approaches. Most of the proof of presence solutions are challenged for one or more of the following reasons: requiring specialized hardware or software, being immobile, unable to track movement in real-time (or requiring extensive ongoing context scans either by Data Provider or Data Requestor), being computationally hard or infeasible, or being extremely privacy-invasive. Hence, in practice, the adoption of these schemes by organizations is challenging (e.g. [13], [26]).

Here, we propose and implement two alternative approaches to ensure proof of presence and enforce location-based access control using PEPS model. These schemes are not originally built to replace existing solutions. Instead, we are interested to use them as the first layer of defense (i.e. the outer layer of defense in depth model). We define a scenario in which there are two organizations both with SDN networks. The Data Provider (DP) resides in network B, and the Data Requestor (DR) is located in network A. We have implemented the following scheme within a simulated environment using Mininet 2.2.0 and Floodlight V.1 running as the SDN controllers. The applications have been developed for this controller and communicate over a secure TLS connection with an open source database server, MariaDB, as the Data Provider. We have integrated an extra module into MariaDB, which mediates communication and coordinates with SDN PEPS APP both in the local and remote networks.

PEPS-enabled location-based access control with real-time location tracking

SDN-based location tracking: we use OpenFlow to retrieve the location of users in real-time. This is a new approach to track users and can be easily deployed without any specialized hardware in SDN networks. Whenever a packet is received by a switch, and it does not match any of its existing forwarding rules then a *packet_in* message containing the *switch ID* and *port ID* is sent to the governing controller. The controller uses this information to create a dynamic geo-location lookup table. This table matches the user's device IP to a switch port. The network locations retrieved through *switch ID* can be matched to different sections within the building. For example, in Figure 3, *Location 1* is associated to *AP 1*. An issue to consider for wireless devices would be managing the signal coverage that could mislead this scheme. This can be solved using proper and careful positioning of these devices and signal blocking solutions [5]. Indeed, the cost of performing such is much lower than having specialized equipment for location detection. Moreover, an important advantage of this scheme is that unlike most proof of presence schemes, it is capable of tracking the movement of the user around the locations in real-time. It is possible to ensure that this scheme is secure against IP Spoofing by setting a rule that only packets from a specific IP address are forwarded from the switch port.

PEPS-based Access Enforcement: at this point, using the above scheme, we build a location-based access control model on top of our PEPS model. As depicted Figure 3, we require an *SDN-Location App* (equivalent to PEPS APP referred to earlier) installed on both DP and DR networks. An RPT, issued by the DP, defines that any traffic destined to DP is dropped unless the SDN-Location APP on the requesting side initiates a valid session with the same application on the provider side. A valid session requires that the user requesting data be located by the *SDN-Location App* and is allowed to communicate with DP in accordance with the rules extracted from RPT. Only then a host is allowed to send a request for data. As also depicted in Figure 3, compared to existing approached,

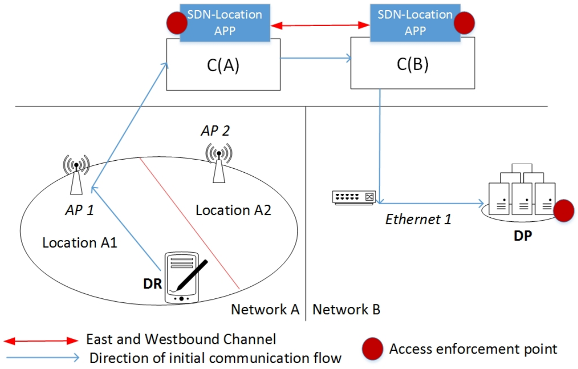


Fig. 3: Policy enforcement points that exist with PEPS are depicted within a simplified location-based access control. Without PEPS, the only PEP would be at DP.

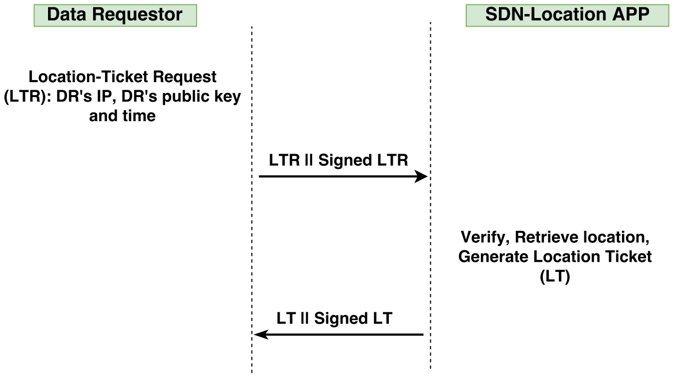

Data Requestor
SDN-Location APP
Location-Ticket Request (LTR): DR's IP, DR's public key and time
LTR II Signed LTR
Verify, Retrieve location, Generate Location Ticket (LT)
LT II Signed LT

Fig. 4: Representation of proposed ticketing protocol.

with our location-based access control model there are extra network-level enforcement points both at source network and host.

PEPS-enabled location-based access control with location-tickets

The SDN-based Location Ticketing Scheme: it is possible to use the same location detection scheme to generate location tickets - rather than real-time tracking. The assumptions and requirements for the location-ticket scheme is depicted in Figure 5. Each controller and user are equipped with a public and private key. The DR creates a Location Ticket Request *LTR* containing the DR's IP address, public key and time. It digitally signs *LTR* and sends it to the *SDN-Location App* running on top of the controller. The signature is verified, and the IP address is compared with the one in the packet header. If the IP is legitimate, the user's location is retrieved using the same approach mechanism described earlier. A Location Ticket (LT) is then generated using the DR's IP address, its public key, time and location. LT is signed and sent along with LT to the DR. The protocol is represented in Figure 4.

The proposed location ticket scheme binds the DR's IP and public key together. This helps to prevent one of the main threats against proof of presence schemes such as Sybil Attack, where users create several fake identities in several locations within the network.

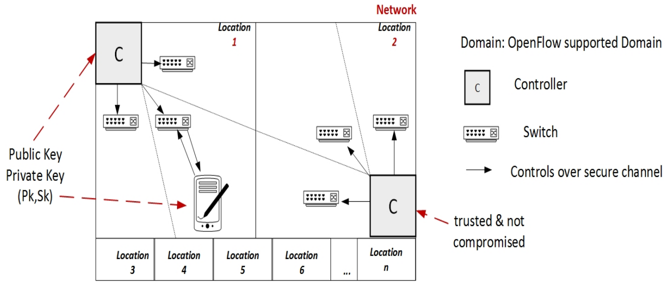


Fig. 5: Assumptions and requirements for the location-ticket (LT) scheme.

PEPS-based Access Enforcement: the location-ticket scheme facilitates the integration of PEPS with existing application and services. Specifically, unlike the real-time approach, there is no requirement of having SDN APP on both DP and DR. A location ticket issued by SDN APP at DR may be provided to any application or service requesting proof of presence. The LT scheme also removes the requirement of session establishment between remote controllers, which may be more practical in many scenarios. We implemented the LT scheme and sent location tickets along with access requests to MariaDB as part of our prototype implementation.

A. Security and Performance Analysis

Performance Analysis: we simulated a network with 32 switches and four threads and sent location ticket requests to the application running on top of the controller. Figure 5.a shows the standard performance of the Floodlight controller when not running the *SDN-Location App*. We then ran the application and issued 1000 LTR. The controller performance was steady and cumulative distribution function (CFD) showed reasonable performance impact. However, as we increased the LTR numbers the performance of the controller when handling incoming flows degraded — compare Figure 5.b with 5.a. This points us to the fact that it may be a better approach to outsource demanding processes and use solutions such as NFV.

Security Analysis: we include an analysis of SDN-based location detection scheme. The security and performance of PEPS is included in Section 5.

The scheme does not rely on user's device peripherals and is built on capabilities available at network infrastructure level. Hence, it is much harder for an attacker to compromise the system. Also, since this scheme does not rely on context measurement information, it is secure against most recent attacks including Context Guessing Attack [16]. Moreover, this scheme could be used as a standalone solution — not for proof of presence but actual location detection. If so, it allows the protection of user's privacy against service providers that retrieve a huge amount of personal information when retrieving the device location. However, the original scheme is vulnerable to the Wormhole attack. It is possible to solve this problem using authenticated Ping and various other network delay measurement techniques. As further security analysis

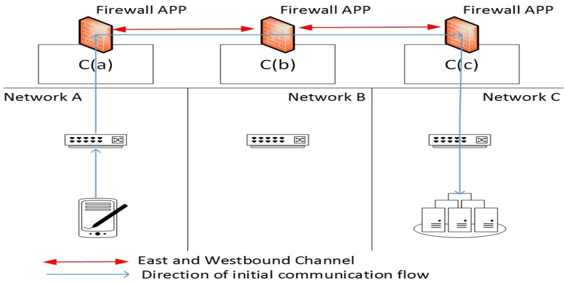


Fig. 6: Abstract representation of progressive layered firewall model.

and improvement is beyond the scope of this paper and we leave this for our future work.

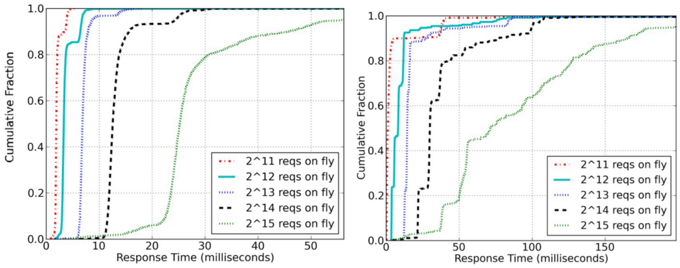
V. Discussion

As illustrated in Figure 3, our PEPS-based access control model allows having a defense in depth model of protection. This change in access control enforcement has several advantages. For example, it allows network bandwidth to be saved by blocking unauthroized requests at the source. It also enables s of certain categories of attacks, where the attack is based on challenge and response (e.g Port Scanning). Evidently, dropping traffic before engaging services or systems also facilitates protection against DoS threats.

PEPS enables having a more context-aware access control. For example, if the remote enforcement is not blocking traffic as expected then it could be considered as less trustworthy. Accordingly, if controllers in different domains were to share knowledge about this, they could block all, or specific, access requests originated from the suspicious network until further investigation (e.g. the controller may be compromised or the PEPS APP may be malfunctioning).

We presume the aforementioned are only some of the advantages of brought with a PEPS model of access control enforcement. Specifically, the co-operation of domains in access control could lead developing novel security services never sought before. For example, we are investigating the development of a PEPS-enabled inter-domain firewall system, which gradually and progressively applies policies (see Figure 6 for an abstract representation). In other words, the RPT mechanism used to define non-conflicting remote policies could be used between firewall applications of SDN controllers to progressively block unwanted traffic reaching an organization network. It should be noted that, from a practical point of view, such approach may not have been feasible with existing firewall solutions without SDN and conceptualization of PEPS. For example, firewalls may have been from different providers and cooperation would not have been feasible. We leave further investigation and exploration as future work.

PEPS is currently at its conception phase and requires much further exploration and development before coming into practice. Specifically, the translation of PT and RPT for the network hosting PEPS is a challenging issue – e.g. which forwarding devices will have to apply the remote policies in the network. Moreover, the impacts of PEPS on network performance and security threats associated with it require proper analysis. We remind that our early performance evaluation is not prohibitive (see §IV).



(a)

(b)

Fig. 7: Impact of SDN-Location APP on Floodlight controller. Figure (a) is without the application running and Figure (b) is with the application running.

VI. Conclusion and Future Work

In this paper, we revisited the Policy Enforcement Point (PEP) of access control. We introduced Policy Enforcement Point as a Service, or PEPS, by leveraging the capabilities of Software-Defined-Network (SDN). PEPS allows cooperation of PEP among application-layer and network-layer services either in the same network or remote domains. It enables improving the security of application-layer services hosted in networks and promises the development of innovative collaborative network-based security services. Beyond conceptualization, we made an early attempt to discuss practical requirements for PEPS and reported on our prototype implementation. Detailed analysis of some of the security challenges of PEPS and a more technical exploration on how to integrate remote policy is left as our future work.

References

[1] OpenFlow Switch Specification Version 1.1. Open Networking Foundation.

[2] I. Alsmadi and D. Xu. Security of software defined networks: A survey. *Computers & Security*, 53:79–108, 2015.

[3] R. Anderson. *Security engineering*. John Wiley & Sons, 2008.

[4] B. Anwer, T. Benson, N. Feamster, D. Levin, and J. Rexford. A slick control plane for network middleboxes. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 147–148. ACM, 2013.

[5] D. D. Coleman, D. A. Westcott, B. E. Harkins, and S. M. Jackman. Certified wireless security professional official study guide, 2010.

[6] J. Crampton and M. Huth. Towards an access-control framework for countering insider threats. In *Insider Threats in Cyber Security*, pages 173–195. Springer, 2010.

[7] V. Dangovas and F. Kuliesius. Sdn-driven authentication and access control system. In *The International Conference on Digital Information, Networking, and Wireless Communications (DINWC2014)*, pages 20–23. The Society of Digital Information and Wireless Communication, 2014.

[8] S. C. David M. Upton. The danger from within. *Harvard Business Review*, 2014.

[9] Y. Desmedt and A. Shaghaghi. Function-Based Access Control (FBAC): From Access Control Matrix to Access Control Tensor. In *Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats co-located with ACM CCS 2016*. ACM, 2016.

[10] S. K. Fayazbakhsh, V. Sekar, M. Yu, and J. C. Mogul. Flowtags: Enforcing network-wide policies in the presence of dynamic middlebox actions. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 19–24. ACM, 2013.

[11] H. H. Gharakheili, L. Exton, V. Sivaraman, J. Matthews, and C. Russell. Third-party customization of residential internet sharing using sdn. *International Telecommunication Networks and Applications Conference (ITNAC)*, 2015.

[12] D. Kreutz, F. M. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig. Software-defined networking: A comprehensive survey. *proceedings of the IEEE*, 103(1):14–76, 2015.

[13] W. Luo and U. Hengartner. Proving your location without giving up your privacy. In *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, pages 7–12. ACM, 2010.

[14] D. M. F. Mattos, L. H. G. Ferraz, and O. C. M. B. Duarte. Authflow: Authentication and access control mechanism for software defined networking.

[15] S. A. Mehdi, J. Khalid, and S. A. Khayam. Revisiting traffic anomaly detection using software defined networking. In *Recent Advances in Intrusion Detection*, pages 161–180. Springer, 2011.

[16] M. Miettinen, N. Asokan, F. Koushanfar, T. D. Nguyen, J. Rios, A.-R. Sadeghi, M. Sobhani, and S. Yellapantula. I know where you are: Proofs of presence resilient to malicious provers. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pages 567–577. ACM, 2015.

[17] J. Park and R. Sandhu. The ucon abc usage control model. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):128–174, 2004.

[18] Z. A. Qazi, C.-C. Tu, L. Chiang, R. Miao, V. Sekar, and M. Yu. Simple-fying middlebox policy enforcement using sdn. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 27–38. ACM, 2013.

[19] R. S. Sandhu and P. Samarati. Access control: principle and practice. *Communications Magazine, IEEE*, 32(9):40–48, 1994.

[20] S. Scott-Hayward, S. Natarajan, and S. Sezer. A survey of security in software defined networks. *Communications Surveys Tutorials, IEEE*, PP(99):1–1, 2015.

[21] S. Shin and G. Gu. Cloudwatcher: Network security monitoring using openflow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?). In *Network Protocols (ICNP), 2012 20th IEEE International Conference on*, pages 1–6. IEEE, 2012.

[22] S. Subramanya and B. K. Yi. Digital rights management. *Potentials, IEEE*, 25(2):31–34, 2006.

[23] U. Toseef, A. Zaalouk, T. Rothe, M. Broadbent, and K. Pentikousis. C-bas: Certificate-based aaa for sdn experimental facilities. In *Software Defined Networks (EWSDN), 2014 Third European Workshop on*, pages 91–96. IEEE, 2014.

[24] P. Tsankov, S. Marinovic, M. T. Dashti, and D. Basin. *Decentralized composite access control*. Springer, 2014.

[25] Ulfar Erlingsson, Keynote. *Advances in Cryptology – ASIACRYPT 2011: 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011, Proceedings*. 2011.

[26] N.-C. Wu, M. Nystrom, T.-R. Lin, and H.-C. Yu. Challenges to global rfid adoption. *Technovation*, 26(12):1317–1323, 2006.

[27] Yvo Desmedt, Keynote. *Security and Privacy in Communication Networks: 7th International ICST Conference, SecureComm 2011, London, September 7-9, 2011*.