Service Dependencies-Aware Policy Enforcement Framework Based on Hierarchical Colored Petri Net

Yosra Ben Mustapha and Hervé Debar

Telecom Sudparis, SAMOVAR UMR 5157
9 rue Charles Fourier, 91011 EVRY, France
{yosra.ben_mustapha,herve.debar}@telecom-sudparis.eu

Abstract. As computer and network security threats become more sophisticated and the number of service dependencies is increasing, optimal response decision is becoming a challenging task for security administrators. They should deploy and implement proper network security policy enforcement mechanisms in order to apply the appropriate countermeasures and defense strategy.

In this paper, we propose a novel modeling framework which considers the service dependencies while identifying and selecting the appropriate Policy Enforcement Points during an intrusion response process. First, we present the security implications of the service dependencies that have been developed in the literature. Second, we give an overview of Colored Petri Nets (CPN) and Hierarchical CPN (HCPN) and its application on network security. Third, we specify our Service Dependencies-aware Policy Enforcement Framework which is based on the application of HCPN. Finally and to illustrate the advantage of our approach, we present a webmail application use case with the integration of different Policy Enforcement Points.

1 Introduction

As computer and network security threats are becoming more sophisticated as described in [1] and the number of service dependencies is increasing, optimal response decision is becoming a challenging task for security administrators. Applying appropriate countermeasures and defense strategy implies the deployment of proper network security policy enforcement while taking into account service dependencies and their different interactions. Most of the current automated Intrusion Response Systems (IRS)s are based on the risk assessment and the cost-sensitive analysis as detailed in [2–4]. They are still suffering from several drawbacks as described in [5]. Usually, they provide isolated response applied in a single Policy Enforcement Point (PEP) of the Information System.

The lack of formal representation of the interaction between service dependencies and policy enforcement mechanisms is a motivation for our proposed approach. Moreover, there is still a gap between service dependencies, attack graphs and policy enforcement reaction decisions. Thus, our main objective is

Sabu M. Thampi et al. (Eds.): SSCC 2013, CCIS 377, pp. 313–321, 2013.

to design and develop new strategies to optimize policy enforcement response decision against single alert first, then, multiple alerts.

We propose to extend the existing service dependency model by including a clear and explicit representation of policy enforcement mechanisms (Firewalls, User Directories, ...). We consider the service architecture and its dependencies in order to explore several enforcement possibilities in a response decision. Contrary to the majority of research which consider the security as a service, we distinguish between a service component and a security component. By doing so, we aim at giving a clear representation of PEPs in the service dependencies model. In fact, in the existing Service Dependency Models, as described in [6,7], the interaction between two dependent services is only constrained by the presence of required privileges. This hides the concrete and explicit presence of PEPs in the dependencies model.

Therefore, we propose to deploy Hierarchical Colored Petri Nets (HCPN) by introducing the notion of substitution transitions to represent PEPs. Each PEP is characterized by its enforcement capabilities. We model these capabilities by specifying the substitution transition functions. In this paper, we propose a novel modelling framework which considers service dependencies while selecting appropriate PEPs in an intrusion response process. First, we present the security implications of the service dependencies that have been developed in the literature. We also detail the definition of Policy Enforcement and Access Control. Second, we give an overview of deploying Colored Petri Nets (CPN) as well as Hierarchical CPN (HCPN) and motivations to apply this latter in network security issues. Third, we specify our Service Dependencies-aware Policy Enforcement Framework which is based on the application of the HCPN. Last but not least in order to illustrate the advantage of our approach, we present a webmail application use case with the integration of different Policy Enforcement Points.

2 Policy Enforcement Point Definition

According to [8], *the Policy Decision Points (PDPs) process Access Control policies, along with other data such as network state information, and take policy decisions regarding what policies should be enforced and how this will happen. These policies are sent as configuration data to the appropriate Policy Enforcement Points (PEPs), which reside on the managed devices and are responsible for installing and enforcing them.*

The concept of PEP was also introduced in [9] as an entity that performs access control by making decisions requests and *enforcing* authorization decisions by the PDP. Referring to the latest version of [9], the PEP comes in many forms. It can be part of a remote-access gateway, part of a web server or email user-agent, etc. In [10], the PEP is defined as the most security critical component, which protects the resources and enforces the PDP's decision. Generally, the PDP and the PEP are combined to control access and enforce the security policy.

In [11], authors specifies that the logical level integration allows the enforcement of policies through a PEP. Usually, in applications where multiple archi-

tecture is layered, the PEP is located between the presentation layer and the logical layer, intercepting the calls, using the controls, and in case the invocation is authorized, it can be executed.

3 Service Dependencies Model and Security Implications

Recently, the application of service dependencies in the response decision process is gaining an expanding interest, [6,7,12,13]. As mentioned in [6], the high number of service dependencies increases the challenge of response decision and policy enforcement. Therefore, authors propose to use service dependencies as frames for attack impact and response costs evaluation.

In [12], authors present in depth explanation about the use of service dependency to enforce policy-based response. They propose to find the best suitable set of PEPs capable of applying a response that has been already selected. Authors define the PEP capability as the ability of the PEP to apply a security rule on a specific set of subjects, actions and objects.

In [7], authors demonstrate how service dependencies have security implications and extend the work presented in [12]. First, they consider that a service is a set of components C_i. A simple service model is composed of connected components. The basic model M_S of the service S is defined by $M_S = \{(C_i, PC_i, SC_i)\}, i \in \{1 \dots n\}\}$. PC_i and SC_i are respectively the providers of resources to C_i and the subscribers of resources provided by C_i. This model is extended by adding (1) the privileges granted to the service Pr_C, (2) the credentials accredited to this service Cr_C and (3) the trust it has regarding other privileges and/or credentials Tr_C. This modelling approach links dependent services by defining these sets: providers and subscribers and required privileges of each service.

The proposed model has been used in order to assist the administrators in selecting the appropriate Policy Enforcement Point (PEP). Authors does not formally define this latter -PEP- and consider each service as a PEP having limited access control capabilities. But, this approach does not allow us to dynamically and automatically identify PEPs capable to implement a proper response decision.

In [13], authors combines service dependency graphs with attack graphs in order to enable security analysts to make more informed decisions. According to them, there is a gap between system components and service dependencies graphs and attack graphs. Therefore, they propose to combine attack and dependency graphs to provide security analysts with a better picture of the cyber situation.

4 Colored Petri Nets: Definitions and Related Works

4.1 Brief Definition of Petri Nets

Petri Nets formalism was introduced by C. A. Petri in 1962, [14] for the mathematical representation of discrete distributed systems. Known also place/transition nets, it graphically depicts the structure of systems as a direct bipartite

graph with annotations. Places and transitions are respectively denoted by circles and rectangles. Directed edges (arcs) connect places to transitions and vice versa. A transition can be fired only if the input place(s) have the required number of tokens.

4.2 Colored Petri Nets: CPNs and Hierarchical CPNs

Colored Petri Nets (CPNs) are an extension of original Petri Nets. Contrary to ordinary Petri Net which do not define types for tokens and modules, using CPNs, we are able to assign tokens with different data value called token colors. Moreover, connections between places and transitions are labeled by a color function. These CPN additional features allows us to model more complex and large systems. However, both Petri Nets and CPNs do not ensure modularity modelling of such systems with multi level and heterogeneous activities. Hierarchical CPNs have been proposed to answer to such modelling requirements. HCPNs make the modelling of distributed systems and different level of abstraction feasible. Similar to modular programming, HCPNs is composed of smaller CPNs (called subpage) that are represented in the upper net (called superpage) by *substitution transitions*. Due to space limitation, we invite the reader to refer to reference [15] and [16] to get more details about the CPNs and HCPNs formalism.

CPNs and HCPNs are used in a wide variety of application domains [17], for instance distributed systems [18], information systems, communication protocols, security protocols [19–21], data networks, security policy enforcement mechanisms [22], attack modeling [23], etc.

In [24], HCPNs verify and analyze Web Service Composition. CPNs have been also applied to model service dependencies as described in [25]. Authors model service privileges by CPN tokens. A dependency between two services will be

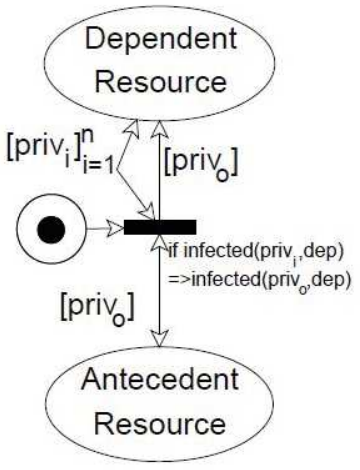

Fig. 1. CPN dependency [25]

then satisfied if and only if all the required privileges exist in the input places (the dependant service), see figure 1.

The resulting model represents both services and PEP without distinction; since authors confound service and PEP. This modeling approach has some drawbacks regarding the representation of the service and the policy enforcement capabilities of the monitored systems.

5 Service Dependency-Aware Policy Enforcement Framework

In this section, we first define the concepts that are deployed in the proposed framework. Then, we describe the main components of our proposed framework.

5.1 Definitions

Service. A service is an implementation of an interface which provides resources to its users. We define a service as a tuple composed of the following attributes:

- Name: the name identifying the service
- Service-ID: service identifier.
- Description: brief functional description of the service
- A-SD (Service Dependency) List: a list of the antecedent service(s) on which the service depends
- D-SD List: a list of the dependent service(s) which depend on the service

Attack and Threat Context. An attack is a potential malicious event detected by an IDS. Usually attacks are only discernible in terms of IDS alerts. In addition, this definition of attack makes it interchangeable with the IDS alert. Thus, we will not always explicitly state that an attack is represented by the alerts. We also assume that Intrusion Detection Systems and Alert Correlation techniques provide us with clear identification of the attack.

Each attack is characterized by the following attributes respecting the Intrusion Detection Message Exchange Format IDMEF [26]:

- CreateTime: The time when the attack is detected.
- Source: The source attribute includes information about the attacker. It can be a tuple of { IP address, Port, protocol }
- Target: The target includes information about the exploited entity. It can be a tuple of { IP address, Port, protocol, service, User }
- Classification: The attack type. Usually, it refers to the exploited vulnerability.
- Assessment: impact assessment, action, confidence.

Threat context are used referring to these alert attributes and respecting the Or-BAC formalism as described in [26]. For each detected attack, a new threat context is activated. Our proposed framework performs while considering this context.

Policy Enforcement Point. We define each PEP by the following attributes:

- Role Set (RS): The set of subjects that can be controlled and protected by the PEP. Referring to this set, we are able to identify if the PEP is capable to counter a detected threat with a specific enforcement action or not.
- Activity Set (AS): The set of activities that the PEP can analyze and mitigate in case of attack.
- View Set (VS): The set of objects that can be protected by the PEP.

We identify these sets referring to the configuration of the PEP. This latter is represented by the set of configured rules which are usually represented as follows:

$$rule : Conditions \rightarrow Decision(s) \tag{1}$$

Decision(s) is an instance from the set of elementary decisions $\{d_1, d_2, d_3, \ldots, d_p\}$ when *Conditions* on Subject, Objects and Actions are satisfied.

5.2 Proposed Framework

Our proposed approach supports reasoning about the global optimality of a chosen set of responses and their application points (PEPs). Global optimality means that a response decision must take into account the fact that there exist dependencies between different services and different PEPs capable of enforcing the same mitigation decision.

In a first stage, we propose to extend the existing service dependency CPN model as described in [7,12] by including a clear representation of policy enforcement mechanisms. In fact, we should be aware of the service architecture and its dependencies in order to explore possibilities to enforce the response decision at different points. Contrary to the approach proposed in [6,7,12,25], we distinguish service components and a PEPs. By doing so, we give an explicit representation of PEPs in the service dependencies model as described in figure 2. In fact, in the existing Service Dependency model, the activation of a transition between two services is constrained by the existence of specific tokens as explained earlier. This hides the concrete and explicit presence of PEP in the service dependencies model. Thus, analyzing potential *attack paths* to a particular service while including the PEPs becomes challenging using such service dependencies framework. Therefore, we propose to deploy HCPNs by introducing the notion of substitution transitions to represent the PEPs. Figure 2 presents an overview of the proposed approach. Each substitution transition is mapped to a PEP capability as defined earlier by defining the ML functions using CPN tool.

6 Illustrative Example: Enforced E-Mail Application

To demonstrate the usability of our proposed framework, we consider the case of a simple network offering three main services: Webmail application, Internet Message Access Protocol (IMAP) and Post Office Protocol (POP). In order

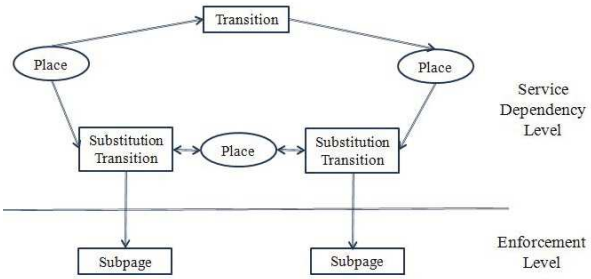

Fig. 2. Overview of the proposed Framework

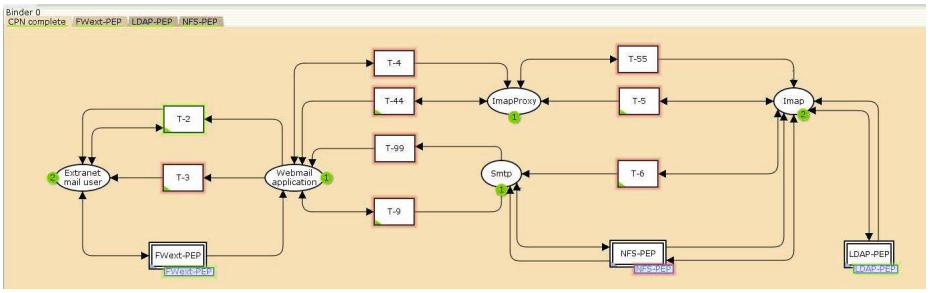

Fig. 3. Use Case: webmail application

to offer the e-mail service to the client, dependencies between webmail application, IMAP and POP services have to be satisfied. This is enforced by the deployment of proper PEPs that are in our case: the LDAP (Lightweight Directory Access Protocol) server, the NFS (Network File System) and the Network Firewall. The firewall monitors the access to the webmail server at the network level. The LDAP controls access to user accounts and the NFS is capable to control the access to the requested files. Each described PEP has its proper access control capabilities that are mapped to appropriate functions in the substitution transitions (labelled boxes). We implement our framework using the CPN Tools simulator, [27]. The use case is illustrated in Figure 3. Each PEP has its corresponding subpage (referenced by labelled boxes: FW-PEP, LDAP-PEP and NFS-PEP) which implement the capability functions. These functions support the security administrator in selecting the appropriate PEP by identifying whether the PEP is capable to mitigate the detected attack.

7 Conclusions and Future Works

In this paper, we propose to integrate Policy Enforcement Point in the service dependencies model in a well presented framework. A significant benefit of

deploying HCPN in our proposed framework is to provide the security administrator with a clear representation of service dependencies and their interactions with Policy Enforcement Points. This support the administrator in detecting which security constraints are violated to access a set of dependent services. The proposed framework is capable to dynamically assist the response process in selecting the Policy Enforcement Points capable of applying dynamic response rules. At present, we are focusing on formally modeling and developing PEP capability functions in order to detect the most appropriate PEPs that are capable to mitigate a detected attacks.

Acknowledgement. The research leading to these results has received funding from the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257495, "Visual Analytic Representation of Large Datasets for Enhancing Network Security (VIS-SENSE)".

References

1. Hachem, N., Mustapha, Y.B., Granadillo, G.G., Debar, H.: Botnets: Lifecycle and Taxonomy. SAR-SSI (2011)
2. Gonzalez Granadillo, G., Débar, H., Jacob, G., Gaber, C., Achemlal, M.: Individual countermeasure selection based on the return on response investment index. In: Kotenko, I., Skormin, V. (eds.) MMM-ACNS 2012. LNCS, vol. 7531, pp. 156–170. Springer, Heidelberg (2012)
3. Stakhanova, N., Basu, S., Wong, J.: A Taxonomy of Intrusion Response Systems. Int. J. Inf. Comput. Secur. (2007)
4. Strasburg, C., Stakhanova, N., Basu, S., Wong, J.S.: Intrusion Response Cost Assessment Methodology. In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security. ACM (2009)
5. Hachem, N., Debar, H., Garcia-Alfaro, J.: HADEGA: A novel MPLS-based mitigation solution to handle network attacks. In: 2012 IEEE 31st International on Performance Computing and Communications Conference, IPCCC (2012)
6. Kheir, N., Debar, H., Cuppens-Boulahia, N., Cuppens, F., Viinikka, J.: Cost evaluation for intrusion response using dependency graphs. In: International Conference on Network and Service Security, N2S 2009 (2009)
7. Debar, H., Kheir, N., Cuppens-Boulahia, N., Cuppens, F.: Service Dependencies in Information Systems Security. In: International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security (2010)
8. Boutaba, R., Polyrakis, A.: Towards Extensible Policy Enforcement Points. In: Sloman, M., Lobo, J., Lupu, E.C. (eds.) POLICY 2001. LNCS, vol. 1995, pp. 247–261. Springer, Heidelberg (2001)
9. eXtensible Access Control Markup Language (XACML) (2003), https://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf
10. Zaborovsky, V., Mulukha, V., Silinenko, E.: Access Control Model and Algebra of Firewall Rules (2011)
11. Betarte, G., Gatto, A., Martinez, R., Zipitria, F.: ACTkit: A Framework for the Definition and Enforcement of Role, Content and Context-based Access Control Policies. IEEE Latin America Transactions (2012)

12. Kheir, N., Debar, H., Cuppens, F., Cuppens-Boulahia, N., Viinikka, J.: A Service Dependency Modelling Framework for Policy-based Response Enforcement. In: Flegel, U., Bruschi, D. (eds.) DIMVA 2009. LNCS, vol. 5587, pp. 176–195. Springer, Heidelberg (2009)
13. Albanese, M., Jajodia, S., Pugliese, A., Subrahmanian, V.S.: Scalable Analysis of Attack Scenarios. In: Atluri, V., Diaz, C. (eds.) ESORICS 2011. LNCS, vol. 6879, pp. 416–433. Springer, Heidelberg (2011)
14. Petri, C.A.: Communication with automata. Ph.D. dissertation, Darmstadt University of Technology (1962)
15. Jensen, K., Kristensen, L.M., Wells, L.: Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. Int. J. Softw. Tools Technol. Transfer, 213–254 (2007)
16. Lakos, C.: From coloured petri nets to object petri nets (1995)
17. Gehlot, V., Nigro, C.: An Introduction to Systems Modeling and Simulation with Colored Petri Nets. In: Proceedings of the, Winter Simulation Conference (2010)
18. Lv, Y.Q., Lee, C.K.M.: Application of Hierarchical Colored Petri Net in Distributed Manufacturing Network. In: Proceedings of the 2010 IEEE IEEM (2010)
19. Long, S.: Analysis of Concurrent Security Protocols Using Colored Petri Nets. In: International Conference on Networking and Digital Society (2009)
20. Chen, Z., Chen, X.-W., Zhang, Z.-W.: IPSec Modeling Based Color Petri Nets. In: International Conference on Communications, Circuits and Systems Proceedings, vol. 3, pp. 1655–1659 (2009)
21. Xu, Y., Xie, X.: Modeling and analysis of security protocols using colored petri nets. Journal of Computers 3 (2011)
22. Ayachit, M.M., Xu, H.: A Petri Net Based XML Firewall Security Model for Web Services Invocation. In: Proceedings of the Iasted International Conference Communication, Network, and Information Security (October 2006)
23. Wu, R., Li, W., Huang, H.: An Attack Modeling Based on Hierarchical Colored Petri Nets. In: International Conference on Computer and Electrical Engineering (2008)
24. Yang, Y., Tan, Q., Xiao, Y.: Verifying Web Services Composition based on Hierarchical Colored Petri Nets. In: Proceedings of the first International Workshop on Interoperability of Heterogeneous Information Systems. ACM (2005)
25. Kheir, N., Cuppens-Boulahia, N., Cuppens, F., Debar, H.: A Service Dependency Model for Cost-Sensitive Intrusion Response. In: Gritzalis, D., Preneel, B., Theoharidou, M. (eds.) ESORICS 2010. LNCS, vol. 6345, pp. 626–642. Springer, Heidelberg (2010)
26. Debar, H., Curry, D., Feinstein, B.: The Intrusion Detection Message Exchange Format (IDMEF). RFC 4765, Internet Engineering Task Force (March 2007), http://www.ietf.org/rfc/rfc4765.txt
27. Vinter Ratzer, A., Wells, L., Lassen, H.M., Laursen, M., Qvortrup, J.F., Stissing, M.S., Westergaard, M., Christensen, S., Jensen, K.: CPN tools for editing, simulating, and analysing coloured petri nets. In: van der Aalst, W.M.P., Best, E. (eds.) ICATPN 2003. LNCS, vol. 2679, pp. 450–462. Springer, Heidelberg (2003)