Federated Autonomic Network Access Control

Simon N. Foley, William M. Fitzgerald and Wayne Mac Adams
Department of Computer Science,
University College Cork, Cork, Ireland.
Email: s.foley@cs.ucc.ie,wfitzgerald@4c.ucc.ie, w.macadams@4c.ucc.ie

I. Introduction

Network Access Controls (NAC) are widely used to provide endpoint security typically complementing existing application-based security controls. NAC security mechanisms, for instance firewalls, are routinely prescribed as requirements for compliance to security standards such as PCI-DSS and ISO 27000. However, the effectiveness of a NAC configuration may be hampered by poor understanding and/or management of the overall security configuration, which may in turn, unnecessarily expose the enterprise to known security threats. New threats and/or service requirements often result in firefighting by ad-hoc modification to an already large and complex configuration. This complexity is further compounded by the diverse range of NAC mechanisms used to secure an enterprise; ranging from firewalls and proxies to NAC-style controls within applications themselves. As a consequence, it can be difficult to ensure that the current NAC configuration is effective, that is, it sufficiently mitigates threats while providing necessary access to services. Ensuring ongoing best-practice NAC administration can be costly as it requires expert knowledge in a rapidly changing field.

Knowledge Engineering techniques can be used to provide expert and automated support for the management of NAC configurations. A knowledge base is being developed that contains detailed prescriptions for NAC configurations that are compliant with security standards and best practices. These *catalogues of best practice* describe how known threats are mitigated by NAC configurations and are continually updated to reflect newly discovered vulnerabilities and revisions to best practice. We are building a distributed security management framework whereby a network of agents can monitor threats and use these catalogues of best practice to automatically generate compliant NAC configurations. Following [1], this discovery-analysis-reconfiguration form the basis of a security risk management process for Network Access Controls.

II. Approach

This research builds upon [2], where knowledge about detailed NAC configuration, enterprise-level security requirements including best practice recommendations and their relationships are modelled, queried and reasoned over within an ontology-based framework. An ontology provides a conceptual and formal model of a domain of interest. Ontologies are developed for infrastructure-level access-control mechanisms [2] (Linux iptables and TCP-Wrapper firewalls) and application-level access-control mechanisms [3] (Openfire and Ejabbered XMPP servers).

A threat-based approach is taken to structure knowledge about the management of a NAC configuration. *Semantic Threat Graphs* [2], a variation of the traditional threat tree, are encoded within the ontology-based framework in order to relate knowledge about enterprise-level security requirements, best practice recommendations and access-control rules in terms of assets, threats, vulnerabilities and countermeasures.

Consider, an enterprise-level security requirement that states: *Server-to-Server federation is not permitted except between trusted XMPP servers with an IP address of 143.239.75.235 and 193.1.193.140*. The relationships between assets, threats, vulnerabilities and countermeasures can be described in the following way. The XMPP server (asset) through conventional TCP protocol communication cannot authenticate IP addresses (vulnerability) and thus, cannot control access to its XMPP federation port. As a consequence, the XMPP server is threatened by unintended IP address access. This threat is mitigated by a countermeasure, for example a firewall access-control rule, that only permits the intended IP address access to the XMPP server.

The semantic threat graphs approach takes advantage of an ontology's ability to share and integrate knowledge within other ontologies. Thus, the iptables and TCP-Wrapper firewall ontologies and XMPP server ontologies are reused to describe detailed countermeasure configurations. For example, the countermeasure for the previous example could be configured as an iptables firewall rule that considers the following filter attributes: the protocol (TCP), source IP address (Intranet subnet), destination IP address (XMPP server's IP address) and destination port (port 5269).

A knowledge-base of best practice countermeasures against known threats are encoded as semantic threat graphs. Ontologies for best practice standards for firewalls, Email servers, Web servers and XMPP servers, are developed [2], [3]. For example, XEP-0205 [4] and NIST SP800-123 [5] recommend multiple countermeasures such as connection throttling that mitigate against the threat of Denial of Service, when hosting an XMPP server. This knowledge-base is searchable—a suitable countermeasure can be found for a given threat—and provides the basis for the automatic generation of NAC configurations.

Synthesis of NAC configurations uses a set of pre-defined catalogues of NAC best practice countermeasures, rather than over the space of all possible NAC configurations. While this

approach has the disadvantage of not discovering 'new' forms of access-control countermeasures, it has the advantage of allowing a complete modelling of NAC configuration.

The ontology-based framework facilitates the construction of a knowledge-base of candidate queries. Thus, providing inexperienced security administrators with the ability ask expert questions about the effectiveness of an existing NAC configuration. These candidate queries could be based on testing for best practice compliance. For example, to test if a specific XMPP access-control configuration is XEP-0205 [6] compliant, the security administrator can draw upon a set of candidate queries from the knowledge-base, such as 'Is the XMPP server protected from simultaneous connection attempts from the same IP address?'

While the access-control rules on an individual basis may be compliant with the enterprise-level security requirements, the structural relationships between the access-control rules themselves may introduce a scenario such that the overall configuration is inconsistent with the enterprise-level security requirements. Such misconfiguration may occur between access-control rules that are deployed on a single NAC mechanism or across multiple inter-operating NAC mechanisms. Note, while ontology-based structural analysis, for example the detection of shadowed access-control rules have been explored, the current prototype does not consider structural analysis.

III. Prototype

A prototype NAC configuration agent has been implemented and its high-level components are depicted in Figure 1. An

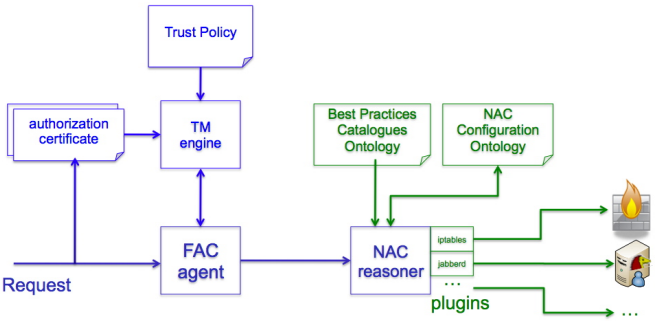

Fig. 1. FANAC Agent Components

Agent manages the current configuration-state of multiple NAC controls modeled as an ontology, as outlined in the previous section and NAC-specific plugins map the configuration ontology to low-level NAC configuration files. The goal of the Agent is to accept remote requests to reconfigure the NAC controls (ontology) while remaining compliant with the best-practices catalogues. The KeyNote Trust Management system (TM) is used by the Agent to determine whether a principal is authorized for its request. The current Agent accepts requests to grant new access to services; we are extending this to include revocation of access and threat/mitigation update.

A (grant access) request is a tuple $\langle \mathsf{srcIP}, \mathsf{dstIP}, \mathsf{dstPrt} \rangle$ (signed by requester owning public key K_R). If it is authorized by the TM system, the request is translated to a SWRL rule [9] with which to execute against the configuration ontology. This in turn synthesizes new access-control rules in the configuration ontology. The SWRL synthesis rules are provided by NAC-specific plugins which are used by the NAC reasoner [9] following a *Wrapper Design Pattern* [7]. For example, the iptables plugin provides a SWRL rule, part of which generates a new iptables access-control rule, to grant the access request, while mitigating a threat $?t$.

$$
\begin{aligned}
&Asset(?a) \wedge Threat(?t) \wedge Vulnerability(?v) \wedge \\
&TemplateIPTablesRule(\mathsf{iptr}_{temp}) \wedge hasWeakness(?a, ?v) \wedge \\
&exploits(?t, ?v) \wedge mitigates(\mathsf{iptr}_{temp}, ?v) \wedge \\
&hasDstIP(?a, \mathsf{dstIP}) \wedge hasSrcIP(?t, \mathsf{srcIP}) \wedge \\
&hasDstPort(a?, \mathsf{dstPrt}) \wedge \\
&swrlx : makeOWLIndividual(?iptr, \mathtt{iptr}_{temp}, ?t, ?a, ?v) \\
&\rightarrow IPTablesRule(?iptr) \wedge mitigates(?iptr, ?v) \wedge \\
&\quad hasSrcIP(?iptr, \mathsf{srcIP}) \wedge hasDstIP(?iptr, \mathsf{dstIP}) \wedge \\
&\quad hasDstPort(iptr?, \mathsf{dstPrt}) \wedge hasAction(?iptr, \mathsf{accept})
\end{aligned}
$$

The plugin also defines the mapping from (iptables) ontology elements to the iptables access-control rule syntax. We have implemented plugins for ejabberd, TCP Wrapper and iptables.

An early version of this Agent [8] hardcoded only XMPP configuration requests: it did not provide an extensible plugin architecture nor use use an ontology to model configuration state. A demonstration of the current Agent implemented in a Home Area Network router can be found at `www.4c.ucc.ie/security/hanac.mp4`.

Acknowledgment

This research has been supported by Science Foundation Ireland grant 08/SRC/11403.

References

[1] S. N. Foley, "Security risk management using internal controls," in *Proceedings of the first ACM workshop on Information security governance*, ser. WISG '09. New York, NY, USA: ACM, 2009, pp. 59–64. [Online]. Available: http://doi.acm.org/10.1145/1655168.1655179

[2] S. N. Foley and W. M. Fitzgerald, "Management of Security Policy Configuration using a Semantic Threat Graph Approach," *Journal of Computer Security, Volume 19, Number 3, IOS Press*, May 2011.

[3] W. M. Fitzgerald and S. N. Foley, "Management of Heterogeneous Security Access Control Configuration using an Ontology Engineering Approach," *3rd ACM Workshop on Assurable and Usable Security Configuration, Chicago, USA*, October 2010.

[4] P. Saint-Andre, "XEP-0205: Best Practices to Discourage Denial of Service Attacks," *http://xmpp.org*, 2009.

[5] K. Scarfone, W. Jansen, and M. Tracy, "Guide to General Server Security: Recommendations of the National Institute of Standards and Technology," *NIST Special Publication 800-123*, July 2008.

[6] J. Wack, K. Cutler, and J. Pole, "Guidelines on Firewalls and Firewall Policy: Recommendations of the National Institute of Standards and Technology," *NIST-800-41*, 2002.

[7] E. Gamma *et al.*, *Design Patterns, Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995.

[8] S. N. Foley and W. M. Adams, "Trust management of xmpp federation," in *Integrated Network Management*, N. Agoulmine, C. Bartolini, T. Pfeifer, and D. O'Sullivan, Eds. IEEE, 2011, pp. 1192–1195.

[9] M. O'Connor, H. Knublauch, S. Tu, B. Grossof, M. Dean, W. Grosso, and M. Musen, "Supporting Rule System Interoperability on the Semantic Web with SWRL" *4th International Semantic Web Conference, Ireland*, 2005.