

Using an Information Model and Associated Ontology for Selection of Policies for Conflict Analysis

Steven Davy, Brendan Jennings
Waterford Institute of Technology, Ireland
{sdavy, bjennings}@tssg.org

John Strassner
Motorola Labs, USA
john.strassner@motorola.com

Abstract

We present an analysis process targeting identification of potential policy conflicts within sets of policies relating to multiple network devices and the security services deployed on them. The process targets pre-deployment identification of potential conflicts between a newly created (or modified) policy and already deployed policies. It employs an algorithm which, with the aid of an ontology, selects the relevant subset of policies that should be compared with the "candidate" policy, together with an algorithm that identifies the relationships between a given pair of policies and compares these to a conflict signature pattern encoded in an information model. Operation of the process is illustrated via a scenario describing how it can identify conflicts between firewall filtering policies and IPSec VPN policies that are deployed on different network devices.

1. Introduction

Security services for communications networks require deployment of policies across multiple network devices. Detection of conflicts between policies for a given service in a distributed context is itself a challenging task, the difficulty of which is exacerbated by undesired interactions between policies relating to different security services. For example, policies dictating the use of IPsec in tunnel mode result in original IP headers being masked, so that filtering policies deployed in egress firewalls can become ineffective. This paper outlines initial work on development of a process that facilitates conflict analysis of policies relating to different services, deployed on multiple devices. Central to the process is the use of an information model and associated ontology to embody knowledge relating to the relationships between policies, and their use by policy selection and conflict identification algorithms.

In recent years many researchers have addressed issues relating to the problem of ensuring the consistency and efficacy of sets of deployed policies. For example Al-Shaer et

al. [1] address firewall policy conflict detection from both a centralized (single firewall) and distributed (multiple) firewall basis, using a policy relationship tree to detect potential conflicts. Hamed et al. [7, 6] investigate the analysis of centralised and distributed IPsec policy conflicts, examining specific complex conflicts that can arise across multiple devices. Similar work addresses other aspects of firewall policy conflict [3] and IPsec policy conflict [4]. We do not claim that our proposed process is more powerful than such approaches, rather that it provides an encompassing framework in which the knowledge indicating the likelihood of a conflict is encoded in an information model and ontology, so that the analysis algorithms can be generic in nature. Furthermore, our approach involves the use of a policy selection algorithm, similar in motivation to that described by Lin et al. [8], which improves the efficiency of the policy analysis process through pre-selection of relevant subsets of the deployed policy based for further analysis. The paper is structured as follows: section 2 outlines the policy conflict analysis process; section 3 describes a usage scenario illustrating the process as applied to interaction firewall filtering and IPsec VPN policies deployed across multiple routers; finally, section 4 summarizes the contribution of the paper.

2. Policy Conflict Analysis Process

The policy conflict analysis process is designed to analyse, on a pairwise basis, a candidate (new or modified) policy with already deployed policies. As a preliminary step a selection algorithm returns the subset of the deployed set of policies that are relevant for comparison with the candidate policy. A conflict analysis algorithm then uses knowledge embodied in an information model and ontology to detect policy inter-relationships that may indicate potential conflict.

Our policy analysis process is depicted in figure 1; its steps are as follows. When a policy is created or modified by a policy author it is submitted to the selection algorithm (1). The policy type is used to discover the policy selection rules defined in the ontology that determine the deployed

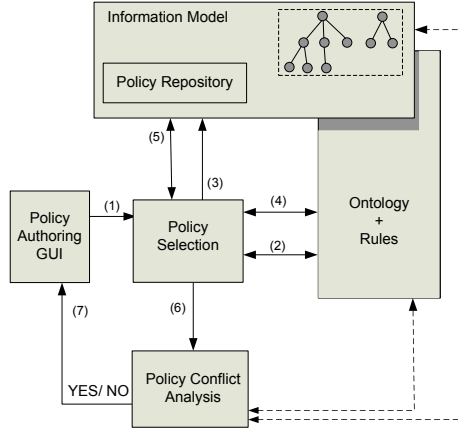


Figure 1. Policy Selection and Conflict Analysis

$$\begin{bmatrix}
 s_{sb} & s_{sp} & s_{eq} & s_{cor} & 0 & s_{ot} \\
 t_{sb} & t_{sp} & t_{eq} & t_{cor} & 0 & t_{ot} \\
 e_{sb} & e_{sp} & e_{eq} & e_{cor} & e_{mux} & e_{ot} \\
 c_{sb} & c_{sp} & c_{eq} & c_{cor} & c_{mux} & c_{ot} \\
 a_{sb} & a_{sp} & a_{eq} & a_{cor} & a_{ctd} & a_{ot}
 \end{bmatrix}$$

Figure 2. Conflict Signature Matrix

policies that should be retrieved and analysed for conflict (2). Selection rules are added to the associated ontology and are specific rule patterns that must evaluate to true for a policy to be selected for further analysis. An example of selection rules are provided in the case study in section 3. Once the rules are returned, they are applied by the policy selection algorithm, with each rule being used to retrieve a subset of policies from the repository of deployed policies (3). The ontology (4) and the information model (5) may be queried in order to fully satisfy the requirements of a selection rule. The selected subset of policies are forwarded to the conflict analysis algorithm (6), where, as described below, they are individually compared with the candidate policy (using the information model and ontology as appropriate). If no potential conflict is identified, the policy author is notified that the candidate policy can be deployed, otherwise a potential conflict is flagged and information regarding the relationship between the policies is relayed to the author (7).

The conflict analysis algorithm is broken into two phases, as described fully in [5]. The first phase builds a *policy relationship matrix* between the two policies. The policy relationship matrix, as depicted in figure 2, classifies each policy by its base component types: events, conditions,

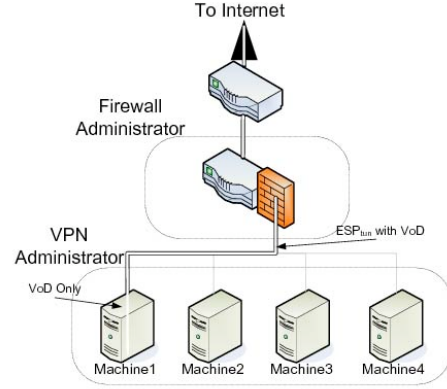


Figure 3. Policy Conflict Scenario

actions, subjects and targets. Note that this can be adapted to disparate policy language representations through the use of alternative component types. For each component type, the candidate and deployed policies are compared in a number of ways, through querying of the information model and ontology. The first row in the matrix relates the subjects of the two policies together (s- prefix), the second relates the targets (t- prefix) and similarly for events (e-), conditions (c-) and actions (a-). The basic relationships are for subset (-sb), superset (-sp), equality (-eq) and correlation (-cor), all of which can be identified via a well formed information model (as described in [5] we use the DEN-ng model). Additionally, we use a single relationship type "ontology" (-ot) to represent arbitrary semantic relationships between policies.

In the second phase of the algorithm the constructed policy relationship matrix is compared against a conflict signature relationship matrix using a comparison operator similar to that defined in [5] (but extended to support the ontology based relationships). If this operator returns a value of 1 then a potential conflict has been identified. It should be noted that the conflict analysis algorithm is independent of the application domain, with all the information required for the analysis being encoded in the information model and the ontology. The advantage of harnessing the ontology in this fashion is that the policy components no longer need to rely on the ability of the information model to represent relationships, as more extensive semantic relationships can now be represented and discovered. Furthermore, the algorithm can benefit from relationships that have been inferred or otherwise automatically generated due to ontology classification rules or subsumption.

ID	SrcIP	DestIP	DstPrt	Action
1	Machine1	InternetIP1	VoD	D_{rop}
2	Machine2	InternetIP2	HTTP	A_{llow}
3	Machine3	InternetIP2	IPsec	A_{llow}

Figure 4. Firewall policies

ID	SrcIP	DestIP	DstPrt	Action
1	Machine1	InternetIP1	*	ESP_{tun}
2	Machine2	InternetIP2	*	AH_{tun}
3	Machine3	InternetIP2	*	AH_{tra}

Figure 5. VPN policies

3. Scenario

The scenario we use to illustrate the capabilities of the policy selection and conflict analysis process involves a specific form of conflict that can arise amongst security-related IP traffic management policies (firewall filtering and IPsec VPN policies) deployed across multiple routers. The network topology, as depicted in figure 3, is assumed to be owned by a single organization, which divides it into multiple management domains, controlled by multiple network administrators. The conflict we explore relates to the fact that IPsec in tunnel mode masks the original IP header, making it impossible for egress firewall routers to recognise the true nature of the IP traffic that passes through them.

The firewall administrator has the responsibility of defining policy that controls the type of traffic that can enter and leave the network. A sample of the policies defined by the administrator are depicted in figure 4. The three policies defined are used to drop outgoing video on demand (VoD) traffic, allow HTTP traffic and allow IPsec VPN traffic, respectively. A VPN administrator has the responsibility of defining policies that control the initialization and securing of VPNs through the configuration of IPsec services on end-user machines. The policies defined on the end-user machines are depicted in figure 5. Policy 1 defines that IP traffic leaving Machine1 destined to a machine on the Internet with IP address InternetIP1 must be encapsulated and encrypted in an IPsec tunnel. Policies 2 and 3 are similar except that policy 2 ensures authentication in tunnel mode and policy 3 ensures authentication in transport mode.

A conflict arises when a user on Machine1 sends VoD traffic to InternetIP1. The VPN policy 1 ensures that the packet is encrypted, including the header information, thus masking the fact that the traffic is of type VoD. When the IP traffic passes through the firewall router, it is classed as IPsec traffic (not VoD traffic) and thus allowed pass. This is a conflict as the passing of VoD traffic through the firewall router violates the intention of the firewall filtering policies. We now demonstrate how our selection and conflict analysis

$Policy \sqsubseteq SecurityPolicy$
 $SecurityPolicy \sqsubseteq IPsecPolicy \sqcup FirewallPolicy$
 $IPsecPolicy \sqsubseteq IPsecTunPolicy \sqcup IPsecTraPolicy$

Figure 6. Policy classifications

$disjointFrom (FirewallPolicy, IPsecTunPolicy)$
 $disjointFrom (ESP_{Tun}, Drop)$
 $disjointFrom (ESPAH_{Tun}, Drop)$

Figure 7. Disjoint classes

processes aid in the discovery of this conflict.

The first step is to generate the associated ontology from the information model of the system and augment it with search rules and classifications to aid in policy conflict detection. The ontology can be constructed by following the procedure laid out in [2]. Some important classifications in the ontology are policy types as depicted in figure 6. A Policy can therefore be specialised into a SecurityPolicy, which in turn can be either an IPsecPolicy or a FirewallPolicy. Another addition to the ontology is depicted in figure 7, which states that FirewallPolicy is disjointFrom IPsecTunPolicy, and that any *tunnel encryption action* class is also disjointFrom the *drop action* class.

One of the specific search rules that are added to the ontology is outlined in figure 8. This search rule is designed especially for FirewallPolicy types (line 1). This particular search rule is looking for IPsecTunPolicy deployed policies (line 2). Lines 3 and 4 effect the retrieval of the associated targets of the policies referring to the router interfaces they are deployed on. Line 5 ensures that the deployed policies retrieved are on different routers' interfaces. Therefore, only deployed policies on other router interfaces can be considered to satisfy this rule. Lines 6 and 7 effect retrieval of the source and destination IP address of the candidate policy, whereas line 8 effects retrieval of the IP address of the target device of the candidate firewall policy. In line 9 the

1. $FirewallPolicy (?cand) \wedge$
2. $IPsecTunPolicy (?dep) \wedge$
3. $hasTarget (?cand, ?tc) \wedge$
4. $hasTarget (?dep, ?tx) \wedge$
5. $differentFrom (?tc, ?tx) \wedge$
6. $sourceIP (?dep, ?sipd) \wedge$
7. $destIP (?dep, ?dipd) \wedge$
8. $interfaceAddress (?tc, ?tip) \wedge$
9. $info : onPath (?sipd, ?dipd, ?tip) \wedge$
10. $\rightarrow select (?dep) \wedge$
11. $linked (?cand, ?dep)$

Figure 8. Search rules

information model is queried to ascertain whether the traffic flow referenced in the source / destination IP addresses on the IPsecTunPolicy pass through the interface that contains the candidate policy. If this returns true then there may be a tunneling conflict and more analysis is required. (Note that information model queries may occur at any phase in the query, however it is advised to wait as late as possible as the queries may be time consuming). By line 10 the policy has been selected and in line 11 a relationship called 'linked' is created to establish explicitly that the two policies are linked.

With the search rule defined we can now step through the process of detecting the policy conflicts. The process begins as the firewall administrator adds the policy numbered 1 in figure 4. We assume that all other policies discussed above have already been added and analysed for conflict. The next step in the process is policy selection. The appropriate search rule is retrieved from the ontology which refers to FirewallPolicy types, therefore the search rule depicted in figure 8 is returned. This rule is evaluated and the list of potential policies is enumerated. The policies returned are policies 1 and 2 from figure 5, since they both satisfy all the criteria outlined in the search rule. Now that the policies have been selected they are sent to the conflict analysis algorithm for further processing.

The first comparison is between the candidate firewall policy and the deployed policy, VPN policy 1. Given the results of the selection rule discussed above the targets of the two policies are linked together (note that in the information model-only approach outlined in [5] this would not have been the case). The existence of the target relationship is reflected as a 1 in the 'tot' field of the relationship matrix; similarly the subjects of the policies are marked as related and the policy events are equal. The conditions overlap in that both policies are applied to overlapping IP header information; specifically, the firewall rule is a superset of the VPN policy. The action components of the two policies cannot however be directly compared, so the ontology must be queried. The candidate policy action is *DROP* and the deployed policy action is *ESP_{tun}*, the ontology specifies that these two classes of actions are *disjointFrom* each other and we can conclude that they conflict with each other. The second policy in the list is however not deemed to conflict as it is an authentication tunnel and the internal header information is not encrypted and can be accessible at the firewall for examination.

4. Summary

This paper outlined a process for policy analysis, incorporating policy selection and conflict analysis algorithms that harness knowledge embodied within a system information model and associated ontology. Separation of the

application-specific information and knowledge required for conflict analysis from the policy language representation of individual policies, as well as from the analysis algorithms themselves, results in a more flexible approach that is more readily extensible as the system evolves. Furthermore, the information model and ontology can be used to dynamically build relationship models for deployed policy sets that can be used by the selection algorithm, which has the effect of reducing the number of policy comparisons required before a candidate policy is deployed.

Acknowledgements

This work has received support from Science Foundation Ireland under the "Autonomic Management of Communications Networks and Services" award (grant no. 04/IN3/I404C).

References

- [1] E. Al-Shaer, H. Hamed, R. Boutaba, and M. Hasan. Conflict classification and analysis of distributed firewall policies. *IEEE Journal on Selected Areas in Communications*, 23(10):2069–2084, 2005.
- [2] K. Barrett, S. Davy, B. Jennings, S. van der Meer, and J. Strassner. A Model Based Approach for Policy Tool Generation and Policy Analysis. *Proc. IEEE Global Information Infrastructure Symposium (GIIS 2007)*, pages 99–105, 2007.
- [3] V. Capretta, B. Stepien, A. Felty, and S. Matwin. Formal correctness of conflict detection for firewalls. *Proc. of the ACM workshop on Formal Methods in Security Engineering (FMSE 2007)*, pages 22–30, 2007.
- [4] K.-H. Chen, Y.-S. Liu, T.-J. Liu, and C.-R. Dow. ZERO-Conflict: A Grouping-Based Approach for Automatic Generation of IPSec/VPN Security Policies. In *Proc. of the 17th IFIP/IEEE Distributed Systems: Operations and Management, (DSOM 2006)*, pages 197–208, 2006.
- [5] S. Davy, B. Jennings, and J. Strassner. Application Domain Independent Policy Conflict Analysis Using Information Models. *to appear, Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS 2008)*, 2008.
- [6] H. Hamed and E. Al-Shaer. Taxonomy of conflicts in network security policies. *IEEE Communications Magazine*, 44(3):134–141, 2006.
- [7] H. Hamed, E. Al-Shaer, and W. Marrero. Modeling and Verification of IPSec and VPN Security Policies. *Proc. of the 13th IEEE International Conference on Network Protocols, ICNP*, pages 259–278, 2005.
- [8] D. Lin, P. Rao, E. Bertino, and J. Lobo. An approach to evaluate policy similarity. In *Proc. of the 12th ACM symposium on Access control models and technologies (SACMAT 2007)*, pages 1–10, New York, NY, USA, 2007.