Reasoning about the Security Configuration of SAN Switch Fabrics

William M. Fitzgerald
Dept. of Computer Science
University College Cork
Cork, Ireland
Email: wfitzgerald@4c.ucc.ie

Simon N. Foley
Dept. of Computer Science
University College Cork
Cork, Ireland
Email: s.foley@cs.ucc.ie

***Abstract*—Management of a switch fabric security configuration, a core component of Storage Area Networks, is complex and error prone. As a consequence, misconfiguration of and/or a poor understanding of a switch fabric may unnecessarily expose an enterprise to known threats. A formal model of a switch security configuration is presented. This model is reasoned over to help manage complex switch fabric security configurations.**

I. Introduction

An integral part of a *Storage Area Network* (*SAN*) is its switch fabric. Intuitively, a *SAN* switch provides interconnections between SAN client nodes (initiators) and storage array nodes (targets), primarily to exchange SCSI traffic [1].

A significant challenge in providing security for a SAN is attaining a degree of confidence that the security configuration of its switches adequately addresses the (security) threats. A misconfiguration may result in unapproved access, the denial of approved access or inadequate mitigation of threats to SAN nodes. In practice, switch security configuration is more complex than simply providing firewall-like access-control rules. Rather, switch security configuration must be considered in the context of other security services required by the switch fabric, thereby increasing the likelihood of misconfiguration. For example, when generating access-control rules for the enterprise-level SAN security requirement: "*permit switch fabric interaction with trusted remote authentication servers*", a security configuration is not just defined in terms of a set of packet-filter access-control rules. The security configuration of the switch fabric must also consider a set of authentication access-control rules that specify, at an application-level, the trusted remote authentication servers, and, furthermore, whether the authentication access-control rules defined (perhaps on a single switch source) are to be distributed across the entire switch fabric (another set of application-level access-control rules).

This paper considers the problem of reasoning about and managing complex security configurations of SAN switches. A formal model for switch security configuration is developed using *Description Logic* [2]. This approach builds on previous research [3]–[5] that demonstrated the effectiveness of using Description Logic to model and reason about firewall rules.

While there are extensive research results on query [3], [6], [7] and structural [3], [8], [9] analysis of firewall rules, the authors are unaware of research published on the similar analysis of SAN switch rules. Switch security configuration is typically a more complex proposition than conventional firewalls and the paper describes how such reasoning can be done in the proposed model of switch security. For example, a query such as: "*has read-only access to a particular disk on a specific target storage array within a given zone defined in the context of a particular VSAN been granted to trusted initiator client nodes?*" may be formalized within the model. Structural analysis is also considered to detect access-control rule conflicts.

The contribution of this paper is a formal model for switch security configuration with which to perform query and structural analysis. In addition, structural analysis definitions that define inter-shadowing and inter-spurious conflicts between firewall-based and zone-based switch security configurations are presented.

The paper is organised as follows. Section II provides an overview of the switch fabric security configuration challenges. An overview of Description Logic and Semantic Web Rule Language is presented in Section III. Section IV presents a Description Logic based model for SAN switch security configuration. Additional structural analysis definitions are defined in Section V. Section VI provides some examples that demonstrate analysis of switch fabric security configurations.

II. Switch Security Configuration

This section provides examples of security configuration challenges that need to be considered with respect to configuring access-control rules for switch management and for traffic routed through the switch.

A. Switch Management

Configuring switch management access controls is not simply about opening the relevant management service ports such as SSH or remote authentication service ports for example RADIUS. One needs to consider whether management access to the switch is permitted for out-of-band management (over an Ethernet interface) or in-band management (where TCP/IP traffic is tunnelled over a Fibre Channel interface). One may also wish to permit certain SAN initiators (for example, IP address white-list) and deny others. It may also be considered prudent to define a set of access-control rules

that explicitly state what privileges a SAN administrator is authorised for, once authenticated to a switch. For example, SAN administrators are typically authorised for root privileges while SAN network operators are not. Similarly, are *Authentication, Authorisation & Accounting* (*AAA*) services performed locally on each switch within the SAN fabric or centrally through a remote AAA server farm typically hosted on a separate LAN. If performed remotely, are the access-control rules correctly providing intended secure SAN-to-LAN AAA communication? One has also to consider reflecting locally on each switch, a comparable set of AAA access-control rules defined on remote AAA servers as a precautionary redundancy countermeasure should remote AAA communication fail.

B. Traffic Management

With respect to traffic routed through the switch between initiator clients and target storage arrays, it is not simply about making for example the iSCSI TCP port 3260 accessible for all traffic. Furthermore, iSCSI traffic for example does not necessarily have to communicate on the IANA [10] recommended port of 3260. It may also be a security requirement to deny certain SAN initiators (for example, IP address or World Wide Name black-list) access to all storage arrays, only accept iSCSI traffic from some initiators and require other initiators to use iSCSI over IPSec. One may also need to consider access-control rules that restrict certain initiators access to a specific disk or set of disks within a storage array and so forth.

C. Structural Misconfiguration Management

While the SAN 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. For example, firewall-based access-control rules in a switch configuration are tested in the sequence in which they appear in the configuration. That is, once a packet has been successfully matched against an access-control rule, no further rule tests are carried out for that packet. Thus, an access-control rule placed out of sequence may unintentionally cause a misconfiguration. Consider two access-control rules, where one permits iSCSI traffic to a storage array and the other denies access to all network resources, including the storage array. Depending on the sequence of these two access-control rules, a misconfiguration may result.

Misconfiguration may also occur between access-control rules deployed on multiple inter-operating SAN access controls. For example, an upstream switch may be unintentionally denying intended storage array iSCSI traffic that is correctly configured on an another downstream switch.

In practice, managing a SAN switch security configuration that is aligned with the enterprise-level security requirements is complex and error-prone. Configuration is largely dependent on the expert-knowledge of the SAN administrator drawing upon best practice and standards.

III. Description Logic and SWRL

Description Logic (*DL*) is a decidable portion of first-order logic [2]. DL concepts represent sets of individuals (instances) and properties (roles) represent binary relations applied to individuals. For example, the DL assertion:

$$\mathit{StorageArray} \sqsubseteq \mathit{Server} \sqcap \exists_{\geq 1} \mathit{hosts.Service} \sqcap \exists_{\geq 1} \mathit{isProtectedBy.ProtectionServer}$$

specifies that a storage array hosts one or more services (for example iSCSI) and is protected by one or more protection servers (for example firewalls and/or switches).

The *Semantic Web Rule Language* (*SWRL*) complements DL, providing the ability to infer additional information but at the expense of decidability. SWRL rules are horn-clause like rules written in terms of DL concepts, properties and individuals. A SWRL rule is composed of an antecedent (body) part and a consequent (head) part, both of which consist of positive conjunctions of atoms [11]. For example, the requirement that servers hosting iSCSI based services protected by protection servers require those protection servers to open port 3260, is expressed in the following SWRL rule.

$$\mathit{StorageArray(?sa)} \wedge \mathit{hosts(?sa,?iSCSI)} \wedge \mathit{hasPort(?iSCSI,\ 3260)} \wedge \mathit{isProtectedBy(?sa,?ps)} \rightarrow \mathit{hasOpenPort(?ps,3260)}$$

where `3260` is an atom/constant and *?sa*, *?iSCSI*, *?ps* represent unbound variables in the rule.

IV. Switch Access Control Model

A formal model for SAN switch security configuration is developed using *DL* [2] and the *SWRL* [11]. Note that in presenting the model components, for reasons of space, complete specifications in particular, disjoint axioms, sub-properties or closure axioms are omitted. Similarly for reasons of space, not all switch access controls modelled, for example port-based, iSCSI-based and remote authentication-based access controls, are presented. However, a relevant portion of the model is presented that conveys the challenges identified within this paper. While the model for the SAN switch security configuration presented in this paper is Cisco centric [12], [13], much of the model attributes are implementation neutral.

Concept $CiscoMDSRule$ defines the set of Cisco MDS access-control rules. This concept is further specialised to more specific sub-concepts, namely concepts $ExecRule$ and $ConfigRule$.

$$\mathit{CiscoMDSRule} \equiv \mathit{ExecRule} \sqcup \mathit{ConfigRule}$$

Concept $ExecRule$ defines a set of execution-mode access-control rules that enable temporary switch configuration modifications, debugging and display of system information. Concept $ConfigRule$ defines a set of configuration-mode access-control rules that enable permanent (across reboots) switch configuration modifications. These two sub-concepts are in turn further specialised to form a hierarchy of access-control rules. A fragment of this hierarchy is illustrated in Figure 1. The double-headed arrow represents a subsumption relation.

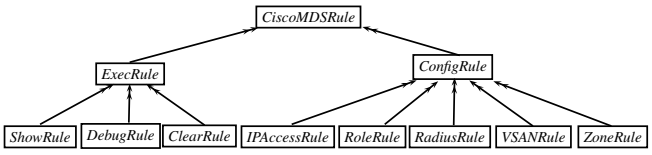

Fig. 1. Fragment of Cisco MDS Rule Hierarchy.

A. TCP/IP-based Access Control List

Traditional TCP/IP based traffic plays an important role within a SAN fabric. For example, configuration management of a SAN switch or communication with remote AAA servers are performed over TCP/IP protocols. Modern SAN switches have the ability to route TCP/IP traffic over traditional Ethernet or tunnelled over Fibre Channel interfaces. As a consequence, the SAN fabric is exposed to traditional TCP/IP based threats.

While upstream network access control mechanisms such as firewalls and Intrusion Detection Systems provide protection against known TCP/IP based threats, it is considered best practice to adopt a security in depth approach [14], [15]. Therefore, switch Access Control Lists (ACLs) are used to make decisions about whether or not to permit a packet.

Consider as a running example switch configuration management where management may be performed using the Fabric Manager (via SNMP) or the Command Line Interface (via SSH or Telnet). These management services are accessible by all SAN nodes connected to the switch whether authorised or not. Therefore, it becomes necessary, in addition to a management service's own security controls, to restrict management access to SAN administrator IP addresses only.

Concept *IPAccessListRule* represents a set of TCP/IP-based ACL rules. Each access-control rule takes the form of a series of filter conditions on packet fields that must be met in order for that access-control rule to be applicable with a consequent action for the matching packet. A TCP/IP access-control rule is composed of one or more ($\exists$ restriction) protocols (property $hasProto$), one or more source and destination IP addresses (properties $hasSrcIP$ and $hasDstIP$), one or more ports (property $hasDstPort$), zero or more ($\forall$ restriction) connection states (property $hasState$), zero or more ICMP Types (property $hasICMPType$) and one action.

$$\begin{aligned} \textit{IPAccessListRule} \sqsubseteq\ & \textit{ConfigRule} \sqcap \\ & \exists_{\geq 1} \textit{hasProto.Protocol} \sqcap \\ & \exists_{\geq 1} \textit{hasSrcIP.IPAddress} \sqcap \\ & \exists_{\geq 1} \textit{hasDstIP.IPAddress} \sqcap \\ & \exists_{\geq 1} \textit{hasDstPort.Port} \sqcap \\ & \forall_{\geq 0} \textit{hasState.State} \sqcap \\ & \forall_{\geq 0} \textit{hasICMPType.ICMPType} \sqcap \\ & \exists_{=1} \textit{hasAction.Action} \sqcap \\ & \exists_{=2} \textit{dependsOn.CiscoMDSRule} \end{aligned}$$

For reasons of space, concepts $Protocol$, $IPAddress$, $Port$, $State$ and $ICMPType$ are not defined in this paper. However, the reader is referred to [3] where these concepts are defined within a model of the TCP/IP stack developed as part of previous research [4].

Concept $Action$ is an enumerated set of actions (individuals) that can be taken against a matching packet.

$$\textit{Action} \equiv \{\texttt{permit}, \texttt{deny}, \texttt{log\&deny}\}$$

For a TCP/IP-based access-control rule to take effect it must be first assigned a switch interface and then assigned a direction (inbound or outbound) in which it is to filter packets [12]. Therefore, each access-control rule has a dependency ($dependsOn$ property) with exactly two other Cisco MDS rules, namely those defined by concepts *InterfaceRule* and *IPAccessGroup* respectively.

A hierarchy of TCP/IP-based ACLs are defined to represent categories of access-control rules that may be implemented by SAN switches. For example, concept $MgmtACL$ is representative of the TCP/IP-based access-control rules that are applicable to a switch's management interface.

$$\textit{MgmtACL} \sqsubseteq \textit{IPAccessListRule}$$

The following SAN security requirement stating that "*a SAN administrator host having an IP address of 192.168.1.20 is permitted inbound SSH access to the SAN switch management interface (mgmt0) with an IP address of 192.168.1.1*" is characterised by the following three Cisco MDS rules.

```
ip access-list mgmt-ACL
     permit tcp 192.168.1.20
     192.168.1.1 eq port ssh
interface mgmt 0
ip access-group mgmt-ACL in
```

Individual `mACL`, an instance of concept $MgmtACL$, represents the '`ip access-list`' access-control rule including its dependency relationship with access-control rules '`interface`' and '`ip access-group`'.

$$\begin{aligned} \textit{MgntACL}(\texttt{mACL}) \leftarrow\ & \textit{hasProto}(\texttt{mACL}, \texttt{tcp}) \sqcap \\ & \textit{hasSrcIP}(\texttt{mACL}, \texttt{ip192.168.1.20}) \sqcap \\ & \textit{hasDstIP}(\texttt{mACL}, \texttt{ip192.168.1.1}) \sqcap \\ & \textit{hasPort}(\texttt{mACL}, \texttt{p22}) \sqcap \\ & \textit{hasAction}(\texttt{mACL}, \texttt{permit}) \sqcap \\ & \textit{dependsOn}(\texttt{mACL}, \texttt{mgnt0}) \sqcap \\ & \textit{dependsOn}(\texttt{mACL}, \texttt{mgnt0Inbound}) \end{aligned}$$

where individuals `mgnt0` and `mgnt0Inbound` are instances of concepts *InterfaceRule* and *IPAccessGroup* respectively.

B. Role-based Access Control

The previous section discussed the importance of restricting access to management services of a switch to SAN administrators only. Adopting the principle of least privilege is considered best practice. Therefore, one may also want to restrict the set of privileges in terms of access-control rule permissions that each SAN administrator is authorised for.

Concept $RoleRule$ represents the set of roles (individuals) such that each access-control rule is composed of a single rule order index (property $hasOrder$), a privilege (property $canExecute$) with respect to one of five Cisco MDS rule categories, a single action (individuals `permit` or `deny`) applicable to that privilege, one or more users that a role is intended for (property *isRoleOf*) and zero or more VSANs for which that role $isAppliedTo$. Note, VSANs are discussed in Section IV-C and are analogous to *Virtual Local Area Network*s (*VLAN*s).

$$\begin{aligned} \textit{RoleRule} \sqsubseteq\ & \textit{ConfigRule} \sqcap \\ & \exists_{=1} \textit{hasOrder.Integer} \sqcap \\ & \exists_{=1} \textit{canExecute.}(\textit{ConfigRule} \sqcup \textit{ExecRule} \sqcup \\ & \quad \textit{DebugRule} \sqcup \textit{ShowRule} \sqcup \textit{ClearRule}) \sqcap \\ & \exists_{=1} \textit{hasAction.Action} \sqcap \\ & \exists_{\geq 1} \textit{isRoleOf.User} \sqcap \\ & \forall_{\geq 0} \textit{isAppliedTo.VSANDB} \end{aligned}$$

Concept $Role$ is further specialised to provide a hierarchy of roles. For example, a role for a SAN operator (concept $OpRole$) and a SAN administrator (concept $AdminRole$).

$$\textit{OpRole, AdminRole} \sqsubseteq \textit{RoleRule}$$

Role-based access control is important to minimise unnecessary threats with respect to unauthorised modification of switch configuration as a consequence of a compromised SAN administrator account with root privileges or the (un)intended execution of commands by a SAN administrator not intended to have such privileges. For example, the following Cisco MDS access-control rules may be (un)intentionally misused by user Eve such that user Alice who is logged into the switch is forcibly logged out or has her password changed, thus a Denial of Service ensues. As a consequence, it is important to avoid role-based access control misconfiguration.

```
clear user Alice
username Alice password y0u'v3B33nH4ck3d
```

The following SAN security requirement: "*User Eve, a restricted SAN administrator, is authorised for Exec access-control rules with the exception of the 'clear' access-control rules*" is encoded as the following set of concept $AdminRole$ individuals.

$$\begin{aligned} \textit{AdminRole}(\texttt{exec1}) \leftarrow\ & \textit{hasOrder}(\texttt{exec1}, 1) \sqcap \\ & \textit{canExecute}(\texttt{exec1}, \texttt{exec}) \sqcap \\ & \textit{hasAction}(\texttt{exec1}, \texttt{permit}) \sqcap \\ & \textit{isRoleOf}(\texttt{exec1}, \texttt{eve}) \\ \textit{AdminRole}(\texttt{exec2}) \leftarrow\ & \textit{hasOrder}(\texttt{exec2}, 2) \sqcap \\ & \textit{canExecute}(\texttt{exec2}, \texttt{clear}) \sqcap \\ & \textit{hasAction}(\texttt{exec2}, \texttt{deny}) \sqcap \\ & \textit{isRoleOf}(\texttt{exec2}, \texttt{eve}) \end{aligned}$$

Individuals `exec1` and `exec2` are representative of the following low-level Cisco MDS rule-set.

```
role name adminrole rule 1 permit exec
role name adminrole rule 2 deny exec feature clear
username eve role adminrole
```

Note due to page constraints, the `username` access-control rule that explicitly assigns user Eve to the administrator role is not presented. However the inverse property of *isRoleOf* is property $hasRole$ that has concept *User* as its domain and concept $Role$ as its range.

C. VSAN-based Access Control

A *Virtual Storage Area Network* (*VSAN*) is a logical partition of a physical SAN and provides a basis for traffic isolation between nodes that are physically connected within the same SAN [12]. VSANs, with advantages of redundancy and reduced hardware costs aside, play an important part in provisioning network access control. SAN nodes may only be a connected to a single VSAN thereby ensuring that traffic communicated within a VSAN is isolated from traffic in other VSANs. Note, each VSAN has its own dedicated routing and configuration management services.

From a security perspective, VSANs within an enterprise maybe used to define a virtual SAN fabric for each of its departments. For example, a separate VSAN for the Research, Sales and Human Resources departments. Consider the following SAN security requirement that states: "*Nodes within each department should have access to data stored on their respective storage arrays where access to data from other departments is prohibited*" as a running example. Without defining separate VSANs, all nodes connected to the same physical SAN fabric have the potential to communicate with nodes across departments (also noted in Section IV-A). Therefore, a SAN administrator should define relevant VSANs to ensure proper departmental demarcation.

A VSAN rule (individual) is defined to have a unique identifier (property $hasID$), operate over one or more interfaces (property *hasIface*), have an optional human readable name (property $hasName$) and a decision whether or not that VSAN is currently activated (property $isActivated$).

$$\begin{aligned} \textit{VSANRule} \sqsubseteq\ & \textit{ConfigRule} \sqcap \\ & \forall_{=1} \textit{hasName.String} \sqcap \\ & \exists_{=1} \textit{hasID.Integer} \sqcap \\ & \exists_{\geq 1} \textit{hasIface.Interface} \sqcap \\ & \exists_{=1} \textit{isActivated.Boolean} \end{aligned}$$

Further explanation of concept *Interface* is required. A SAN switch has many kinds of interfaces for example Fibre Channel interfaces (individuals of concept *FCIface*) and iSCSI interfaces (individuals of concept *ISCSIIface*. Such interfaces have a slot value and an interface value.

$$\begin{aligned} \textit{FCIface}, \textit{FVIface}, & \\ \textit{FCIPIface}, \textit{ISCSIIface} \sqsubseteq\ & \textit{Interface} \sqcap \\ & \exists_{\geq 1} \textit{hasSlotValue.Integer} \sqcap \\ & \exists_{\geq 1} \textit{hasIfaceValue.Integer} \end{aligned}$$

Consider the Research department VSAN as a running example, where the following security requirement states: "*External research partner nodes, internal research department nodes, the shared research storage array and the backup storage array should be isolated from other SAN traffic by defining a dedicated Research VSAN*".

The following Cisco MDS rule-set defines a Research VSAN called '`rSAN`' with an ID of `2` and operates over switch interfaces `iscsi 1/1`, `iscsi 2/1`, `fc 1/3` and `fc 1/4` with which to connect external and internal iSCSI initiator nodes and, research and backup storage array fibre channel target nodes respectively.

```
vsan database vsan 2 name rSAN interface iscsi 1/1
vsan database vsan 2 name rSAN interface iscsi 2/1
vsan database vsan 2 name rSAN interface fc 1/3
vsan database vsan 2 name rSAN interface fc 1/4
```

These low-level Cisco MDS access-control rules are encoded within the following DL assertions.

$$\begin{aligned} \mathit{VSANRule}(\texttt{rVSAN}) \leftarrow\ & \mathit{hasName}(\texttt{rVSAN}, \text{“}\texttt{rSAN}\text{”}) \sqcap \\ & \mathit{hasID}(\texttt{rVSAN}, 2) \sqcap \\ & \mathit{hasIface}(\texttt{rVSAN}, \texttt{iscsi1-1}) \sqcap \\ & \mathit{hasIface}(\texttt{rVSAN}, \texttt{iscsi2-1}) \sqcap \\ & \mathit{hasIface}(\texttt{rVSAN}, \texttt{fc1-3}) \sqcap \\ & \mathit{hasIface}(\texttt{rVSAN}, \texttt{fc1-4}) \sqcap \\ & \mathit{isActivated}(\texttt{rVSAN}, \texttt{true}) \end{aligned}$$

where the following holds for individuals `iscsi1/1`, `iscsi2/1`, `fc1/3` and `fc1/4`:

$$\begin{aligned} \mathit{ISCSIIface}(\texttt{iscsi1-1}) \leftarrow\ & \mathit{hasSlotValue}(\texttt{iscsi1-1}, 1) \sqcap \\ & \mathit{hasIfaceValue}(\texttt{iscsi1-1}, 1) \\ \mathit{ISCSIIface}(\texttt{iscsi2-1}) \leftarrow\ & \mathit{hasSlotValue}(\texttt{iscsi2-1}, 2) \sqcap \\ & \mathit{hasIfaceValue}(\texttt{iscsi2-1}, 1) \\ \mathit{FCIface}(\texttt{fc1-3}) \leftarrow\ & \mathit{hasSlotValue}(\texttt{fc1-3}, 1) \sqcap \\ & \mathit{hasIfaceValue}(\texttt{fc1-3}, 3) \\ \mathit{FCIface}(\texttt{fc1-4}) \leftarrow\ & \mathit{hasSlotValue}(\texttt{fc1-4}, 1) \sqcap \\ & \mathit{hasIfaceValue}(\texttt{fc1-4}, 4) \end{aligned}$$

D. Zone-based Access Control

Zone-based Access Control (or *Zoning*) provides a basis for fined-grained demarcation of nodes within a VSAN [12], [16]. A zone consists of members (initiator and target nodes). Members of a zone can access each other, while members across different zones cannot. Zones are defined within a VSAN. Zone membership is based on either IP addresses or World-Wide Names (WWNs) of the nodes connected to a switch fabric [12]. For example, a node may be referred to by its *pWWN*, that is, its *port World Wide Name* (analogous to a Ethernet MAC address).

Note, while VSANs and Zones provide traffic isolation they are different to one another. For example, VSANs provide routing, naming and zone protocols. These protocols are not available on a per-zone basis [12].

Concept $ZoneRule$ represents the set of access-control zone rules (individuals) that have a name (property $hasName$), $isAssignedTo$ a VSAN and have one or more $hasMember$ relationships with individuals of concept $Member$.

$$\begin{aligned} \mathit{ZoneRule} \sqsubseteq\ & \mathit{ConfigRule} \sqcap \\ & \exists_{=1} \mathit{hasName.String} \sqcap \\ & \exists_{=1} \mathit{isAssignedTo.VSANRule} \sqcap \\ & \exists_{\geq 1} \mathit{hasMember.Member} \end{aligned}$$

Concept $Member$ represents individuals (SAN nodes) that are members of one or more zones along the *isMemberOf* property. There are a number of disjunction axioms that define necessary conditions for concept membership. For example, an individual that is a member of a particular zone may be referred to by its pWWN (property *hasPWWN*). Members may have zero or more ports. An explanation of property $hasLUNID$ will be discussed in Section IV-D2. Note, only the relevant fragment of this definition is provided due to page limitation.

$$\begin{aligned} \mathit{Member} \sqsubseteq\ & \mathit{ConfigRule} \sqcap \\ & \exists_{\geq 1} \mathit{isMemberOf.Zone} \sqcap \\ & (\exists_{\geq 1} \mathit{hasIPAddress.IPAddress} \sqcup \\ & \exists_{\geq 1} \mathit{hasFWWN.FWWN} \sqcup \\ & (\exists_{\geq 1} \mathit{hasPWWN.PWWN} \sqcap \\ & \forall_{\geq 0} \mathit{hasLUNID.String})) \sqcap \\ & \forall \mathit{hasPort.Port} \end{aligned}$$

While a VSAN provides a logical demarcation of nodes it hosts from nodes hosted within other VSANs, it may be a further security requirement to restrict nodes within the same VSAN from communicating with each other. Consider as running example the following SAN security requirement: "*External research partner nodes are permitted access to the research storage array while internal Research department nodes are permitted to access both the research and backup storage arrays*".

The following is an example implementation of the SAN security requirement described above, where individual `exZone` and individual `inZone` are representative of external and internal zone access-control rules that permit external research partners (initiators `InitA` and `InitB`) access to the research storage array (target individual `resSA`), and an internal research department initiator (individual `InitC`) access to both the research storage array and the backup storage array (target

individual bakSA).

$$\begin{aligned}
\textit{ZoneRule}(\texttt{exZone}) \leftarrow\ & \textit{hasName}(\texttt{exZone}, \text{“ExZone”}) \sqcap \\
& \textit{isAssignedTo}(\texttt{exZone}, \texttt{rVSAN}) \sqcap \\
& \textit{hasMember}(\texttt{exZone}, \texttt{initA}) \sqcap \\
& \textit{hasMember}(\texttt{exZone}, \texttt{initB}) \sqcap \\
& \textit{hasMember}(\texttt{exZone}, \texttt{resSA}) \\
\textit{ZoneRule}(\texttt{inZone}) \leftarrow\ & \textit{hasName}(\texttt{inZone}, \text{“InZone”}) \sqcap \\
& \textit{isAssignedTo}(\texttt{inZone}, \texttt{rVSAN}) \sqcap \\
& \textit{hasMember}(\texttt{inZone}, \texttt{initC}) \sqcap \\
& \textit{hasMember}(\texttt{inZone}, \texttt{resSA}) \sqcap \\
& \textit{hasMember}(\texttt{inZone}, \texttt{bakSA})
\end{aligned}$$

where the following also holds:

$$\begin{aligned}
\textit{Member}(\texttt{initA}) &\leftarrow \textit{hasPWWN}(\texttt{initA}, \texttt{pwwn00:11:22:ab}) \\
\textit{Member}(\texttt{initB}) &\leftarrow \textit{hasPWWN}(\texttt{initB}, \texttt{pwwn00:11:22:cd}) \\
\textit{Member}(\texttt{initC}) &\leftarrow \textit{hasPWWN}(\texttt{initC}, \texttt{pwwn11:22:33:cd}) \\
\textit{Member}(\texttt{resSA}) &\leftarrow \textit{hasFWWN}(\texttt{resSA}, \texttt{fwwn22:33:44:ef}) \\
\textit{Member}(\texttt{bakSA}) &\leftarrow \textit{hasFWWN}(\texttt{bakSA}, \texttt{fwwn33:44:55:gh})
\end{aligned}$$

The following is the corresponding Cisco MDS rule-set.

```
zone name ExZone vsan 2 member pwwn 00:11:22:ab
zone name ExZone vsan 2 member pwwn 00:11:22:cd
zone name ExZone vsan 2 member fwwn 22:33:44:ef
zone name InZone vsan 2 member pwwn 11:22:33:cd
zone name InZone vsan 2 member fwwn 22:33:44:ef
zone name InZone vsan 2 member fwwn 33:44:55:gh
```

1) Read-Only Zones: Within a SAN, VSAN or zone, initiators by default have read/write access to their permitted target storage arrays. Switches, for example the Cisco MDS series, have the ability to restrict read/write access to a storage array at a switch-level by explicitly defining *read-only* zones. Note, while modern storage arrays provide their own access-controls with respect to read/write access, it is considered best practice to adopt a security in depth approach [15].

Concept $ZoneRule$ is therefore extended to include details ($isReadOnly$ property) about read/write access within a zone.

$$\begin{aligned}
\textit{ZoneRule} \sqsubseteq\ & \textit{ConfigRule} \sqcap \\
& \exists_{=1} \textit{isReadOnly.Boolean}
\end{aligned}$$

The SAN security requirement outlined in Section IV-D may be further refined such that: “*External research partner nodes are permitted* ***read-only*** *access to the research storage array* ...”. This will require the following Cisco MDS rule in addition to the previously defined external zone rule-set.

```
zone name ExZone attribute read-only
```

Individual exZone, representative of the external zone security requirement is updated to include detail that defines the external zone as read-only. Note ‘...’ refers to the original individual assertion.

$$\textit{ZoneRule}(\texttt{exZone}) \leftarrow \ldots \sqcap\ \textit{isReadOnly}(\texttt{exZone}, \texttt{true})$$

2) Lun-Zoning: A storage array is composed of a number of disks or LUNs. By default any initiator node within the same zone as the target storage array has access to all of its LUNs. *LUN-Zoning* is an access control mechanism that restricts access to a storage array on a per-LUN basis [12]. Note, while modern storage arrays provide their own access-controls with respect to LUN access (*LUN Masking* [16]), it is best practice to consider a security in depth approach [15].

The SAN security requirement outlined in Section IV-D1 may be further refined such that: “*External research partner nodes are permitted read-only access to* ***specific LUNs*** *within the research storage array* ...”.

Consider the following scenario. External research partners identified as initA and initB both require read access to a specific LUN (LUN identifier 0X63) where both partners collaborate with respect to the data stored on this LUN. However, external research partner initB requires read access to another LUN (LUN identifier 0X64) such that external research partner initA has no collaboration. The following low-level Cisco MDS rules uphold these requirements.

```
zone name ExZone vsan 2 member pwwn 00:11:22:ab lun 0X63
zone name ExZone vsan 2 member pwwn 00:11:22:cd lun 0X63
zone name ExZone vsan 2 member pwwn 00:11:22:cd lun 0X64
zone name ExZone vsan 2 member fwwn 22:33:44:ef
zone name ExZone attribute read-only
```

Individual exZone defined in Section IV-D1 remains the same. However, the individuals of concept $Member$ are encoded with additional knowledge that ensures external initiators initA and initB are permitted read-only access to their respective LUNs ($hasLUNID$ property relationship) within the research storage array (resSA).

$$\begin{aligned}
\textit{Member}(\texttt{initA}) \leftarrow\ & \ldots \sqcap \textit{hasLUNID}(\texttt{initA}, \texttt{0X63}) \\
\textit{Member}(\texttt{initB}) \leftarrow\ & \ldots \sqcap \textit{hasLUNID}(\texttt{initB}, \texttt{0X63}) \sqcap \\
& \textit{hasLUNID}(\texttt{initB}, \texttt{0X64})
\end{aligned}$$

V. Structural Conflict Analysis Definitions

Structural Analysis examines the relationship that rules have with one another within a security configuration or across multiple inter-dependent security configurations [3], [8], [9]. This paper extends the existing work on firewall structural analysis techniques and considers inter-configuration conflicts that may occur between a firewall and a zone security configuration.

For ease of exposition, the relevant properties are described using a basic discrete math notation. These properties have been implemented in SWRL to perform structural analysis and an example is provided in Section VI.

A. SAN Rule Composition

In the context of inter-configuration conflicts between a firewall and a zone configuration, a SAN access-control rule is defined as a tuple.

$$\textit{Rule} \equiv \textit{Condition} \times \textit{Action}$$

Given $r : Rule$, rule r is a tuple (r_1,r_2) where $r_1 \in Condition$ and $r_2 \in Action$. For simplicity, and when no ambiguity arises, $r.Condition$ is a syntactic sugar for r_1 and so forth.
Filter Conditions. The set of filter conditions ($Condition$) that describes the common filter conditions between firewall and zone access-control rules is defined as an 2-tuple.

$$Condition \equiv SrcIP \times DstIP$$

The source and destination IP addresses (*SrcIP* and *DstIP*) are defined as an inclusive interval, where an interval is a subset of all possible IP addresses for source and destination IP addresses.

$$SrcIP,DstIP \equiv [0, 2^{32}-1]$$

Given *r:Rule*, then $r.Condition_1 \in SrcIP$ and $r.Condition_2 \in DstIP$. For simplicity, and when no ambiguity arises, $r.SrcIP$ is a syntactic sugar for $r.Condition_1$ and so forth. For example, $r.SrcIP$ is used to access the attribute value of set $SrcIP$, instead of $r.Condition_1$ (or r_{1_1}).
Action. The set of target actions are:

$$Action \equiv \{\texttt{permit}, \texttt{deny}, \texttt{log\&deny}\}$$

B. Subsumption Filter Conditions

The filter conditions of rule s ($s.Condition$) subsumes those of rule r ($r.Condition$), if every filter condition field of r, for example $r.SrcIP$, is equal to or is a subset of the corresponding filter condition fields of rule s, for example $s.SrcIP$. Consider rules $r, s{:}Rule$ with identical filter conditions except for their source IP address ranges; $r.SrcIP = \{\texttt{10.37.2.12}\}$ and $s.SrcIP = \{\texttt{10.37.2.10}, \ldots, \texttt{10.37.2.15}\}$, then $r.Condition$ is subsumed by $s.Condition$ if $r.SrcIP \subseteq s.SrcIP$ holds. Formally, given rules $r, s : Rule$, then $r.Condition$ is subsumed ($\subseteq_{cond}$) by $s.Condition$ if and only if $r.Condition \subseteq_{cond} s.Condition$ holds across each of the corresponding filter condition fields.

$$\begin{aligned} r.Condition \subseteq_{cond} s.Condition \equiv\ & r.SrcIP \subseteq s.SrcIP \wedge \\ & r.DstIP \subseteq s.DstIP \end{aligned}$$

C. Inter SAN Rule conflicts

Inter-Shadowed Conflict. Given rules $r_{zone}, s_{fw}{:}Rule$, rule r_{zone} (zone access-control rule) is shadowed by rule s_{fw} (firewall access-control rule), if the filter conditions of r_{zone} are a subset of or equal to the corresponding filter conditions of rule s_{fw}, where s_{fw} is denying what rule r_{zone} is intending to `permit`.

$$\begin{aligned} & r_{zone}.Condition \subseteq_{cond} s_{fw}.Condition \wedge \\ & r_{zone}.Action = \texttt{permit} \wedge \\ & (s_{fw}.Action = \texttt{deny} \vee s_{fw}.Action = \texttt{log\&deny}) \end{aligned}$$

Note, zone rules only have an `permit` action and therefore do not cause a shadowing conflict with firewall rules.

Inter-Spurious Conflict. Given rules $r_{zone}, s_{fw}{:}Rule$, rule s_{fw} is spurious to rule r_{zone}, if the filter conditions of r_{zone} is a subset of the corresponding filter conditions of rule s_{fw}, where s_{fw} is allowing more than what rule r_{zone} is intending.

$$\begin{aligned} & r_{zone}.Condition \subset_{cond} s_{fw}.Condition \wedge \\ & r_{zone}.Action = \texttt{permit} \wedge s_{fw}.Action = \texttt{permit}) \end{aligned}$$

Note, zone rules are not defined over IP address ranges and therefore do not cause spurious conflicts with firewall rules.

VI. Analysis of Access Control Configuration

A. Structural Analysis

Firewall centric structural analysis techniques, for example [3], [8], [9], are also applicable to firewall-based SAN access-control rules. The following SWRL rule extends these works and detects inter-shadowing conflicts between firewall access-control rules (SWRL variable *?fwr*) and zone access-control rules (SWRL variable *?zr*). Relevant filter conditions of each firewall access-control rule (lines 2-5) and each zone access control rule (lines 8-12) are then examined against the subsumption relation (lines 14-17) defined in Section V-B.

IPAccessListRule(?fwr)$\wedge$ (1)
hasSrcIPStart(?fwr,?sIPS)$\wedge$*hasIPValue(?sIPS,?sIPSV)*$\wedge$ (2)
hasSrcIPEnd(?fwr,?sIPE)$\wedge$*hasIPValue(?sIPE,?sIPEV)*$\wedge$ (3)
hasDstIPStart(?fwr,?dIPS)$\wedge$*hasIPValue(?dIPS,?dIPSV)*$\wedge$(4)
hasDstIPEnd(?fwr,?dIPE)$\wedge$*hasIPValue(?dIPE,?dIPEV)*$\wedge$(5)
hasAction(?fwr,`deny`*)*$\wedge$ (6)
ZoneRule(?zr)$\wedge$ (7)
hasMember(?zr,?m1) $\wedge$ *hasMember(?zr,?m2)*$\wedge$ (8)
hasIPStart(?m1,?ipS1)$\wedge$*hasIPValue(?ipS1,?ipSV1)*$\wedge$ (9)
hasIPEnd(?m1,?ipE1)$\wedge$*hasIPValue(?ipE1,?ipEV1)*$\wedge$ (10)
hasIPStart(?m2,?ipS2)$\wedge$*hasIPValue(?ipS2,?ipSV2)*$\wedge$ (11)
hasIPEnd(?m2,?ipE2)$\wedge$*hasIPValue(?ipE2,?ipEV2)*$\wedge$ (12)
differentFrom(?m1,?m2)$\wedge$ (13)
swrlb:greaterThanOrEqual(?ipSV1,?sIPSV)$\wedge$ (14)
swrlb:lessThanOrEqual(?ipEV1,?sIPEV)$\wedge$ (15)
swrlb:greaterThanOrEqual(?ipSV2,?dIPSV)$\wedge$ (16)
swrlb:lessThanOrEqual(?ipEV2,?dIPEV)$\wedge$ (17)
$\rightarrow$ *isInterShadowedBy(?zr,?fwr)* (18)

Zone members may also be identified by their WWNs. Firewall access-control rules do not consider WWN's. As a consequence, a relationship (*hasWWNIPMapping*) between each zone member's WWN and its corresponding IP address must be asserted or inferred (for example using SWRL).

B. Query Analysis

Query Analysis provides a way to ask hypothetical 'what-if' questions of a security configuration [3]. A switch can be analysed to check whether or not the answers to queries made of its configuration are consistent with the enterprise-level

security requirements. For example the following SQWRL query asks: "*What role or roles are users authorised for?*"

$$\textit{User}(?user) \wedge \textit{RoleRule}(?role) \wedge \textit{isRoleOf}(?role, ?user) \rightarrow \textit{sqwrl:select}(?user, ?role)$$

The following SQWRL query asks what initiators (members) have been assigned read-only access to what disks (identified by LUN ID's) within a given zone defined in the context of a particular VSAN?

$$\begin{aligned}&\textit{Member}(?mem) \wedge \textit{ZoneRule}(?zone) \wedge \textit{VSANRule}(?vsan) \wedge \\ &\textit{hasName}(?zone, ?zname) \wedge \textit{hasName}(?vsan, ?vname) \wedge \\ &\textit{isReadOnly}(?zone, \texttt{true}) \wedge \textit{hasLUNID}(?mem, ?lunid) \wedge \\ &\textit{isAssignedTo}(?zone, ?vsan) \wedge \textit{hasMember}(?zone, ?mem) \\ &\rightarrow \textit{sqwrl:select}(?mem, ?lunid, ?zname, ?vname) \wedge \\ &\textit{sqwrl:columnNames}(\text{"}\textit{Initiator}\text{"}, \text{"}\textit{Disk}\text{"}, \text{"}\textit{Zone}\text{"}, \text{"}\textit{VSAN}\text{"})\end{aligned}$$

VII. Related Research

While there is extensive research on synthesis [17], [18], query [6], [7] and structural [8], [9] analysis, these firewall-centric works do not consider switch security configuration. This paper builds on previous firewall centric research [3]–[5] to model and reason about SAN switch security configurations.

VIII. Discussion and Conclusion

This paper considered the problem of reasoning about and managing complex switch security configurations. A formal model for switch security configuration was presented. Structural analysis definitions that define inter-shadowing and inter-spurious conflicts between firewall-based and zone-based switch security configurations where presented.

Future work will explore additional structural analysis techniques that involve other switch security mechanisms. For example, role-based access-control rules are order dependent. Depending on the sequence of these rules, a misconfiguration may result. An exploration of possible conflicts between individually consistent VSAN configurations due to *Inter-VSAN Routing* [12] will be also be investigated.

Future work will adopt the threat-based approach in [4] to structure knowledge about switch security configurations in terms of threats with which to construct a catalogue of best practice countermeasures. One may then, in conjunction to structural and query analysis described in this paper, automatically generate suitable switch security configurations (countermeasures) that mitigate known SAN threats.

This threat-based approach will also facilitate the construction of a best practice catalogue of candidate queries. Thus, providing inexperienced security administrators with the ability ask expert questions about the effectiveness of an existing switch security configuration. These candidate queries could be based on testing for best practice compliance. For example, to test if a switch security configuration is SNAI [19] compliant, the security administrator can draw upon a catalogue of SNAI candidate compliance queries.

Acknowledgment

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

References

[1] J. Somasundaram and A. Shrivastava, *Information Storage and Management: Storing, Managing, and Protecting Digital Information*. Wiley, 2009.

[2] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, March 2003.

[3] W. M. Fitzgerald, "An Ontology Engineering Approach to Network Access Control Configuration," *PhD Thesis, University College Cork, Ireland*, August 2010.

[4] 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.

[5] 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.

[6] A. Mayer, A. Wool, and E. Ziskind, "Offline Firewall Analysis," *International Journal of Information Security*, vol. 5, no. 3, pp. 125–144, May 2006.

[7] R. Marmorstein and P. Kearns, "A Tool for Automated iptables Firewall Analysis," *Usenix Annual Technical Conference, Freenix Track, Pages: 71-81, Anaheim, CA, USA*, April 2005.

[8] E. S. Al-Shaer, H. H. Hamed, R. Boutaba, and M. Hasan, "Conflict Classification and Analysis of Distributed Firewall Policies," *IEEE Journal on Selected Areas in Communications, Issue: 10, Volume: 23, Pages: 2069 - 2084*, October 2005.

[9] F. Cuppens, N. Cuppens-Boulahia, and J. García-Alfaro, "Detection and Removal of Firewall Misconfiguration," *IASTED International Conference on Communication, Network and Information Security (CNIS), Phoenix, AZ, USA*, November 2005.

[10] "IANA Port Numbers," http://www.iana.org/assignments/port-numbers.

[11] 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 (ISWC2005), Galway, Ireland*, 2005.

[12] *Cisco MDS 9000 Family CLI Configuration Guide, Release 4.x.*, Cisco, February 2009.

[13] *Cisco MDS 9000 Family Command Reference, Release 5.x.*, Cisco, February 2010.

[14] K. Scarfone and P. Hoffman, "Guidelines on Firewalls and Firewall Policy: Recommendations of the National Institute of Standards and Technology," *NIST Special Publication 800-41, Revision 1*, September 2009.

[15] 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.

[16] J. Tate, F. Lucchese, and R. Moore, *Introduction to Storage Area Networks*. CIBM Redbooks, 2006.

[17] F. Cuppens, N. Cuppens-Boulahia, T. Sans, and A. Miège, "A Formal Approach to Specify and Deploy a Network Security Policy," *2nd Workshop on Formal Aspects in Security and Trust (FAST), Toulouse, France*, August 2004.

[18] M. G. Gouda and X.-Y. A. Liu, "Firewall Design: Consistency, Completeness and Compactness," *24th IEEE International Conference on Distributed Computing Systems (ICDCS), Japan*, March 2004.

[19] *Storage Security Best Current Practices (BCPs), Version 2.1.0*, Storage Networking Industry Association (SNIA), September 2008.