Firewall Automatic Script Configuration – a critical review

R. Alkareem, D Veal, S P Maj

Edith Cowan University, Perth, Western Australia

Summary
It is recognized that firewall configuration is complex. This is especially the case when firewalls are configured using the text based command line interface. This is potentially problematic because an incorrectly configured firewall represents a major security threat to an organization. To address this problem a Human Computer Interface (HCI) can be employed. Whilst use of a HCI considerably simplifies firewall configuration it can potentially increase the problems associated with firewall maintenance and management. This is because the HCI automatically generates configuration code that may be not only complex but also extensive. In this context should a configuration problem occur it may well be problematic for the network administrator to effectively troubleshoot configuration code. Whilst the HCI may provide additional interface options

Key words:
Firewall, network security, GUI

1. Firewall configuration

Firewalls are an essential component of corporate security and as such must be correctly configured and appropriately managed. According to Rubin, the most important factor of firewall security is how to configure it [1]. However, an analysis of corporate firewalls indicated that many firewalls were misconfigured [2]. According to Wool,

> *Cisco's approach is typical of most firewall vendors: it exposes the raw and confusing direction-based filtering functionality to the firewall administrators. Other vendors that follow the same approach (with different syntactic mechanisms) include, among others, Lucent, NetScreen and open-source tools such as ipchains and netfilte*r. [3]

Configuration can be considered as a human factor in security [4]. As such use-ability is of paramount importance and may be improved by means of a Human Computer Interface (HCI). It should be noted that a HCI is synonymously also referred to as a Graphical User Interface (GUI).

There are HCI design and evaluation criteria. Nielson identified ten criteria that include:

1. Visibility of system status
2. Match between system and real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalistic design
9. Help users recognize, diagnose and recover from errors
10. Help and documentation [5]

Given the importance of security, secure HCI (HCI-S) guidelines have been proposed. They include:

1. Convey features
2. Visibility of system status
3. Learnability
4. Aesthetic and minimalistic design
5. Errors
6. Satisfaction
7. Trust [6]

It is considered that a system that is easier to use will result in fewer errors. This in turn may lead to a more secure system [7].

2. Security Device Manager

In order to simplify secure device configuration Cisco, the world's largest supplier of network equipment, introduced the Security Device Manager (SDM). The initial SDM interface provides network device details (hardware specification) and software details (Internetwork Operation System version), along with the device configuration overview (Figure 1).

Selecting the configuration tab it is possible to deploy a wide range of security measures such as: Virtual Private Networks (VPNs), security audits, Network Address Translation (NAT), Intrusion Prevention, Authentication, Authorization and Accounting (AAA) etc – all by means of a HCI (Figure 2).

Manuscript received July 5, 2011
Manuscript revised July 20, 2011

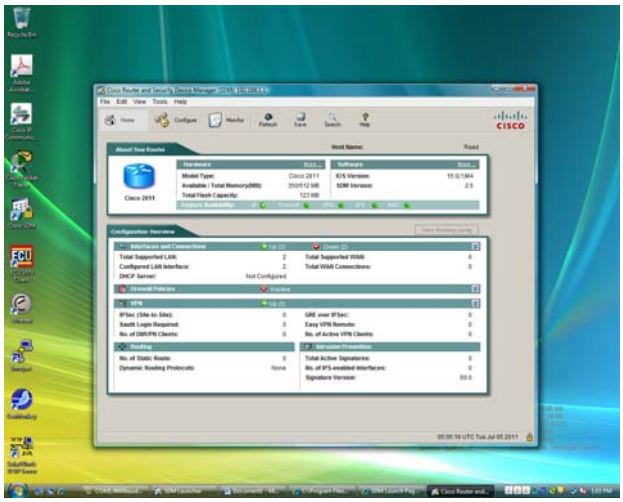

Figure 1. Security Device Manager

Selecting the firewall and ACL tab, SDM allows the network administrator to create a firewall and also edit a firewall policy (Figure 2).

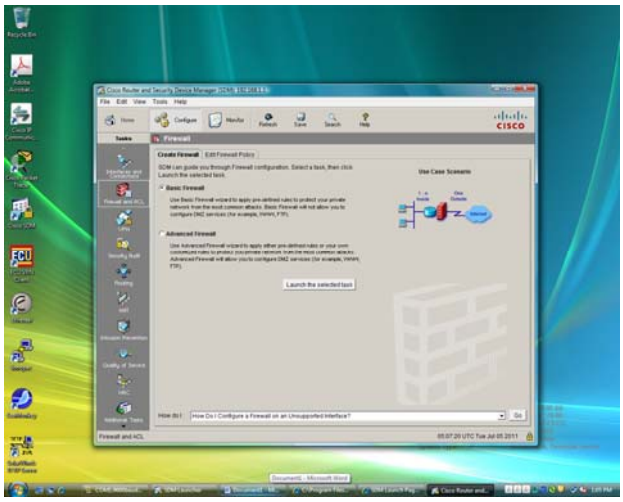

Figure 2. SDM Basic/advanced firewall wizards

SDM allows the user to select either a basic firewall or an advanced firewall wizards. The basic firewall applies pre-defined rules to protect a network from the most common attacks. By contrast the advanced firewall wizard can be used to apply either pre-defined rules or customized rules to protect a network from attack. Furthermore it is possible to select between High, Medium and Low security (Figure 3).

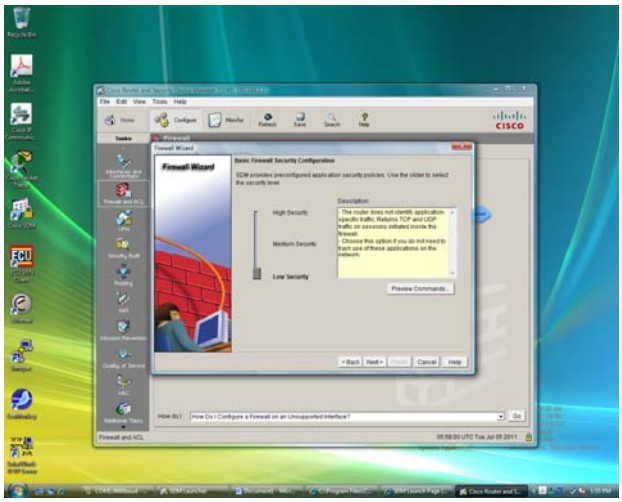

Figure 3. SDM High, Medium and Low security

Regardless of which options are chosen, at the conclusion of the wizard configuration code is copied to the router's running configuration and hence is actively deployed.

3. Configuration code

If the low security option is selected approximately eight pages of configuration code is generated consisting of the following security implementations:

- Parameter-map type protocol-info
- Crypto pki details
- Class map types
- Policy map types
- Zone security
- Access control lists

The class map type output consists of over two pages of configuration code, a partial extract of which is as follows:

class-map type inspect imap match-any sdm-app-imap
match invalid-command
class-map type inspect match-any sdm-cls-protocol-p2p
match protocol edonkey signature
match protocol gnutella signature
match protocol kazaa2 signature
match protocol fasttrack signature
match protocol bittorrent signature
class-map type inspect match-any sdm-cls-insp-traffic

match protocol cuseeme
match protocol dns
match protocol ftp
match protocol h323
match protocol https
match protocol icmp
match protocol imap
match protocol pop3
match protocol netshow
match protocol shell
match protocol realmedia
match protocol rtsp
match protocol smtp
match protocol sql-net
match protocol streamworks
match protocol tftp
match protocol vdolive
match protocol tcp
match protocol udp
class-map type inspect match-all sdm-insp-traffic
match class-map sdm-cls-insp-traffic
etc

Compared to entering every command manually using the text based command line interface the development of firewall configuration code using the SDM HCI is certainly rapid. However this is potentially problematic because the administrator may not be aware of the exact functionality of every line of code. In this context fault diagnosis could prove to be very difficult.

By contrast, manually entering configuration code is certainly considerably slower. However this method of firewall configuration is normally based on the administrator doing so with some understanding of each line of code.

The SDM provide a firewall editing page (Figure 4).

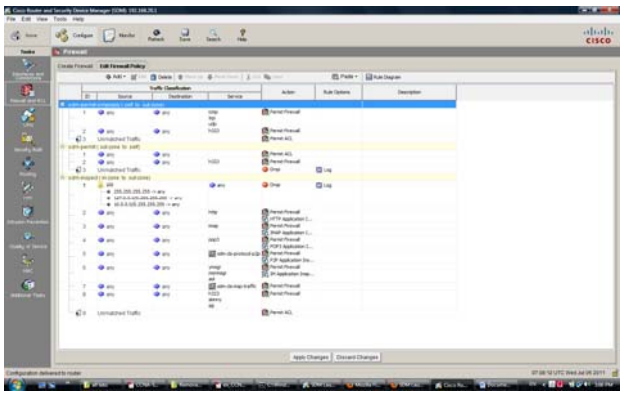

Figure 4. SDM Basic Firewall editing page

This editing page contextualizes much of the configuration code and assists to some degree configuration code management. This is a hierarchically arranged rule diagram listing with associated source, destination, service, action and rule options. Using the editing page changes can be developed and deployed.

In addition to this SDM provides a firewall status monitor tab allowing the user to select the firewall policy and in real-time (default every 10 seconds) to observe dropped and allowed packets (Figure 5).

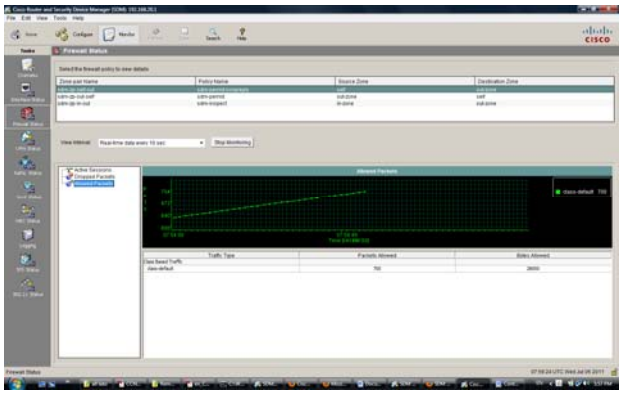

Figure 5. SDM Firewall status monitor

4. Mis-configuratoin analysis

Security threats are based on vulnerabilities. Vulnerabilities are based on either inadequate security policies or the incorrect operational implementation of those policies [8]. Within this context it is important to distinguish between implemented policy and intended policy [9]. A policy mis-configuration can then be defined as an inconsistency between intended and actual implemented policies. Mis-configurations are more likely to occur if it is complex to inspect and verify device configuration code. Whilst the SDM firewall editing page provides a hierarchical arrangement of policies it is never the less not straight forward to identify relationships, for example:

- Service-policy types and the associated contextually higher level zone pair-security.
- Service-policy with the associated policy-map type
- Class-types with the associated class-maps

This chain of relationships is of fundamental importance to ensuring correct device configuration. The authors propose this problem may be address by an alternative mapping

based on the State Model Diagram method [10-12]. This modeling method has been successfully used to model a wide range of security devices and protocols. A key feature of this method is the maintenance of context by means of hierarchically arranged tables. This can be demonstrated as follows. Tables 1 and 2 define in-zone and out-zone mappings.

Interface	Fa0/1
Zone-member security	In-zone

Table 1. Interface fa0/1

Interface	Fa0/0
Zone-member security	Out-zone

Table 2. Interface fa0/0

These mappings are employed in the definition of zone-pair security (Sdm-zp-in-out) with the associated service policy type ispect (Sdm-inspect) (table 3).

Zone-pair security	**Sdm-zp-in-out**
Source	In-zone
Destination	Out-zone
Service policy type inspect	**Sdm-inspect**

Table 3. Zone-pair security

Details of Sdm-inspect are defined in table 4 along with the associated class types.

Policy-map type inspect	**Sdm-inspect**
Class type inspect	Sdm-invalid-src
	Drop log
Class type inspect	**Sdm-insp-traffic**
	inspect

Table 4. Policy-map type

Class types details can be displayed by two further tables (table 5, table 6).

Class-map type inspect	
Match-all	**Sdm-insp-traffic**
Match class-map	**Sdm-cls-insp-traffic**

Table 5. Class-map

Class-map type inspect	
Match-any	**Sdm-cls-insp-traffic**
Match protocol	udp

Table 6. Class map

Significantly it is possible to observer all these interrelationships on a single screen. It is of course possible to define multiple zone-pairs with the associated table mappings.

Conclusions

The use of a well designed HCI can considerably simplify the configuration of a firewall. The Security Device Manager provides the user with clearly defined options and the automatic generation and deployment of configuration code. However, the main problem is that a considerable amount of configuration code can be generated. The SDM allows the user to edit firewall policies by means of a series of HCI interfaces however there is considerable scope for improving how this information is displayed. This is important in order to reduce the possibilities of mis-configuration. However when firewall configuration details are modeled using State Model Diagrams it is possible to clearly see the inter-relationships between: zone defined interfaces; traffic classes; policies; zone-pair security; parameters etc.

References

[1] Rubin, A.D., Geer, D., Ranum, M. J., *Web Security Sourcebook*. 1997: John Wiley & Sons.

[2] Wool, A., *How not to configure a firewall: a field guide to common firewall misconfigurations*, in *15th USENIX System Administrators Conference (LISA)*. 2001.

[3] Wool, A., *The use and usability of direction-based filtering in firewalls.* Computers & Security, 2004. **23**: p. 459-468.

[4] Schultz, E., *The Human Factor in Security.* Computers & Security, 2005. **45**: p. 425-426.

[5] Nielsen, J. *Ten Usability Heuritics*. December, 2008]; Available from: http://www.useit.com/papers/heuristic/heuristic_list.html.

[6] Johnston, J., J.H.P. Eloff, and L. Labuschagne, *Security and human computer interfaces.* Computers & Security, 2003. **22**(8): p. 675-684.

[7] Furnell, S., *Making security usable: Are things improving?* Computers & Security, 2007. **26**: p. 434-443.

[8] Benantar, M., *Access Control Systems: Security, Identity Management and Trust Models*. 2005, Secaucus, NJ, USA: Springer-Verlag.

[9] Bauer, L., Garriss, S., Reiter, M., K. *Detecting and resolving policy misconfigurations in access-control systems*. in *13th ACM Symposium on Access Control Models and Technologies (SACMAT 2008)*. 2008. Colorado, USA: ACM/SIGSAC.

[10] Maj, S.P., Makasiranondh, W., Veal, D., *An Evaluation of Firewall Configuration Methods.* International Journal of Computer Science and Network Security, 2010. **10**(8): p. 1-7.

[11] Maj, S.P., Veal, D., *An Evaluation of State Model Diagrams for Secure Network Configuration and Management.* International Journal of Computer Science and Network Security, 2010. **10**(9): p. 66-72.

[12] Maj, S.P., Veal, D., *Using State Model Diagrams to Manage Secure Layer 2 Switches.* International Journal of Computer Science and Network Security, 2010. **10**(9): p. 141-144.

Raad Alkareem has completed his Master of Information Technology at Edith Cowan University, also completed his Bachelor of Networking Technology at Edith Cowan University, he also have completed two diplomas at WA TAFE, one for Networking technology and one in Administration. His area of interest is in Networking Security.

Dr. David Veal is a Senior Lecturer at Edith Cowan University. He is the manager of Cisco Network Academy Program at Edith Cowan University and be a unit coordinator of all Cisco network technology units. His research interests are in Graphical User Interface for the visually handicapped and also computer network modeling.

A/Prof S. P. Maj has been highly successful in linking applied research with curriculum development. In 2000 he was nominated ECU University Research Leader of the Year award He was awarded an ECU Vice-Chancellor's Excellence in Teaching Award in 2002, and again in 2009. He received a National Carrick Citation in 2006 for "*the development of world class curriculum and the design and implementation of associated world-class network teaching laboratories*".