Firewall Rulebase Analysis Tool

P. Jain and P.S. Dowland

Centre for Security, Communications and Network Research
Plymouth University, United Kingdom
e-mail: info@cscan.org

Abstract

Research has shown that majority of network penetration attacks is likely because of poor implementation of firewalls; which are actually meant to protect the network at perimeter. The Firewall Rulebase Analysis Tool analyses the inapt firewall rules and helps you defend against penetration attacks. The aim of this project is to highlight that the offline rulebase analysis has more to offer and should be considered in cost-cutting measures and by SMEs. This tool does an intensive analysis on each rule against a pre-defined checklist, and generates a report mentioning necessary actions required.

Keywords

Firewall rules, rulebase analysis, rulebase automation tool, Netscreen, Cisco ASA

1 Introduction

Firewalls are meant to protect the network by analyzing traffic packets against pre-defined rulesets, and thus its significant implementation is extremely crucial. This includes physical configuration, location in network architecture, and managing rules within. A single inappropriate rule is enough to provide an entry point for hackers to penetrate the network.

These rules are created by firewall administrator based on whitelist or blacklist pattern to allow/deny traffic. The objective of having such rules is to create a bottleneck for only authorized packets to enter the network and block all other unnecessary traffic.

With such critical job, managing firewalls is equally critical. Large organizations have multiple firewalls and large rulebase. Firewall management products from leading vendors do real-time analysis with combination of logs and firewall rules. However, such products are heavy and come at a costly price, which SMEs cannot afford or don't actually need. SME's have mostly 1 or 2 firewalls and comparatively less rules within.

The objective of this tool is take a passive approach and use the configuration file to perform rulebase analysis. Configuration file is a file that has entire settings related to the device; from users' passwords (can be masked or unmasked), to hardware configuration settings and all other working parameters. This will help the firewall administrators, and also security auditors to assess their rulebase configuration. This will help them save the cost of integrating firewall management suites and at the same time help out with compliance audits.

In this paper, we look at the overview of existing products and its limitations; the developed tool’s working features, tool-generated reporting structure and tool evaluation.

2 An overview of existing products

Tools like Algosec’s firewall analyzer, RedSeal’s Network Advisor are active tools that need to be integrated with the firewalls. “AlgoSec supports firewall policy management, including the automation of firewall operations, auditing and compliance, change management, and risk analysis”. (Algosec, n.d.). From policy management perspective, we need to perform the similar task using configuration file, an offline approach.

There are tools like NII’s Firesec, 360 Anaytics’ 360-FAAR, and earlier versions of Nipper. However, Nipper and 360-FAAR only helped in interpreting configuration file and present it in a more readable manner. Firesec is the only tool that does some offline analysis, but the checks are very limited as compared to active tools and the reporting format is not user-appealing. The report does not give any specific reason for marking a rule as ‘Unsafe’. In such scenario, the user is left clueless on the modifications required in the rulebase and the next step to be taken.

The need of the hour is to have an interactive report, additional number of checks, further granular analysis to avoid false positives, highlighting unsafe rules with proper analysis comments, and reduce manual effort. This will help users prepare for compliance standard requirements.

3 Firewall Rulebase Analysis Tool

The tool developed is on the grounds of passive analysis and so it is important to understand the basic rule structure. Following this, the tool working and report will be discussed.

3.1 Basics of Rulebase Analysis and the approach used

Before going ahead with rulebase analysis, it is important to understand the basic structure of a ‘rule’. Basically, a rule is a combination of source, destination and service. However, it has some more elements:

Rule no	Rule name	Source	Destination	Service	Allow status	Protocol	Logging

Rule no: Each rule may be associated with a rule number or rule id for reference. This is an optional field, as what matters is the position of the rule. Generally the preceding rule has higher priority unless there is a global policy set.

Rule name: This is used to give information about the interface and the direction of traffic (Inbound or Outbound), the rule is applied for. The approach may vary, but ultimately the information provided is the same.

Source, Destination: Address objects or Group Address objects which are binded with IP address(es)

Service: It defines Service objects or Group Service objects with port numbers/services (e.g. ‘port 53’ or ‘service DNS’)

Protocol: This field mentions the IP protocol number/name which determines the nature of the traffic (e.g. tcp/udp/icmp/ip)

Allow status: The action to be taken, if a packet matches the rule. It can be ‘allow’ or ‘deny’

Logging: Irrespective of the action taken, this field defines whether the packet details that match the rule should be logged or not. Each rule will have to specify this option individually. The logs can be stored locally in a log file or in a central syslog server.

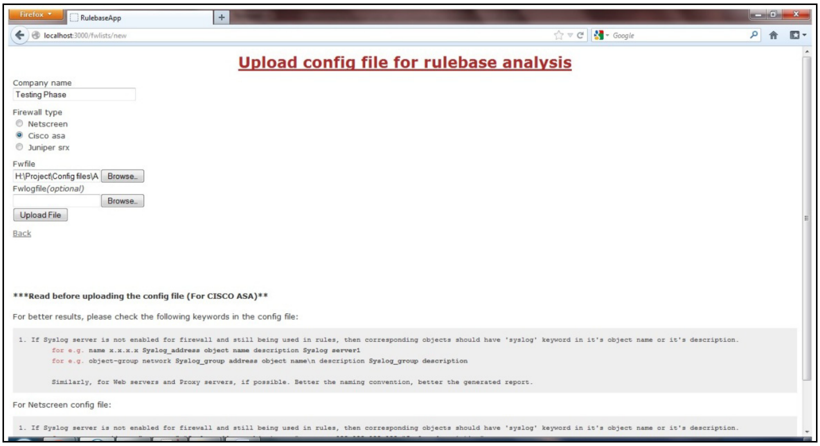
Since, a rule is associated with interfaces, address objects, and service objects; all this data along with rules also need to be collected from the configuration file. If logging is enabled, then generated logs will be useful to count the number of hits on a particular rule, which proves useful.

Thus, a checklist was created that would assess different unsafe patterns of rule settings, which will be discussed in section 4. After surveying the demand for leading firewall vendor products, Juniper’s “Netscreen” and Cisco’s “ASA” firewalls were chosen to be used as experiment models for the tool. The tool is developed using ‘Ruby on Rails’ framework.

3.2 Tool working and its features

The target users of this tool are firewall administrators of SMEs and security auditors. Focusing the criticality of data within configuration file, this tool will have to be installed locally, rather than on internet. Once the tool is installed, and rails server is started; typing ‘localhost:3000’ in the browser URL will load up the application. 3000 is the default port number used by rails server. This will show a page, indexing all the previous uploaded configuration files and associated form details. Click on ‘New Fwlist’ to upload a new file.

Step 1: Fill the form details and upload the config file and log file (optional). Only one check depends on log file and the user may not want to upload such a huge file (size can be in MBs).



Upload config file for rulebase analysis

Company name
Testing Phase

Firewall type
○ Netscreen
● Cisco asa
○ Juniper srx

Fwfile
H:\Project\Config files\A Browse..
Fwlogfile*(optional)*
Browse..
Upload File

Back

*****Read before uploading the config file (For CISCO ASA)****

For better results, please check the following keywords in the config file:

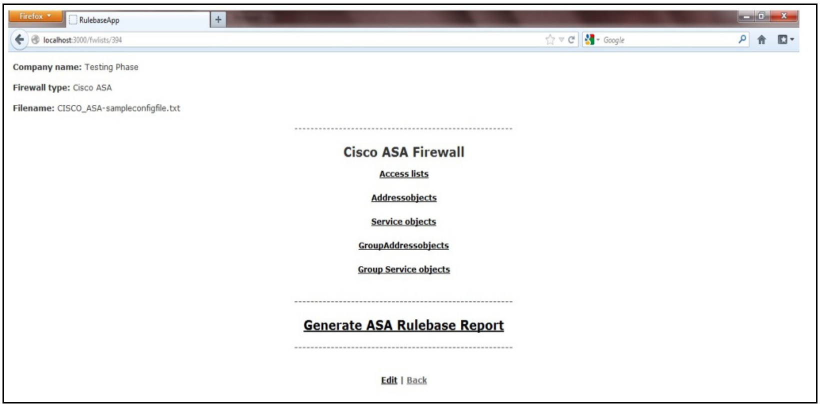
```
1. If Syslog server is not enabled for firewall and still being used in rules, then corresponding objects should have 'syslog' keyword in it's object name or it's description.
       for e.g. name x.x.x.x Syslog_address object name description Syslog server1
       for e.g. object-group network Syslog_group address object name\n description Syslog_group description

       Similarly, for Web servers and Proxy servers, if possible. Better the naming convention, better the generated report.
```

For Netscreen config file:

```
1. If Syslog server is not enabled for firewall and still being used in rules, then corresponding objects should have 'syslog' keyword in it's object name or it's description.
```

Figure 1: Form to upload new file

Step 2: Clicking on Upload file and the Firewall Type chosen, scripts run in the background to parse the data from the uploaded files and store it in database tables. In this case, it's Cisco ASA.



Company name: Testing Phase

Firewall type: Cisco ASA

Filename: CISCO_ASA-sampleconfigfile.txt

Cisco ASA Firewall

Access lists

Addressobjects

Service objects

GroupAddressobjects

Group Service objects

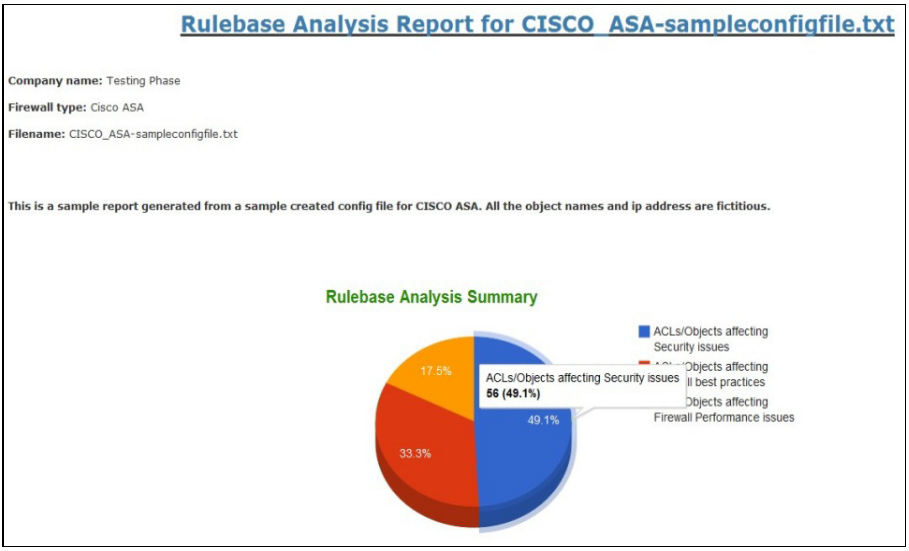
Generate ASA Rulebase Report

Edit | Back

Figure 2: List of database tables for uploaded file and link to generate report

Step 3: Cross check if all the tables; "Accesslists", "Service Objects", "Address Objects" and "Group Objects" have correct data. If not, then edit/delete options are provided for each row.

Step 4: Click on 'Generate ASA report'. This will run scripts for all rulebase checks on the above tables and generate an evaluation report. Each section of the report is discussed in SECTION 4 along with the checklist.



Rulebase Analysis Report for CISCO_ASA-sampleconfigfile.txt

Company name: Testing Phase

Firewall type: Cisco ASA

Filename: CISCO_ASA-sampleconfigfile.txt

This is a sample report generated from a sample created config file for CISCO ASA. All the object names and ip address are fictitious.

Rulebase Analysis Summary
ACLs/Objects affecting Security issues
Objects affecting
ll best practices
Objects affecting
Firewall Performance issues
ACLs/Objects affecting Security issues
56 (49.1%)
17.5%
33.3%
49.1%

Figure 3: Tool-generated report

3.3 Tool-generated Report structure

After researching through firewall management products, PCI DSS standards and NIST considerations, a checklist was prepared, which could be used to determine inapt rules using configuration file. These checks were then converted to scripts, with the intention of reducing manual effort and present a sensible and valuable report. To be more interactive, these ruby scripts are integrated with rails framework so the tool has a User Interface and the user could use it without any knowledge of scripts or commands.

Following are the tests that run against the rules/objects of the uploaded file. Severity levels determine the criticality of the check (in [] brackets).

Check name	Description
Reverse/ Bidirectional rules [Security]	Reverse rules are the ones in which the source object of one rule is present in destination of another and vice-versa. For stateful firewalls, such access might not be needed, except for some special applications. Once the TCP handshake is done and the state is established, the firewall would refer to the state table for allowing incoming traffic.
Rules allowing cleartext services [Security]	Cleartext protocols send data in clear text, without any encryption. This means that the data sent through these protocols are susceptible to network sniffing attacks. Commonly known clear-text protocols are HTTP, Telnet, IMAP, POP, FTP, and NETBIOS. Avoid using such protocols.
Deny-All-Log rules [Security]	This rule rejects and logs all the traffic patterns not covered in the rules. This gives a whitelist approach, where only required traffic is permitted. With the amount of security, this single rule provides, it is required to have an explicit ‘deny-all’ rule (with

	logging enabled) at the end. If an implicit 'deny-all' rule already exists, then the explicit one would act as an additional layer of defense.
Entire Network access [Security]	Many-a-times, firewall administrators mention 'any' in address objects, normally under pressure, to avoid business interruption. This rule exposes full network for the corresponding firewall interface. Always have specific addresses in the ACLs.
Large Port Range access [Security]	Ports and services may have associated vulnerabilities running. Hackers outside, malicious insiders or compromised devices may try to port scan the network. Unnecessary ports, especially large range of port access should be avoided.
Invalid IP addresses [Security]	A valid IP address format is: **x.x.x.x** (IP address) **x.x.x.x** (Netmask)**; where x<=255.** Any address that does not fit in this format is invalid. Generally, these are typo-errors by administrators. If such address is used in permit rule, may cost business interruption. If present in deny rule, may prove the rule invalid; thereby allowing traffic which meant to be rejected.
Inappropriate access rules [Security]	**Connecting to any DNS server on internet:** DNS service resolves queries for domain names into IP addresses to locate devices on the internet. In order to stay protected from malicious DNS servers on the internet, make sure to connect to a dedicated DNS server instead of 'any'. **Connecting Syslog and Web server to Internet:** Syslog and Web servers are very critical servers in terms of information they hold. They should never be connected to the internet. Such rules will also be reported by the tool. This tool will take different approaches to determine presence of such servers from configuration file.
Access to 127.0.0.1 and 0.0.0.0 IP addresses [Security]	127.0.0.1 is a loopback (or localhost) address and presence of this IP address, would give access from/to all ports bound to loopback interface. '0.0.0.0' is an unspecified address. Presence of this address in destination, gives access to all network interfaces of a device (Network Working group, 2002)
Management Interface access [Security]	Management interface should be isolated from any traffic except management traffic and also the number of management hosts should be limited. Rules are not required to provide management access, as firewall has other features to enable mgmt access. This tool will determine the management interface and list down all the related rules present. The user has to decide on whether the rules are required or not.
Redundant / Shadow rules [Performance]	Redundant rules are the ones, when one rule is a subset of the other rule. If the parent rule exists, then the other rule composing of its child objects with similar access proves to be redundant.
Covered rules [Performance]	If two rules have any two of; source objects, destination objects and service objects in common, then those rules can be merged to form one rule. Lesser the number of rules, easier it is to manage the rulebase.

Duplicate rules [Performance]	If two rules are exactly similar, they get reported here in this section. One of the two rules should be removed.
Unused rules and top 10 used rules [Best Practice]	If logging is enabled, one can use the valuable information to increase firewall performance. Firewall's efficiency can be increased by bringing the most used rules at the top and removing all unused rules from the rulebase.
Unused Objects [Best Practice]	Unused objects are the objects which are created, but not used in any of the rules.
Inactive rules [Best Practice]	These are disabled rules and prove no use of staying in the rulebase.
Orphan rules [Best Practice]	These ACLs are the ones which have obsolete objects present. Sometimes, systems are removed from the network infrastructure, but the corresponding rules are not removed. It is not possible to get the list of such objects from the config file. However, presence of certain objects might create doubts, for e.g. generally, there would be only one external proxy server in the network. The tool will check for keywords 'proxy' or 'proxies' in address objects and it's descriptions, to determine presence of a proxy server. If more than one external proxy server is found, all related rules will be reported.

Table 1: List of rulebase checks with description

Each check is presented with a generic description in the report followed by a list of unsafe rules under that check. Each rule is given an appropriate audit comment and presented with line number of the rule within the configuration file. This information will help user to locate the rule in configuration file and make suitable changes. Moreover, if the user wants to check the effect of changes being made, before applying to the config file, then the user can scroll down to the APPENDIX of the tool-report. The APPENDIX of the tool-report has the list of database tables, as discussed in previous section, which will help to edit values, and the effect can be observed by regenerating the report. When all the checks are completely executed, a graph is presented which gives a statistical analysis on the number of unsafe findings being reported.

Thus, the user is given a complete interactive report with graphs, proper comments, description of each rule check, location of rule in file, and additional rule-edit options to test before implementing changes.

4 Conclusion and Future Scope

The tool built is really useful for firewall administrators and security auditors to assess their rulebase offline. Most of the checks relevant to PCI DSS and NIST standards have been covered. Thus, using this tool will aid in preparing for these standards. This tool is developed for Netscreen and Cisco ASA firewalls, and has been tested for performance up to 1000 rules. However, lesser the number of rules, better it is to manage the rulebase. The tool will help boost user’s confidence in managing firewall configuration.

With the configuration having so much information to offer, a full vulnerability assessment of configuration file should be targeted. At present, the tool assists in compliance, but, in future, the report should itself be a compliance standard report. Moreover, this being a prototype, only 2 firewalls were used for analysis, and in future, the support should be extended for maximum firewalls.

5 References

ALGOSEC (n.d.) *Algosec Security Management Suite* [WWW] Algosec Inc. Available from: http://www.algosec.com/en/products/firewall_analyzer [Accessed 18/1/12].

Network Working Group (2002) *Special-Use IPv4 Addresses* [WWW] The Internet Society. Available from: http://www.ietf.org/rfc/rfc3330.txt [Accessed 20/07/12]