Edith Cowan University
Research Online

Australian Information Security Management Conference

Security Research Institute Conferences

2007

# A Comprehensive Firewall Testing Methodology

Murray Brand
*Edith Cowan University*

Originally published in the Proceedings of 5th Australian Information Security Management Conference, Edith Cowan University, Perth Western Australia, December 4th 2007.
This Conference Proceeding is posted at Research Online.
http://ro.ecu.edu.au/ism/24

# A Comprehensive Firewall Testing Methodology

Murray Brand
School of Computer and Information Science, Edith Cowan University,
Bradford Street, Mt Lawley, Western Australia 6050
mbrand0@student.ecu.edu.au

## Abstract

*This paper proposes an all encompassing test methodology for firewalls. It extends the life cycle model to revisit the major phases of the life cycle after a firewall is in service as foundations for the tests. The focus of the tests is to show that the firewall is, or isn't, still fit for purpose. It also focuses on the traceability between business requirements through to policy, rule sets, physical design, implementation, egress and ingress testing, monitoring and auditing. The guidelines are provided by a Test and Evaluation Master Plan (TEMP). The methodology is very much process driven and in keeping with the Security Systems Engineering Capability Maturity Model (SSE-CMM). This provides multiple advantages, including the capture of configuration errors, results are measurable and repeatable, assurance is developed and it can be used as a roadmap for process improvement. Sample tests are provided in the paper, but act merely as a guideline. It would be expected that the test and evaluation master plan be tailored for any specific organisation.*

**Keywords**

Firewall, test, methodology, TEMP, SSE-CMM.

## INTRODUCTION

It is not simply enough to purchase a firewall from a vendor, install, configure, test and monitor it just because it is considered a best practice to have a firewall. This kind of thinking could easily result in a less than effectual solution. The objective of this paper is to outline an effective methodology to test the firewall from as many perspectives as possible. Testing the firewall goes far beyond conducting penetration tests using the tools hackers would use. Testing must begin with examining business rules, legislation and internal IT requirements. These must be translated into constraints in the form of policy documentation. It should hardly be surprising that business rules change regularly. New business relationships are formed, internal structures are changed, business strategic focus shifts to new markets, technology advances and new legislation is introduced. As a result, policies must be regularly reviewed and updated through a formal and procedural testing process.

Policies must be used as the inputs to the design of the network topology and the development of rule sets. There are various types of firewalls including application gateways, circuit gateways, packet filtering and MAC layer firewalls, all of which operate at different levels in the OSI model. The result is that they perform differently. Hybrid firewalls merge the various types of firewalls into one. The design and implementation must be traceable and testable against the policy documents. Otherwise, the resultant firewall may not meet the requirements and specifications in policy. Rule sets are highly integrated with the firewall topological design and are instrumental in functionality and must be tested as well, especially when rules are introduced or modified. The rule sets must be under configuration control to ensure any changes are validated and are traceable.

Appropriate test plans must be written, reviewed, performed and evaluated to ensure that the firewall complies with policy. Test plans are driven by policy and must test the rule sets and the implementation. Firewall rule anomalies can be detrimental to security and performance and they must be determined and resolved through testing. The firewall must also be tested from an ingress and egress point of view and these tests must be reflected in the firewall logs. A significant component of the test plan should be the expected results. The firewall should behave as a predictable system.

The resultant test report must list the tools used as well as the results. The results of the test can then used to modify policy, rules and design if required.

The entire process could be quite complex in a large network, especially where artefacts of the policy creation process, design decisions, purchasing assessments, test plan development and test results must be documented. Traceability of these artefacts can be managed though the use of systems and software engineering traceability

software such as Telelogic Doors or IBM Rational Requisite Pro products, or a specialised database can be produced. This would assist in modifying policies as business rules change, and subsequently identify firewall rules that need to be modified and tests that need to be changed. Firewall rule datasets must be kept under version control and configuration control.

**Test Plan Development**

A Test and Evaluation Master Plan (TEMP) functions as a blue print for the test activities, and is comprised of the following activities discussed by Cole, Krutz and Conley (2005, p.55).

- A detailed test plan for complete test coverage of the system under test.
- Communicates the extent and nature of the tests.
- Schedule of events.
- Specification of equipment and organizational requirements.
- Definition of the test methodology.
- Construction of a deliverables list.
- Determination of the expected outputs.
- Instructions on how to carry out the tests.
- Record of the test inputs and results.

Development of the TEMP is instrumental to the effective and comprehensive testing of the firewall, and should be initiated during the initial design phase of the firewall. It can be updated as time evolves and should also form the foundation of the artefacts of the test process. It also indicates a maturity in security systems engineering of the organization.

**Methodology**

The Security Systems Engineering Capability Maturity Model (SSE-CMM) identifies a framework to measure and improve the performance of security systems engineering practices (SSE-CMM Project, 2003). Its scope covers the life cycle of security systems. The phases of the life cycle include concept definition, requirements analysis, design, development, integration, installation, operations, maintenance and decommissioning. Each of these phases have associated mile stones, and are usually marked by the end of test activities and/or reviews. By following processes and practices in a formal fashion, confidence in repeatable results should be attained. The SSE-CMM can be used to rate the maturity of an organization's security engineering practices.

Developing defined processes provides many benefits. It allows knowledge gained in previous efforts to be used in the future, resulting in the ability to accurately predict how much effort in time and manpower is required to perform similar functions. It ensures that results are repeatable and measurable. It enhances efficiencies, and provides confidence that security needs are being met.

Business requirements, legislation, business partnerships and business rules are the predominant drivers for the development of firewall policy. They are the constraints that limit what is passed in and out of the organisation through the internet. This in turn provides impetus for design changes and subsequent test activities of the firewall including ingress and egress testing. As the firewall enters service, it must be monitored and audited. Changes in business rules, threats and the development of new technologies will most likely impact policy, firewall rules, test procedures, monitoring and auditing. These changes may cascade through of their own volition, but assurance is more likely if a comprehensive test plan is followed that checks the status of each component.

Figure 1 below shows the cyclical flow diagram of the six phases of the proposed firewall testing methodology. Each phase has associated activities of risk assessment, test and evaluation, review, reporting and version control. It is very similar to the life cycle phases, but the idea is to revisit these phases to ensure that the design is complying with legislation, business requirements, performance and security requirements. The fundamental idea is to determine if the firewall is still fit for purpose. It starts with checking business requirements to which everything should be traceable to. This would suit a desktop review, or a proactive meeting to determine short term and long term requirements. This is followed by testing firewall policy to ensure that it reflects requirements and is testable and enforceable. The rule set is then developed or re-examined. The rule set is then tested to ensure that it is traceable to policy and that anomalies in the rule set have not crept in. The rule set should be maintained under configuration control. The next step is to ensure that the design is fit for purpose as well. This checks that the design is appropriate, network diagrams are audited to ensure that they are technically sound and that the network architecture is properly reflected in the diagrams which should be under version

control. Implementation tests may be a monthly test to check a percentage of the physical network against the diagrams. This helps to ensure that the firewall is not being bypassed by rogue access points. Implementation also checks other physical security issues such as the location of the firewall and to confirm it is in a secure location, and the alarms and locks work. This is followed by ingress and egress testing to ensure that the firewall is functioning correctly. This is followed by ensuring that monitoring and auditing activities, which should be defined in policy, are being performed correctly.

Although the TEMP provides the guidelines, schedules, procedures of the tests, the test results, reports, network diagrams, rule sets and authorizations should all be under version control or configuration control. They must be easily accessible by security staff for the purposes of audit, or in the event of a security breach where they will come under scrutiny. It also shows a maturity in process, and time required for test activities can be accurately predicted and budgeted for.

The following sections discuss the testing of each component of the methodology in more detail. It should be clear that the cycle is never ending. Each section provides justifications for testing of each phase together with a sample checklist. It should be clear that the tests listed are suggestions only. It would be up to each organization to tailor their own tests to ensure that the overall design and implementation is fit for purpose as defined in policy.

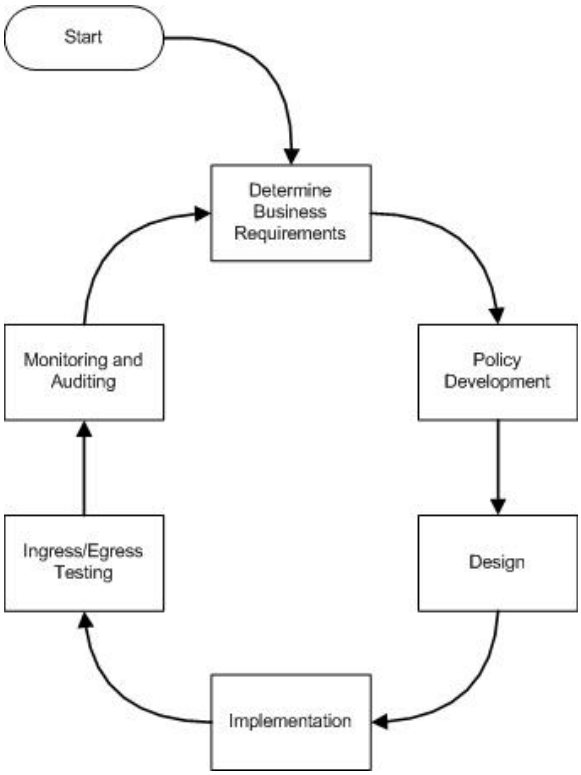


*Figure 1*. Cyclical flow diagram of the test methodology.

**Policy Development and Testing**

Policies must accurately reflect the business needs of the organization. Business rules, partnerships, legislation, risk assessments and technological requirements form the constraints from which policy is developed. Traceability must exist between policy and these requirements to show that policy is protecting business functionality. This also assists in developing assurance. "Assurance is defined as the measure of confidence that the security features and architecture of an information system accurately mediate and enforce an organization's information system security policy" (Cole et.al., 2005, p.591).

Steps recommended by NIST to develop firewall policy (2002, p33) include:

- Identify necessary network applications – These are the applications required to meet business rules and partnerships. It could include having a mail server, web services, and a virtual private network. These are all necessary for the business to operate.
- Identify any vulnerability associated with the applications – The applications could have vulnerabilities and these need to be researched. Threats need to be determined, as well as associated mitigation strategies. The results of these activities must be delivered in the form of a risk analysis document.
- Perform a cost benefit analysis of various methods that can secure the application – This assists in determining if the benefit of having the application outweighs the potential cost. If having the application is going to cost more in terms of employment of security measures than the revenue it will deliver, then it may not be an economically feasible operation.
- Create an application traffic matrix which shows the protection method - The development of a firewall policy can be assisted by the development of an applications traffic matrix. An example of which is shown as table 1 below (NIST, 2002, p.33).
- Create the firewall rule set from the application traffic matrix – A significant measure of traceability should be evident using the above steps in the development of the rule set. This provides assurance and especially that a maturity has been demonstrated by using process.

Table 1

*Sample Application Traffic Matrix*

| TCP/IP Application Service | Location | Internal Host Type | Internal Host Security Policy | Firewall Security Policy (Internal) | Firewall Security Policy (External) |
|---|---|---|---|---|---|
| HTTP | Any | Unix | Proxy | Permit | Permit |
| SMTP/POP | Any | Unix | Anti Spam, Mail Policy | Permit | Permit |
| SSH | Specified Locations | Unix | Remote Access Policy | Permit | Reject All, except by Written Authorization |
| NetBIOS | Any | Windows | Limit Access to Shares | Permit Local Domain Only; Reject Others | Reject |
| NFS | Any | Unix | Limit Exports | Reject | Reject |

Testing and auditing the firewall helps to provide the assurance that is required, but it is essential that policies are written so that they can be implemented and are individually testable. Moyer and Schultz (n.d., p.3) discuss a number of issues that must be answered from the testing process, and are summarized as follows. A firewall must effectively implement policy, and testing to ensure that the firewall implements policy correctly is critical. Policies should be written such that they are testable to ensure compliance. The policy should work hand in glove with the network services that are required. It must be determined if the firewall and other network components provide adequate protection from attacks initiated from external sources. The testing process can help to indicate the ability of the firewall to resist attacks, and help to refine the firewall policy in an iterative and recursive development process. It is equally important to test the firewall from the inside too. Internal weaknesses may exist and these potential leakages must be identified. It is also important to determine how much information about the network is available from the internet. Attacks from the internet can include being able to map the network and determine its configuration.

Policies and rule sets can be peer reviewed on paper, and must have associated test criteria that will have definitive results, with either a pass or fail. All tests, results and reports are artefacts of the test life cycle of the firewall and should be retained. This assists in the iterative test life cycle of the firewall as it evolves and changes as business requirements and technology change. The policies and rule sets should be reviewed regularly to ensure effectiveness. This also assists with traceability and justifications for design and test decisions. Sample policy tests that could be developed further are listed below in table 2.

**Firewall Physical Design**

Firewalls are usually either appliance type devices or software systems that run over an underlying operating system. Whitman and Mattord (2005, p.241) list five major processing categories of firewalls as packet filtering

firewalls, application gateways, circuit gateways, MAC layer firewalls and hybrids. Each different type typically operates at different layers in the Open System Interconnect (OSI) model. Firewalls can also offer additional services such as Network Address Translation (NAT), encryption functionality through a Virtual Private Network (VPN), Dynamic Host Configuration Protocol (DHCP) and application content filtering (NIST, 2002, p.4). Firewalls can be configured in a variety of network connection architectures. These include packet filtering

*Table 2 Policy Tests*

| Test | Description | Date | Checked By | Result |
|---|---|---|---|---|
| Identification of Network Applications | Identification of all network applications | | | |
| Vulnerability Assessment | Perform a vulnerability assessment of network applications | | | |
| Threat Assessment | Perform a threat assessment of network applications | | | |
| Risk Assessment | Perform a risk assessment of network applications | | | |
| Cost Benefit Analysis | Perform a cost benefit analysis of network applications | | | |
| Application Traffic Matrix | Develop an application traffic matrix | | | |
| Policy | Develop firewall policy | | | |
| Peer Review of Policy | Peer review of firewall policy | | | |

routers, screened host firewalls, dual homed host firewalls, screened subnet firewalls (with DMZ) and SOCKS

servers (Whitman et.al., 2005, p256-260). Firewalls can be placed within the organizations intranet to separate LANs, or as a bastion host to the hostile internet.

This shows how much consideration has to go into selecting the right firewall for the right purpose at the right time. There are performance issues to consider, as well as operational and monitoring factors, fitness for purpose, operational costs, vulnerabilities, threats, business rules, and partnerships. Not only do all these factors have to be considered at design time, they also have to be considered during testing during its service life. Artefacts of the original design decisions should have been retained, and the criteria that were used for that phase of the life cycle should be reviewed to ensure that they still meet current and future requirements. Other considerations include performing a physical inspection of the network to ensure that the firewall is not being bypassed, and that network diagrams are up to date. Sample tests are shown in table 3 below.

*Table 3 Firewall Physical Design Tests*

| Test | Description | Date | Checked By | Result |
|---|---|---|---|---|
| Original Design Criteria | Is the original design criteria still valid? Can any new criteria be identified? | | | |
| Network Diagrams | Are the network diagrams up to date? Are the network diagrams technically correct? | | | |
| Physical Inspection | Is the network configured as per the network diagrams? | | | |
| Fitness for Purpose | Is the design still effective? | | | |
| Threat Assessment | List threats to physical design | | | |
| Risk Assessment | Assess the risks for the physical design. | | | |
| Performance Measurement | Is the firewall an excessive bottleneck to performance? | | | |

**Firewall Rule Anomaly Discovery**

"Serious attention has to be given to rule relations and interactions in order to determine the proper rule ordering and guarantee correct security policy semantics" (Al-Shaer, Hamed , 2004, p.1). As the rule set increases, the addition of new rules or modification of existing rules must not conflict with the intent of policy. Anomalies may be introduced if the rule set is not optimised, leading to a less than effective firewall implementation, in terms of both performance and security. In the worst case, it could introduce a vulnerability to the network resulting from a misconfiguration of the firewall rules. The rules are modified as business rules and relationships change, technologies are introduced and removed, administrators come and go, and best practices evolve. Various modelling algorithms have been developed to discover anomalies within the firewall rules. Anomaly classifications listed by Al-Shaer et.al., using the rule set listed in table 4, and the network diagram in figure 2, include the following:

Shadowing Anomaly – A shadowed rule will never be activated because a previous rule matches the same packets. This may result in packets that should be denied to be permitted, or packets that should be permitted, denied. For example, rule 4 is shadowed by rule 3 in table 4.

Correlation Anomaly – Rules can be correlated if one rule matches some packets from a second rule, whilst some packets from the second rule match some packets from the first rule. An example of this is demonstrated by rules one and three in table 4. Rule 1's intention is to deny http traffic from address 140.192.37.20, and rule three accepts all source addresses to destination address 161.120.33.40. If the order is reversed, the traffic denied in rule one will be accepted.

Generalization Anomaly – If one rule matches a preceding rule, have different actions, and if the first rule matches the second rule, the second rule is a generalisation of the first rule. This is demonstrated where rule 1 in table 4 denies http traffic from 140.192.37.20, but rule 2 would accept the traffic from the same address.

Redundancy Anomaly – If policy will not be affected if a rule is removed, the rule is redundant. Rule 7 is redundant because rule 6 matches the same condition, and rule nine is redundant to rule 10. Rules 7 and 9 can then be removed.

Irrelevance Anomaly – If a rule does not match a domain that is catered for by the firewall, then it is an irrelevant rule and can be removed to improve the firewalls performance. Rule 11 is irrelevant because the 140.192.38.* network and the 161.120.35.* networks are not part of the domain on the inside of the firewall.

Table 5 lists possible tests that could be conducted in a desktop review of the rule sets.

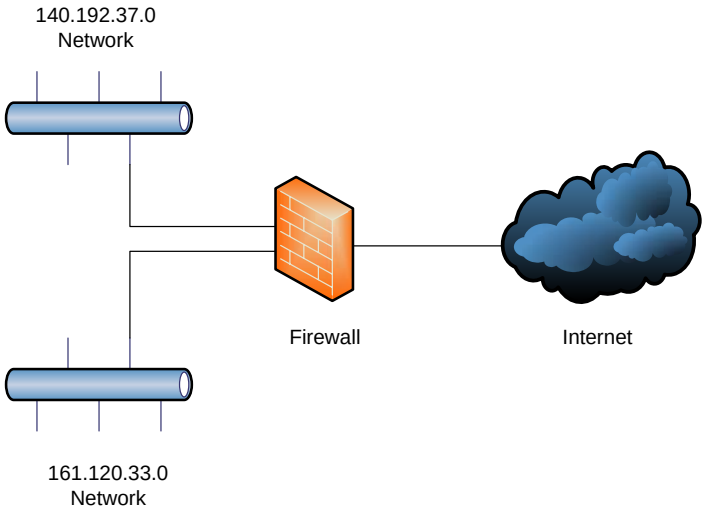


*Figure 2.* Network architecture for the rules in table 4.

Table 4 *Firewall filtering policy (Al-Shaer et.al., 2004, p.2)*

| Rule Number | Protocol | Source Address | Source Port | Destination Address | Destination Port | Action |
|---|---|---|---|---|---|---|
| 1 | tcp | 140.192.37.20 | any | *.*.*.* | 80 | deny |
| 2 | tcp | 140.192.37.* | any | *.*.*.* | 80 | accept |
| 3 | tcp | *.*.*.* | any | 161.120.33.40 | 80 | accept |
| 4 | tcp | 140.192.37.30 | any | 161.120.33.40 | 80 | deny |
| 5 | tcp | 140.192.37.30 | any | *.*.*.* | 21 | deny |
| 6 | tcp | 140.192.37.* | any | *.*.*.* | 21 | accept |
| 7 | tcp | 140.192.37.* | any | 161.120.33.40 | 21 | accept |
| 8 | tcp | *.*.*.* | any | *.*.*.* | any | deny |
| 9 | udp | 140.192.37.* | any | 161.120.33.40 | 53 | accept |
| 10 | udp | *.*.*.* | any | 161.120.33.40 | 53 | accept |
| 11 | udp | 140.192.38.* | any | 161.120.35.* | any | accept |
| 12 | udp | *.*.*.* | any | *.*.*.* | any | deny |

Table 5

*Rule Set Tests*

| Test | Description | Date | Checked By | Result |
|---|---|---|---|---|
| Peer Review | Peer review of rule sets | | | |
| Traceable to Policy | Check that the rules are traceable to policy | | | |
| Anomalies | Check for shadowing anomalies? | | | |
| | Check for correlation anomalies? | | | |
| | Check for generalization anomalies? | | | |
| | Check for redundancy anomalies? | | | |
| | Check for irrelevance anomalies? | | | |
| Implementation | Are the rules implemented as per the rule set properly? | | | |

**Implementation Testing**

Best practices recommend testing the physical security of the environment the firewall is in (NIST, 2002, p.39). Physical access to firewalls must be limited to only those that need access to them. They should be in secure areas that employ monitored alarms, are intruder resistant and protected from disasters such as fire and flood. Electrical supplies should be considered including the supply of an Uninterruptible Power Supply (UPS). Air conditioning and air filtration are environmental controls that may also be required to be tested. Table 6 below contains a checklist of possible inspection tests for physical security of the firewall.

Table 6 *Physical Security Tests*

| Test | Description | Date | Checked By | Result |
|---|---|---|---|---|
| Intruder Resistant | Is the facility resistant to intruders? | | | |
| | Locks on doors that work? | | | |
| | Floor to ceiling walls? | | | |
| Monitored Alarm | Is there a monitored Alarm? | | | |
| | Is there a motion detector? | | | |
| | Is there a fire detector? | | | |
| | Is there a fire control system? | | | |
| UPS | Is there a UPS? | | | |
| | How long can the UPS supply current for? | | | |
| | Is there line conditioning? | | | |
| Air Conditioning | Is the temperature of the facility below 25ºC? | | | |
| Disaster Resistant | Is the facility resistant to disasters? | | | |

**Ingress and Egress Testing**

For the purpose of demonstration, the network topology as shown in figure 3 was configured on VMware using BackTrack 1.0 as the attack platform, Fedora Core 5 configured with iptables v1.3.5 as the firewall, and the three servers in the DMZ simulated by the honeypot software, honeyd 1.5b. The simulated internet is on a separate virtual LAN to the servers in the DMZ.
The firewall is fairly simplistic for demonstration purposes and focuses on servers in the DMZ, as well as the internet facing firewall. The firewall was configured to masquerade the servers in the DMZ using the following commands:

```
# Set Policy, drop all packets on all chains
iptables –P INPUT DROP
iptables –P OUTPUT DROP
iptables –P FORWARD DROP

# Drop all packets with unknown connection states, and that do not have a current connection
iptables –A FORWARD –m state –state INVALID –j DROP

# Allow packets which are from an existing connection
iptables –A FORWARD –m state –state RELATED,ESTABLISHED –j ACCEPT

# eth0 is facing the DMZ, eth1 is facing the internet
iptables –A FORWARD –i eth0 –o eth1 –j ACCEPT
iptables –A FORWARD –i eth0 –o eth0 –j ACCEPT


# Setup network address translation so that the 10.0.0.0 network is hidden from the internet, and the interface
# that is connected to the internet is eth1
iptables –t nat –A POSTROUTING  –s 10.0.0.0/24 –o eth1 –j MASQUERADE

# Use the masquerading to send the right protocol to the right address in the DMZ
iptables –t nat –A PREROUTING –p tcp –dport 53 –j DNAT –to-destination 10.0.0.2
iptables –t nat –A PREROUTING –p tcp –dport 80 –j DNAT –to-destination 10.0.0.3
iptables –t nat –A PREROUTING –p tcp –dport 25 –j DNAT –to-destination 10.0.0.4
```

# Drop anything that is trying to use the private IP address

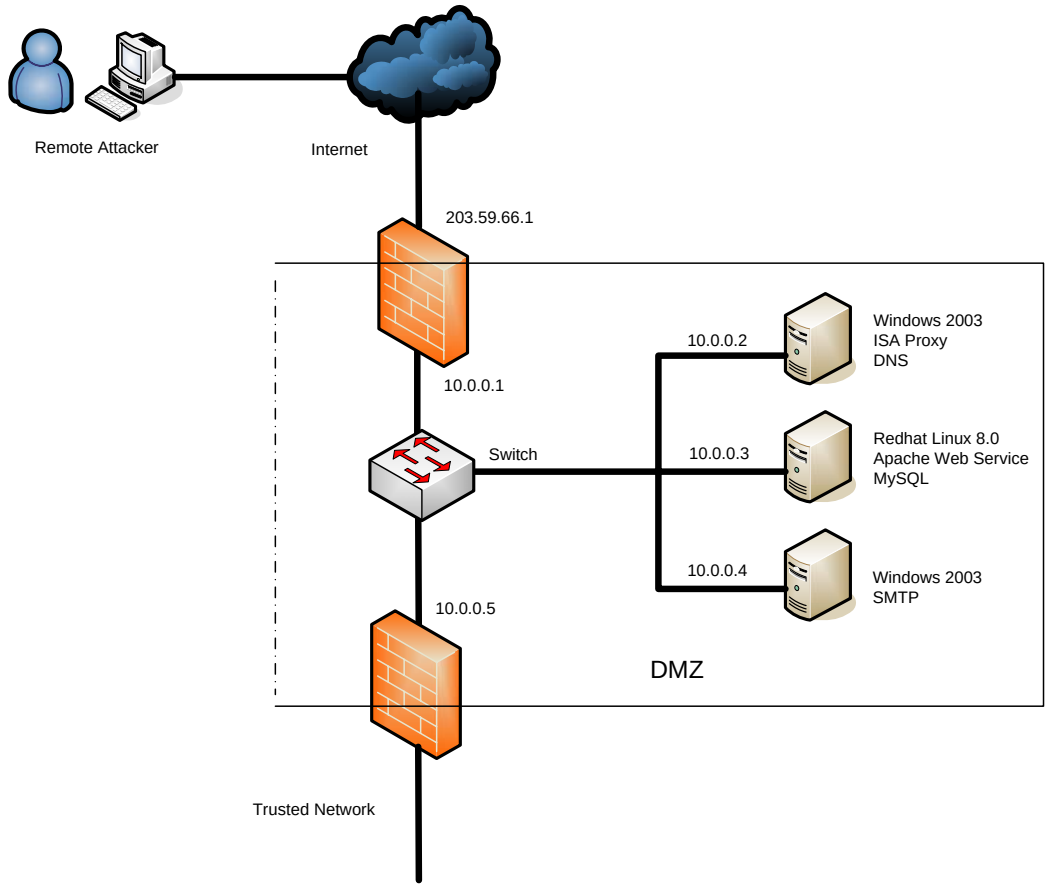


*Figure 3.* DMZ Topology

iptables -t nat -A PREROUTING -i eth1 -s 10.0.0.0/8 -j DROP

The Firewall Analysis Template from the OSSTM (Herzog, 2003, p.86) is an example of access control testing. The following lists the tests suggested by Herzog, together with commands using common tools that will realize his recommendations using the topology given above. An attacker could quickly locate the pseudo IP address of the web server as 203.59.66.1, and no doubt would consider this as a first point of attack.

Fingerprinting - uses packet response to fingerprint the firewall. nmap can be used to try to determine the make of the firewall, as well as any services it may be running.

nmap –sS –O –PI –PT 203.59.66.1

The result of the scan shows that only the SMTP, DNS and HTTP only services were detected, which is a good result.

Stealth – a SYN stealth scan through the firewall in an attempt at enumeration.

nmap –sS –PI –PT 203.59.66.1 –p 80

In this case, it didn't add any new information.

Source port control – scan of specific common, source ports for enumeration. If the firewall allows DNS transfers (port 53), or FTP data (port 20) nmap's –g option allows you to spoof the value of the source port.

nmap –sF –g 53 203.59.66.1

Again nothing new added.

Overlap – uses overlapped fragments such as a teardrop DoS attack.

This would be more effective as a DDoS attack using a tool such as Shaft or Tribal Flood Network.

Fragments – determines if the firewall can handle fragmented packets. This uses nmap's –f switch to send fragmented packets. Splitting up the TCP header over several packets makes it much harder for packet filters to detect an attack.

nmap –sS –f 203.59.66.1

Again, this offered no new information.

SYN flood – can the firewall cope with a series of SYN packets?

Require a SYN flood generator

RST flag – how does the firewall respond to packets with the RST flag set?

UDP – how does the firewall manage standard UDP packets?

ACK – uses ACK packets for enumeration purposes.

nmap –sA –O –PI –PT 203.59.66.1

This determined nothing new.

FIN – uses FIN packets for enumeration purposes.

nmap –sF –O –PI –PT 203.59.66.1

This revealed nothing new.

NULL – uses null packets for enumeration purposes.

nmap –sN –O –PI –PT 203.59.66.1

WIN – uses win packets for enumeration purposes.

nmap –sW –O –PI –PT 203.59.66.1

XMAS – uses packets with all flags set for enumeration purposes.

nmap –sX –O –PI –PT 203.59.66.1

Sustained TCP connections – is the firewall susceptible to a denial of service attack?
This would be more effective as a DDoS attack using a tool such as Shaft or Tribal Flood Network.

Fleeting TCP connections – is the firewall susceptible to a denial of service attack?
This would be more effective as a DDoS attack using a tool such as Shaft or Tribal Flood Network.

Streaming UDP throughput - the firewall susceptible to a denial of service attack?
This would be more effective as a DDoS attack using a tool such as Shaft or Tribal Flood Network.

ICMP responses – how does the firewall respond to different types of ICMP packets?

Spoof responses – can IP addresses be used to determine the access control list?

nmap –S 10.0.0.1 –e eth0 203.59.66.1

Firewall blocked ok.

Protocol – can the firewall stop packets using various protocols?

The results show that the firewall was effective at masquerading the true IP addresses of the servers in the DMZ. The list suggested by Herzog has been composed into a test template below as table 7. An additional test was added that was not explicitly in Herzog's list. That is, is it possible to determine the firewall rule set? This could possibly be done using a tool such as firewalk.

*Table 7 OSSTM Access Control Test List*

| Test | Description | Date | Checked By | Result |
|---|---|---|---|---|
| Fingerprinting | Determine the make of the firewall? | | | |
| Stealth | Scan through the firewall? | | | |
| Source Port Control | Ports open? | | | |
| Overlap | Can overlap fragments pass? | | | |
| Fragments | Does the firewall handle fragmented packets? | | | |
| SYN Flood | Does the firewall handle a SYN flood? | | | |
| RST Flag | What happens when the RST flag is set? | | | |
| UDP | Are standard UDP packets handled? | | | |
| ACK | Use ACK for enumeration | | | |
| FIN | Use FIN for enumeration | | | |
| NULL | Use NULL packets for enumeration | | | |
| WIN | Use WIN packets for enumeration | | | |
| XMAS | Set all flags in packet for enumeration | | | |
| Sustained TCP Connections | Susceptible to a DOS via sustained TCP connections? | | | |
| Fleeting TCP Connections | Susceptible to a DOS via fleeting TCP connections? | | | |
| Streaming UDP Throughput | Susceptible to a DOS attack? | | | |
| ICMP Responses | Try various ICMP packets | | | |
| Spoof Responses | Can IP addresses be spoofed to determine the ACL? | | | |
| Protocol | What happens when various protocols are tried? | | | |
| Determination of Rule Sets | Can the rule set be determined by using a tool such as firewalk? | | | |

**Monitoring and Auditing**

Monitoring and auditing activities should be reflected in policy. Monitoring may be in conjunction with other security systems such as an Intrusion Detection System. Table 8 below shows a sample checklist for testing monitoring and auditing activities.

*Table 8 Monitoring and Auditing Tests*

| Test | Description | Date | Checked By | Result |
|---|---|---|---|---|
| Monitoring | Logging occurring correctly?<br>Providing inputs to IDS? | | | |
| Auditing | Complying with auditing policy? | | | |

**Documentation**

The results of the tests should be reported, reviewed and archived. They should be available for consultation prior to each test to compare previous results. The rule sets should be under configuration control, with authorisation required to make changes. Documents under version control should include the TEMP, policy and network diagrams. All documentation should be available if required by authorised personnel, especially for times of audit, or a security breach investigation.

The complexity of the traceability can be managed in a requirements management tool such as Telelogic's Doors, Rational's Requisite Pro, or a customised database with a forms front end. If a business requirement is modified, related policy, rule sets, design, ingress and egress testing, and monitoring relationships can be brought to the attention of the security personnel and modified appropriately. This maintains assurance, improves testing response and facilitates good management.

## CONCLUSION

The objective of this paper has been to show that there is far more to testing a firewall than just performing ingress and egress testing. Business rules, legislation, technology and business partnerships evolve and drive changes that cascade through policy, rule sets, ingress and egress testing, monitoring and auditing. A cyclical methodology was discussed that outlines a plan that checks that everything is traceable back through to business requirements. Policy should be directly traceable to business requirements, and the design of the firewall including rule sets and physical design should be traceable to policy. Ingress and egress tests must be traceable to the design and auditing and monitoring must be reflected in policy. The purpose is to ensure that the design is fit for purpose in a cyclical fashion that never ends. This can help to capture configuration errors, to optimise performance, and to be proactive in meeting business requirements.

## REFERENCES

Al-Shaer, E.,S., Hamed, H.,H., (2004). *Modelling and Management of Firewall Policies*, Retrieved September 05, From www.mnlab.cs.depaul.edu/projects/FPA/files/tnsm04.pdf

Cole, E., Krutz, R., Conley, J.W., (2005). *"Network Security Bible"*, Wiley Publishing, Inc., Indianapolis.

Herzog, P., (August 23 2003). *OSSTMM 2.2.1. Open-Source Security Testing Methodology Manual*, Retrieved October 5, 2006 From http://www.isecom.info/mirror/osstmm.en.2.1.pdf

Moyer, P.R., Schultz, E.E., (n.d.). *A Systematic Methodology for Firewall Penetration Testing*, Retrieved September 15, 2006 From projects.cerias.purdue.edu/firewall/references/fwtest.doc

NIST, (2002), *SP800-41, Guidelines on Firewalls and Firewall Policy*. Retrieved October 5, 2006 From http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf - 2002-01-04

SSE-CMM Project, (June 15 2003), *Systems Security Engineering Capability Maturity Model, SSE-CMM, Model Description Document, Version 3.0*. Retrieved October 6, 2006 From http://www.sse-cmm.org/docs/ssecmmv3final.pdf

Whitman, M., Mattord, H, (2005). *Principles of Information Security*. Thomson. Boston, Massachusetts.

## COPYRIGHT

Murray Brand ©2007. The author/s assign SCISSEC & Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive license to SCISSEC & ECU to publish this document in full in the Conference Proceedings. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors