Global Information Assurance Certification Paper

GIAC Firewall and Perimeter Protection Curriculum

Practical Assignment for SNAP San Jose
Submitted By: Michael Roney

Assignment 1 – Egress Filter

Why Egress Filters?

Egress filtering is the practice of filtering Internet traffic leaving a local network. Most organizations with an Internet connection are primarily worried about the traffic that is entering their network. All organizations should also be worried about the traffic leaving their network. Implementing egress filtering provides the following:

- Prevents spoofing of outbound packets.
- May help in intrusion detection.
- Makes your network less appealing to hackers.
- Makes your network a "Good Internet Citizen".

Many attacks rely upon the attacker having the ability to change the source IP address of packets to make it appear the packet came from a different network. SMURF requires the ability to send packets with a source IP address of the network being attacked. Connection hijacking also relies upon this. To stop these activities from happening, egress filtering is used to verify that only packets with legal addresses owned by the organization are allowed to leave the network for the Internet. With this type of security in place, an attacker has a more difficult time using your network for attacks and hiding their tracks.

Even if the organization feels they have nothing important to protect, the fact that the organization has a permanent Internet connection and is not performing egress filtering makes the site attractive as a launching point for hackers to use. By using the organization's network for launching attacks, the hacker's real network identity is masked. Spoofed packets are difficult to trace. Spoofed packets arriving from the Internet take the cooperation of several ISPs in order to track down the source. Without implementing egress filtering, the company could be placed in the embarrassing position of having their computer resources used by hackers and/or malicious employees to attack other Internet sites. By implementing egress filtering, the organization is being a responsible Internet citizen.

Egress filtering may also help point out any compromised systems in the organization. By implementing egress filters and logging the spoofed packets leaving the network, the organization can determine the cause of the spoofed packets (a compromised system, a malicious employee, etc…) and take appropriate action. Egress filters also keep the organization's private Internet addresses from being transmitted to the Internet.

Applying Egress Filters

Egress filters are generally applied to each border router of an organization. If an organization is assigned the class-C legal address space of 210.1.2.0, the router needs to be configured to only allow packets in this address space to be received by the router interface connected to the internal network.

If the organization were using a Cisco router, they would first create an extended access list that permits the IP addresses 210.1.2.0 – 210.1.2.255 to any destination. The command to create the list is:

```
access-list 110 permit ip 210.1.2.0 0.0.0.255 any
```

Command Key

access-list – command used to create an access list.
110 – number of the access-list (used to reference the access list by other commands).
permit – keyword that says we are creating an access-list to allow something.
ip – type of packet (in this case TCP, UDP and ICMP packets will be checked).
210.1.2.0 – source IP address to check.
0.0.0.255 – 'wild card' - a binary bit field used to mask bits in the IP address from testing (this case we are checking source addresses in the range 210.1.2.0 – 210.1.2.255).

any – keyword used in place of the destination address/wild card 0.0.0.0 255.255.255.255 (basically, any destination IP address).

To apply the filter to the router interface the **ip access-group** command is used. This command is used after the interface to add it to has been selected with the **int** command (i.e. **int eth0** - if eth0 is the router interface facing the internal network). Here are the commands to apply the filter to the interface.

```
int eth0
ip access-group 110 in
```

Command Key
int – command to select the interface.
eth0 – interface to select.
ip access-group – command to apply the access list to the interface.
110 – number of the access list to apply to the interface (from above).
In – the direction the packet is examined (on its way in or on its way out of the interface).

To add logging of any denied packets, the access list can be modified by adding the following line after the permit entry:

```
access-list 110 deny ip any any log
```

The log switch on the end of this filter tells the Cisco router to record any packets that match the filter's conditions. All traffic with a source IP address of 210.1.2.0 would match the first rule and be allowed through without being logged. All traffic that is not using this address space would be dropped by the router and promptly logged due to the second rule.

It is also possible to send the logged entries to a central logging system with the global configuration command:

```
logging 210.1.2.10
```

Command Key
Logging – command to direct the log messages to an appropriate location.
210.1.2.10 – IP address of the server to capture the logged messages.

When complete, this filter says '*If the source IP is in the range 210.1.2.0 – 210.1.2.255, let the traffic through. Drop all packets using any other address as a source IP address, create a log entry of the denied packet and send the log entry to 210.1.2.10*.' Once this filter has been applied, internal systems will no longer be able to generate packets that will appear to have originated from another network.

Testing the Filter
To test the filter once it has been applied, a PC could be configured to use an illegal or unsupported IP address. Once configured, it can be used in an attempt to access a system on the Internet with FTP, Telnet, a web browser, etc… Because the filter applied to the router is logging, the log can be checked to see the packet has been blocked. Another indicator is that the PC never successfully makes a connection using any application.

Assignment 2 – Firewall Policy Violations

Network Environment

These log entries come from a Gauntlet 5.5 firewall running on HP-UX 10.20. We have two Gauntlet firewalls that protect a full class-B network (except for one class-C subnet that is used on the "dirty" side of the firewalls – between the firewalls and the Internet router).

Firewall Rules

The Gauntlet firewall is a proxy firewall. As such, the "rules" define what source IP address is permitted to use a "service" (i.e. proxy) directed toward a destination IP address. The rule set currently utilized allows for outbound FTP, Telnet, SMTP, HTTP and SSL. Inbound FTP, Telnet, HTTP and SSL is not allowed, and inbound SMTP is only allowed to the firewall(s) which then pass the received email to a Lotus Notes SMTP gateway server. All traffic passing through the Gauntlet to the Internet appears to be originating from the external interface of the Gauntlet firewall.

Network Groups – Gauntlet defines "network groups" that are used in rules as the source and/or destination IP address for the rule. Our configuration includes the following "network groups":

Group Name	Included Addresses
Trusted	xxx.xx.*.*, 127.0.0.1
Untrusted	xxx.xx.2.*, *
Mail	*

Service Groups – Gauntlet defines "service groups" that are used in rules to define what services are being allowed or denied by the rule. Our configuration includes the following "service groups":

Group Name	Services
Trusted	ftp-gw, http-gw, ssl-gw, tn-gw (i.e. FTP, HTTP, SSL, and Telnet)
Untrusted	(everything) ftp-gw, http-gw, ssl-gw, tn-gw…
Mail	smap, smapd (smap and smapd are used to receive SMTP mail on TCP port 25)

Source Rules – The following source rules have been configured on the firewall:

Rule	Network Source	Access	Services
1	Trusted	Permit	Trusted
2	Untrusted	Deny	Untrusted
3	Mail	Permit	Mail

Destination Rules – The following destination rules have been configured on the firewall:

Rule	Services	Access	Network Source
1	Trusted	Permit	*
2	Untrusted	Deny	xxx.xx.*.*

Besides these source and destination rules, there are implicit rules in Gauntlet that state "Whatever is not specifically allowed is denied."

As an added security feature, the various Gauntlet proxies can be enabled and disabled. The firewall currently only has the ftp-gw, http-gw, tn-gw, ssl-gw, smap and smapd proxies running.

Security Alert Logging

Gauntlet security alerts appear in the log file /var/log/messages. The Gauntlet logs not only security alerts, but also all messages to this file. To display just the security alerts, a script is run to extract the "securityalert" messages. All security alert messages, except **unserved port** alerts, have the following format:

- The date and time of the security alert.
- **styx** is the name given to this Gauntlet firewall.
- **vmunix:** is the name of the kernel (HP-UX 10.20) the firewall is running.
- **securityalert**: is a keyword to notify the reporting routines that this log entry is security alert.
- the type of security alert is displayed after the **securityalert:** keyword.
- The type of packet **UDP**, **TCP** or **ICMP** is then displayed.
- **if=** specifies the network interface that observed the security alert. Lan0 is the internal interface (private) and lan1 is the external interface (internet).
- **srcaddr=** IP address the packet originated from.
- **srcport=** Port used by the source IP address (not applicable/specified for ICMP packets).
- **dstaddr=** IP address the packet is/was destined for. (The packet did not make it to its destination if it is logged as a security alert).
- **dstport=** Port to send packet to on destination system.

Below are examples of security alerts log entries:

```
Jun  7 19:44:23 styx vmunix: securityalert: source not allowed on interface: UDP
if=lan1 srcaddr=xxx.xx.10.140 srcport=137 dstaddr=xxx.xx.27.155 dstport=137

Jun 13 04:12:49 styx vmunix: securityalert: packet denied by forward screen: ICMP
if=lan1 srcaddr=213.6.190.114 dstaddr=xxx.xx.250.2

Jun 13 04:54:27 styx vmunix: securityalert: no match found in local screen: UDP
if=lan0 srcaddr=32.82.143.10 srcport=53 dstaddr=xxx.xx.2.10 dstport=53
```

Unserved port alerts have the following format:

- The date and time of the security alert.
- **styx** is the name given to this Gauntlet firewall.
- **vmunix:** is the name of the kernel (HP-UX 10.20) the firewall is running.
- **securityalert**: is a keyword to notify the reporting routines that this log entry is security alert.
- The type of packet is displayed after the **securityalert:** keyword – either **upd** or **tcp**.
- **if=** specifies the network interface that observed the security alert. **Lan0** is the internal interface (private) and **lan1** is the external interface (internet).
- The source IP address and port is then specified after the word **from** in the form {source IP address}:{port number} (ex. 63.248.156.203:1128 specifies IP address 63.248.156.203 and port 1128).
- The destination IP address is specified after the word **to**.
- The port to be used on the destination IP address is specified after the words **on unserved port**. I believe the intention here is to clearly specify what port on the firewall is not open for service.

Below is an example of an **unserved port** security alert:

```
Jun 14 02:24:24 styx vmunix: securityalert: udp if=lan1 from 63.248.156.203:1128 to
xxx.xx.76.15 on unserved port 135
```

For ease of readability and monitoring of long-term trends, I reformat the security alerts so they can be imported into MS Access. I present the database entries for each of the firewall security violations listed. Also, the class-B address of the company I work for has been replaced with x's in all log entries. No attempt has been made to hide/mask the IP address of security alert sources.

Violation #1 – Unknown Access Attempt on UDP port 41530

Date-Time	System	Alert-Type	Interface	Src-IP	Src-Port	Dst-IP	Dst-Port	Packet-Type
05/20 9:13:34	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.161.255	41530	UDP
05/20 9:13:34	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.64.63	41530	UDP

05/20 9:13:34	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.51.63	41530	UDP
05/20 9:13:34	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.20.63	41530	UDP
05/20 9:13:34	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.66.63	41530	UDP
05/20 15:14:14	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.64.127	41530	UDP
05/20 15:14:14	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.66.63	41530	UDP
05/20 15:14:14	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.66.255	41530	UDP
05/20 15:14:14	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.53.63	41530	UDP
05/20 15:14:14	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.20.63	41530	UDP
05/20 15:14:14	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.54.63	41530	UDP
…								
05/22 3:13:39	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.51.255	41530	UDP
05/22 3:13:39	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.53.255	41530	UDP
05/22 6:13:39	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.64.63	41530	UDP
05/22 6:13:39	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.64.127	41530	UDP
05/22 9:13:39	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.64.63	41530	UDP
05/22 9:13:39	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.53.63	41530	UDP
05/22 9:13:39	Styx	unserved port	Lan1	216.101.103.6	1142	xxx.xx.64.255	41530	UDP

Description: Some system claiming to be 216.101.103.6 (this address is registered to the City of Vacaville California) is searching for systems responding on UDP port 41530. The system searching is always using UDP port 1142. It's interesting to note the searching system is using what would be considered broadcast packets depending on the mask being used by the network (i.e. x.x.x.255, x.x.x.63).

Blocking Rule: The implicit deny rule on the Gauntlet firewall has blocked these packets. Because there is no proxy listening on UDP port 41530 and no forward filter rules are defined on the firewall, the packet is dropped and blocked. Gauntlet does not, by default, provide a proxy for UDP port 41530.

Potential Damage: This system appears to be seeking out systems that respond on UDP port 41530. Because the packets are of a broadcast nature, I assume there may be some weakness in an application or system using UDP port 41530 that the hacker is attempting to exploit. Without the firewall blocking this, it is possible the hacker could have found an exploit and compromised one or more of our systems.

Violation #2 – POP2 Access Attempts

Date-Time	System	Alert-Type	Interface	Src-IP	Src-Port	Dst-IP	Dst-Port	Packet-Type
05/21 3:17:23	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.4.1	109	TCP
05/21 3:17:23	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.3.1	109	TCP
05/21 3:17:34	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.5.1	109	TCP
05/21 3:17:34	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.6.1	109	TCP
05/21 3:17:39	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.7.1	109	TCP
05/21 3:17:57	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.10.1	109	TCP
05/21 3:18:18	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.15.1	109	TCP
05/21 3:18:36	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.16.1	109	TCP
05/21 3:18:36	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.17.1	109	TCP
05/21 3:19:45	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.29.1	109	TCP
…								
05/22 6:12:17	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.200.70	109	TCP
05/22 6:12:57	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.208.70	109	TCP
05/22 6:13:01	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.139.75	109	TCP
05/22 6:13:02	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.209.70	109	TCP

05/22 6:13:43	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.217.70	109	TCP
05/22 6:13:47	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.148.75	109	TCP
05/22 6:13:48	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.218.70	109	TCP
05/22 6:15:14	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.165.75	109	TCP
05/22 6:15:15	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.235.70	109	TCP
05/22 6:15:19	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.166.75	109	TCP
05/22 6:15:20	styx	unserved port	Lan1	212.244.133.70	0	xxx.xx.236.70	109	TCP

Description: Some system claiming to be 212.244.133.70 (addresses registered to Polskie Radio in Poland) is attempting to locate a system responding to POP2 packets (TCP port 109). What's very suspicious is the source port value being 0.

Blocking Rule: The implicit deny rule on the Gauntlet firewall has blocked these packets. Because Gauntlet does not provide a proxy for POP2, access to TCP port 109 is not specifically dropped by a deny rule. There are no forward filter rules are defined on the firewall for TCP port 109 packets either, so the packet is dropped and blocked.

Potential Damage: A buffer overflow vulnerability in pop2d version 4.4 or earlier allow malicious remote users to obtain access to the "nobody" user account. The pop2 and pop3 servers support the concept of an "anonymous proxy", whereby a remote user connecting to the server can instruct it to open an IMAP mailbox on some other saver they have a valid account on. In this state the pop2 server runs under the "nobody" user id. Once logged on, issuing a FOLD command with an argument of about 1000 bytes will cause a stack based buffer overflow. The vulnerability exists in the following systems: Debian Linux 2.1, RedHat Linux 5.2 i386, RedHat Linux 5.1, RedHat Linux 5.0, RedHat Linux 4.2, RedHat Linux 4.1, RedHat Linux 4.0, University of Washington imap 4.4, University of Washington pop2d 4.4.
We do not have any of these operating systems or applications currently, but if we did and the firewall was not there to block the access attempt, the hacker could have obtained access to the system.

Violation #3 – Internal Attempt to Access POP3 Server

Date-Time	System	Alert-Type	Interface	Src-IP	Src-Port	Dst-IP	Dst-Port	Packet-Type
05/17 13:31:17	styx	unserved port	Lan0	xxx.xx.129.164	1033	xxx.xx.10.45	110	TCP
05/17 13:31:18	styx	unserved port	Lan0	xxx.xx.129.164	1033	xxx.xx.10.45	110	TCP
05/17 13:31:18	styx	unserved port	Lan0	xxx.xx.129.164	1033	xxx.xx.10.45	110	TCP
05/17 13:31:19	styx	unserved port	Lan0	xxx.xx.129.164	1033	xxx.xx.10.45	110	TCP
05/17 15:09:22	styx	unserved port	Lan0	xxx.xx.129.164	1114	xxx.xx.10.45	110	TCP
05/17 15:09:24	Styx	unserved port	Lan0	xxx.xx.129.164	1114	xxx.xx.10.45	110	TCP

Description: The internal system xxx.xx.129.164 is attempting to access a POP3 server via a firewall proxy. The lan0 entry shows that this entry was seen on the internal interface of the firewall. Xxx.xx.10.45 is the internal IP address of the firewall. TCP port 110 is used by POP3.

Blocking Rule: The implicit deny rule on the Gauntlet firewall has blocked these packets. TCP port 110 is used for POP3, but this proxy is not included in the list of "Trusted" services. Because there are no forward filter rules defined on the firewall for TCP port 110, the packet is dropped and blocked.

Potential Damage: A buffer overflow vulnerability in pop2d version 4.4 or earlier allow malicious remote users to obtain Attempt by an internal system to access a POP3 server probably to retrieve email. Since they are attempting to access the firewall the application or end-user is probably aware of the firewall and is attempting to have the firewall retrieve the email via a proxy. This is a security violation. We do not allow internal systems to access external POP3 servers. The reason this type of activity is not allowed is that it could be used to circumvent the virus scanning software we have in place to scan all incoming email.

Violation #4 – SNMP Community String?

Date-Time	System	Alert-Type	Interface	Src-IP	Src-Port	Dst-IP	Dst-Port	Packet-Type
05/27 3:42:51	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.126.2	161	UDP
05/27 3:42:51	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.127.2	161	UDP
05/27 3:42:51	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.124.2	161	UDP
05/27 3:42:51	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.125.2	161	UDP
05/27 3:42:51	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.122.2	161	UDP
…								
05/28 4:33:50	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.249.132	161	UDP
05/28 4:33:50	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.245.132	161	UDP
05/28 5:31:16	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.122.137	161	UDP
05/28 5:31:16	Styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.127.137	161	UDP
05/28 5:31:16	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.126.137	161	UDP
05/28 5:31:16	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.123.137	161	UDP
05/28 5:31:16	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.121.137	161	UDP
05/28 5:31:16	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.120.137	161	UDP
05/28 5:31:16	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.119.137	161	UDP
05/28 5:31:16	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.118.137	161	UDP
05/28 5:31:16	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.117.137	161	UDP
05/28 5:31:16	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.116.137	161	UDP
05/28 5:31:16	styx	unserved port	Lan1	207.96.37.201	1024	xxx.xx.125.137	161	UDP

Description: A system claiming to be 207.96.36.201 is attempting to locate systems with an open SNMP port (UDP 161). Over a period of several hours, the attacker attempts to access systems in no particular order.

Blocking Rule: These packets are blocked by the second source rule. The Gauntlet firewall has a proxy for SNMP (which is not turned on) and the proxy is listed in the "untrusted" service group. Since the packets are coming from an "untrusted" network source, the packets are blocked and dropped.

Potential Damage: If these packets were allowed through, there is a possibility the attacker would be able to communicate with an SNMP server set to use the generic PUBLIC and/or PRIVATE community strings. They may also be able to locate and exploit other vulnerabilities of servers using SNMP or UDP port 161.

Violation #5 – ICMP Packet Block

Date-Time	System	Alert-Type	Interface	Src-IP	Src-Port	Dst-IP	Dst-Port	Packet-Type
05/18 23:24:59	styx	packet denied by forward screen	lan1	64.41.164.54		xxx.xx.1.98		ICMP
05/18 23:25:08	styx	packet denied by forward screen	lan1	64.41.164.54		xxx.xx.3.98		ICMP
05/18 23:25:11	styx	packet denied by forward screen	lan1	64.41.164.54		xxx.xx.4.98		ICMP
05/18 23:25:16	styx	packet denied by forward screen	lan1	64.41.164.54		xxx.xx.5.98		ICMP
05/18 23:25:20	styx	packet denied by forward screen	lan1	64.41.164.54		xxx.xx.6.98		ICMP
05/18 23:25:32	styx	packet denied by forward screen	lan1	64.41.164.54		xxx.xx.8.98		ICMP
…								
05/18 23:42:07	styx	packet denied by forward screen	lan1	64.41.164.54		xxx.xx.247.98		ICMP
05/18 23:42:11	Styx	packet denied by forward screen	lan1	64.41.164.54		xxx.xx.248.98		ICMP
05/18 23:42:15	Styx	packet denied by forward screen	lan1	64.41.164.54		xxx.xx.249.98		ICMP
05/18 23:42:20	Styx	packet denied by forward screen	lan1	64.41.164.54		xxx.xx.250.98		ICMP
05/18 23:42:24	Styx	packet denied by forward screen	lan1	64.41.164.54		xxx.xx.251.98		ICMP
05/18 23:42:30	Styx	packet denied by forward screen	lan1	64.41.164.54		xxx.xx.252.98		ICMP

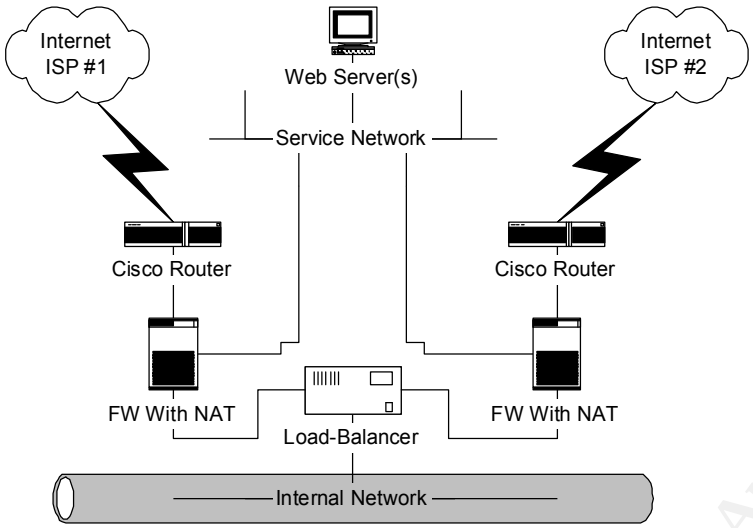
Description: A system claiming to be 64.41.164.54 is attempting send ICMP packets to various systems within our network. These packets are observed on the firewalls external interface. The source and destination port value are blank because ICMP packets do not have/use ports.

Blocking Rule: The implicit deny rule on the Gauntlet firewall has blocked these packets. The only way the Gauntlet firewall would allow an ICMP packet through is by using forward filter rules. Because there are no forward filter rules defined, the packets are blocked and dropped.

Potential Damage: The hacker may be attempting to locate systems that have been previously compromised in an effort to launch a DDOS attack like SMURF. The attacker may also be attempting to map out our internal network with ICMP echo reply packets. In either and all cases, these packets should be blocked. As part of being a good Internet citizen, our network should not allow hackers to utilize its resources in attacking sites.

Assignment 3 – Defense in depth architecture

Part 1 – Design a Network with Two ISP Connections that is Optimized to be Resistant to DDOS Attacks.



With the luxury of having two ISP connections it is possible to build in extra features in an attempt to maintain an Internet presence during a DDOS attack. The diagram above details a configuration with connections to two ISPs.

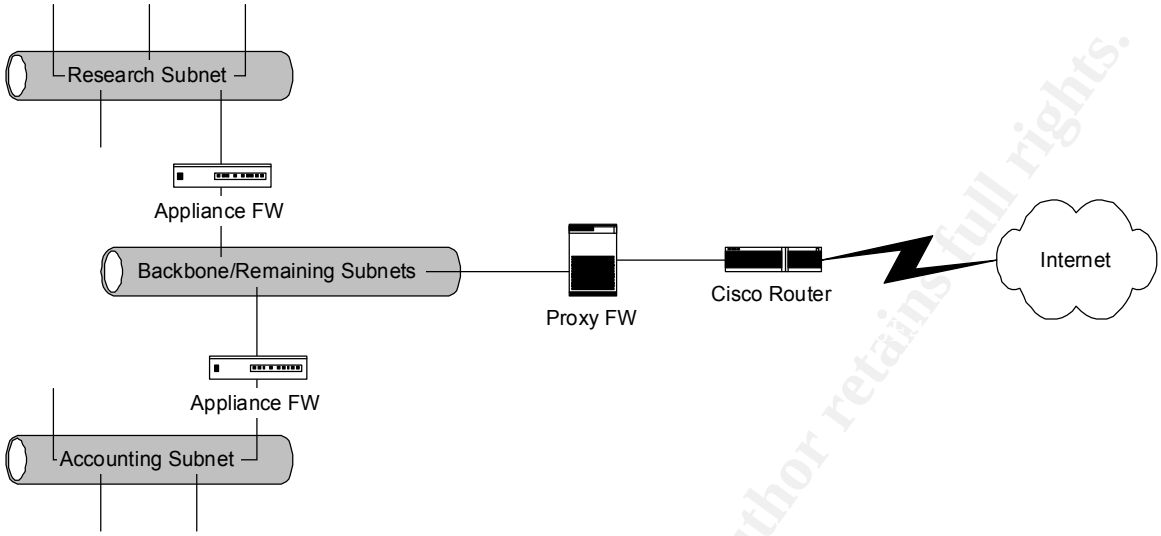
- Each ISP connection comes in to the network via a Cisco router. By using two routers, a DDOS flood attack against one router allows the other router to be unaffected.
- The Cisco routers are then connected to a firewall that is capable of NAT. The firewalls then hide the real addresses of any service network systems that Internet users are allowed to access. The NAT addresses used by each firewall are unique. If an attack is made against one of the NAT'ed addresses, only one firewall is flooded with packets and not the other.
- A load-balancing device can then be used between the internal network and the two firewall systems. This will allow the network to maintain its connection to the Internet (fault-tolerance) if one of the firewalls is flooded by DDOS attack.
- An intrusion detection device could possibly be used to monitor activity of the firewalls and modify the boarder router rules and/or firewall rules in an attempt to route valid traffic to the services they are attempting to reach and block/drop any unwanted traffic.

If only one ISP is used for all inbound traffic and a DDOS is launched against the service network, the internal network will be able to maintain its Internet connection through the other ISP.

I'm not sure if there is a "perfect" answer to creating a network that is optimized to be resistant to DDOS attacks, but I'm much effort is being directed at finding one. I'm sure that when a new solution is found, the hackers will be working just as hard to find a way around the defenses.

Part 2 – Protecting Two Critically Important Internal Sub-networks with Equipment at Hand
Hardware at Hand:
1 – proxy firewall

1 – Cisco router
2 – appliance firewalls



Protection Scheme

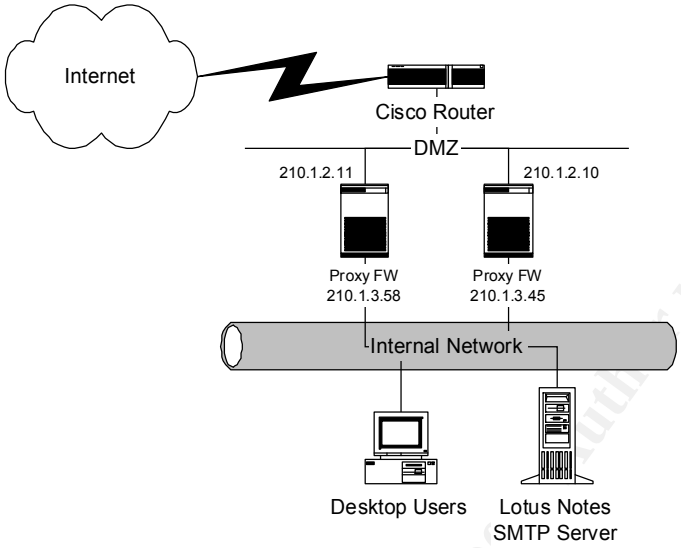
To protect the critically important subnetworks, it is important to have them secured behind layers of defense. Ideally, these defense layers will each enploy a different technology. By using different technologies at each layer of the security, an exploit at one layer cannot be used to gain access through the other layers.

The diagram above shows how the equipment at hand can be employed to provide three layers of security for the two critically important subnetworks.

- The Cisco router is configured as a boarder router with appropriate ingress and egress filters. This is the first layer of defense for the entire network.
- The proxy firewall provides the second layer of security. It is used as the primary security device for all networks.
- To secure the critically important subnetworks further, the one appliance firewall is used on each to isolate and protect the subnetwork. By using an appliance firewall for each subnetwork, it is protected from Internet attacks that have made it through the router and firewall and attacks from employee systems in the less secure subnetworks that have been compromised or are being used by employees with questionable morals.
- With the luxury of one applicance firewall for each critical subnetwork, the rules and access lists used to defend the critical subnetworks are easier to maintain. Without have to worry about rules blocking some accesses for certain subnetworks and allowing access to others there is less of a chance a mistake is made in the configuration of the appliance firewalls.

Assignment 4 – Create a test that demonstrates your knowledge of the subject area

Question/Problem



Current Situation:

- A company has two proxy firewalls each with two 10/100 network cards. The external IP addresses of the firewalls are 210.1.2.10 (resolves to fw1.mycompany.com using an Internet DNS server) and 210.1.2.11 (resolves to fw2.mycompany.com using an Internet DNS server). The internal IP addresses of the firewalls are 210.1.3.45 (resolves to fw1.mycompany.com using the internal DNS) and 210.1.3.58 (resolves to fw2.mycompany.com using the internal DNS).
- All of the browsers have been configured to use one of the firewalls as a proxy server.
- The default route/gateway of last resort (0.0.0.0) on all internal routers has been configured to use the same firewall (210.1.3.45).
- The external DNS MX records for the domains being served by the firewalls are set to:

Mycompany.com	IN	MX	10	fw1.mycompany.com
	IN	MX	100	fw2.mycompany.com

- The firewalls are configured to block all inbound connection attempts except email to the firewalls themselves.
- The firewalls allow access to outbound FTP, HTTP, Telnet, SSL and SMTP (directly to the firewall).

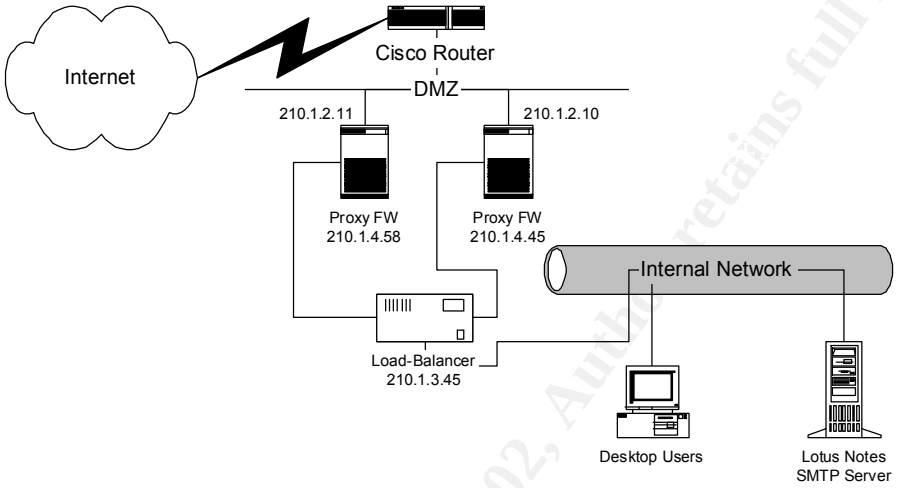
Originally the idle firewall was purchased so if the primary firewall failed the idle firewall could be used for Internet access. Because of this configuration the one proxy firewall (210.1.3.45) is being overloaded with HTTP, FTP, Telnet, SSL and SMTP e-mail processing tasks. Because of the load on the one firewall, emails are not being send in a timely mannor and access to web sites is slow and getting slower as the company grows. Management is asking why there is an idle firewall and is looking for recommendations for utilizing the idle firewall to provide faster Internet access to FTP, HTTP, SSL, Telnet, and SMTP mail. Suggest changes to the environment that will increase firewall throughput and where possible create redundancy and possibly automatic fail-over.

Answers/Resolutions

Recommendations Include:

1. Utilize a DNS round robin to balance the load between the firewalls.
   Configuration:

- To balance the external SMTP based email being sent to the firewall, modify the Internet MX records for the mycompany.com domain. By setting the MX preference values to be identical, inbound SMTP mail would be somewhat balanced between the firewalls.
- Hopefully, the web browsers are used a system name instead of an IP address for the proxy server. If this is the case, the internal DNS can be modified to have two A records for the proxy server name. This way when the browsers looked up the IP address of the proxy server, the IP address passed back from DNS each lookup would bounce back and forth between the two firewall IP addresses.



- If an internal SMTP server is being used for email, it also could be configured to use the DNS proxy server name to send email using both firewalls.

Disadvantages:

- If one firewall fails, every other access for firewall service fails.
- The load on the two firewalls is not optimally balanced – only haphazardly balanced.

Advantages:

- Utilizes both firewalls to some degree.
- No new hardware/software needs to be purchased.

2. Purchase a device to provide load balancing between the firewalls.

Configuration:

- Purchase a load-balancing device (i.e. Big/IP, FireProof, Cisco LocalDirector).
- Place the load-balancing device in front of the firewalls on the internal network.
- Give the load-balancing device the IP address of the overloaded firewall. Since most if not all outbound traffic is configured to use the overloaded firewall, the traffic is now directed to the load-balancing device.
- Configure the load-balancing device to balance the outbound traffic between the two firewalls.
- Since the only inbound traffic is SMTP email, the external DNS MX records can be modified to have the same preference values to effectively balance the inbound SMTP traffic between the two firewalls.

Disadvantages:

- Requires the expenditure of additional funds to secure the load-balancing device.
- If a fault-tolerant load-balancing device is not acquired, it becomes a single point of failure.

Advantages:

- The load between the firewalls is truly balanced.
- If one firewall fails, there is no interruption in Internet service (although it is slower because all Internet traffic is being passed through a single firewall).