Global Information Assurance Certification Paper

Practical Assignment for
GIAC Firewalls, Perimeter Protection, and Virtual Private Networks

Version 1.5b

SANS New Orleans, January 28 – February 2, 2001

Dan L'Heureux

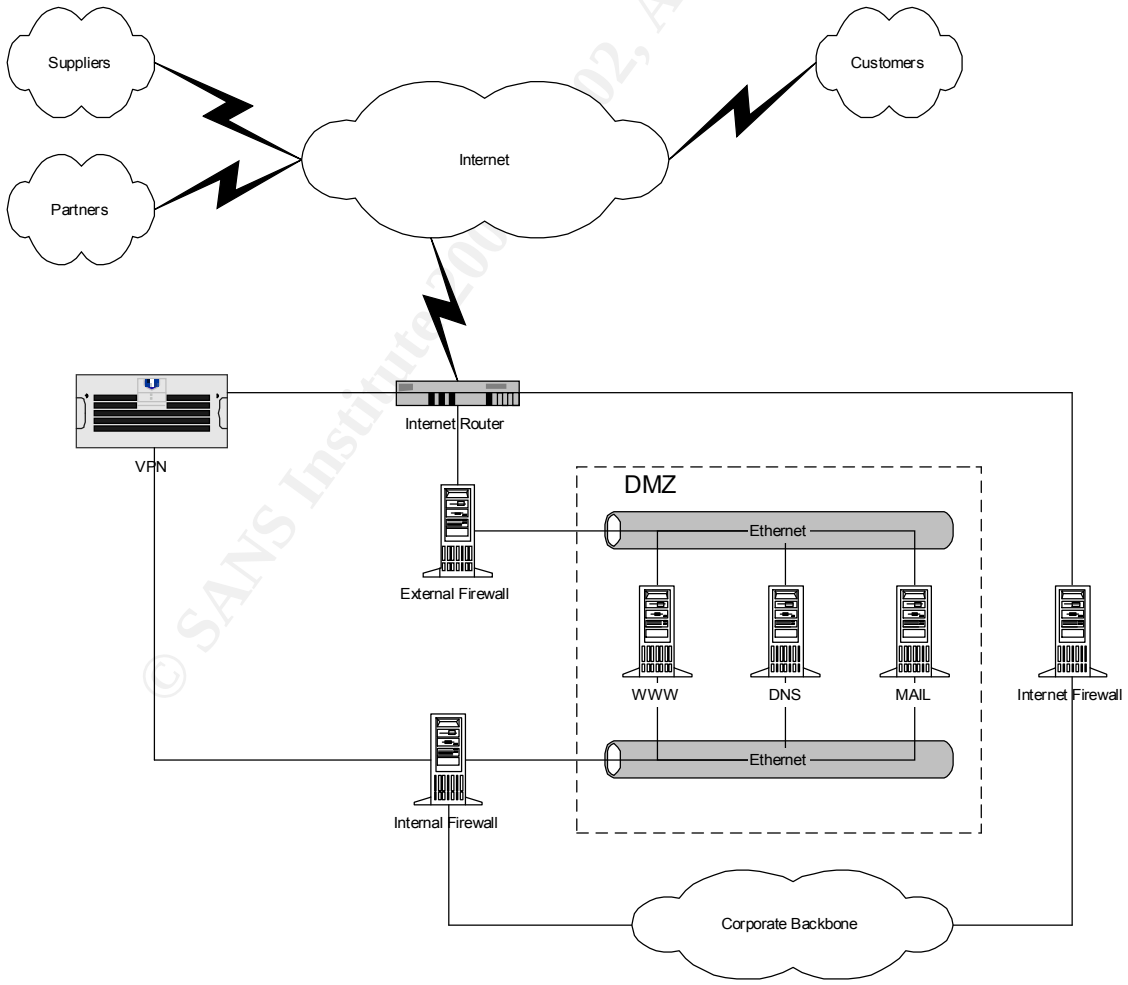
Assignment 1 - Security Architecture (25 Points)

Define a security architecture for GIAC Enterprises, a growing Internet startup that expects to earn $200 million per year in online sales of fortune cookie sayings, and which has just completed a merger/acquisition. Your architecture must specify filtering routers, firewalls, VPNs to partners, secure remote access, and internal firewalls. Be explicit about the brand and version of each perimeter defense component. Produce a diagram or set of diagrams with explanatory text that define how to use perimeter technologies to implement your security architecture.

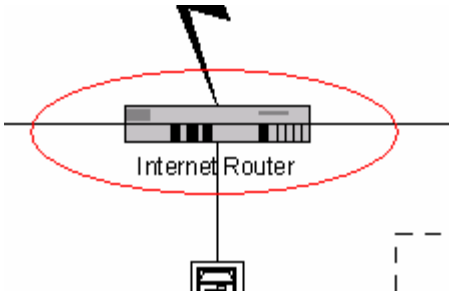
You must consider and define access for:

- Customers (the companies that purchase bulk online fortunes);
- Suppliers (the authors of fortune cookie sayings that connect to supply fortunes);
- Partners (the international partners that translate and resell fortunes).

Network Diagram (See Appendix A for IP Addressing)



Internet Router

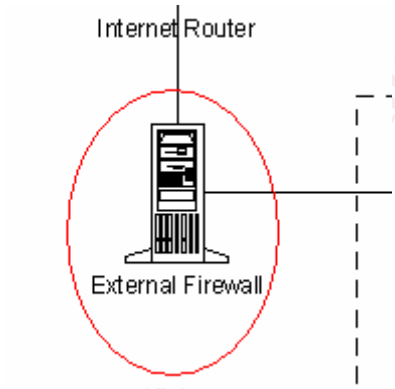


Router: Cisco 3600
Version: IOS 12.0

The Internet router, or border router, is the first device after coming off the Internet and into the company's network. This serves as the first line of defense for all of our network equipment. First, we will use the border router to screen out packets so that the firewall does not have to.

The router contains two interfaces. A serial interface will be connected to the Internet. An Ethernet interface will be connected to a switch that will be connected to the external firewall, the Internet firewall, and the VPN gateway.

External Firewall

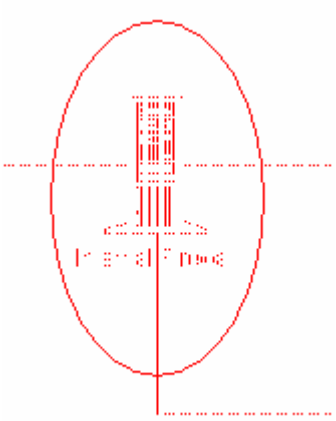


Firewall: Linux Netfilter
Version: Netfilter v1.2, Kernel v2.4.2

The external firewall is our next line of defense. Minimal screening has been performed at the router to filter out the fundamental noise coming from the Internet. Now we will deal with a little more advanced filtering since this is what a firewall was meant to do. The main purpose of this firewall will be to allow customers from the Internet to visit the web server.

This firewall will have two interfaces. One that is considered untrusted and one that is considered trusted. The untrusted interface will be connected to the router. The trusted interface will be connected to an Ethernet segment which contains our web server, DNS server, and mail server. We will call this segment our screened network or DMZ network.

Internal Firewall

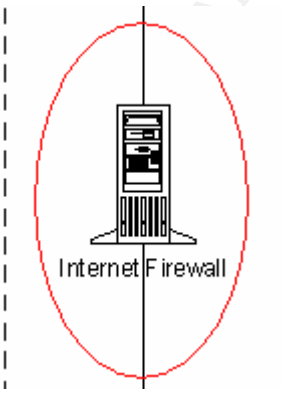


Firewall: Gauntlet
Version: v6.0, Solaris 8

The internal firewall is our next line of defense before getting onto the corporate backbone. This firewall will allow the corporate backbone to access the DMZ network. Also, the VPN gateway will have traffic flowing through this device as well.

This firewall will have three interfaces. Two interfaces will be considered untrusted. One will be connected to the DMZ network and the other will be connected to the VPN gateway. The third interface will be considered trusted and will be connected to the corporate backbone.

Internet Firewall

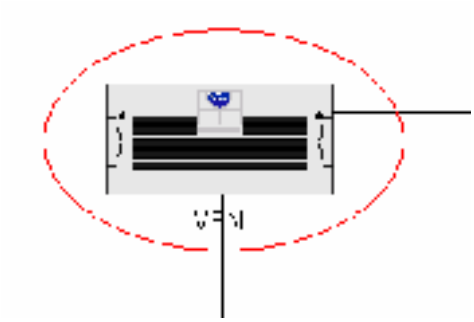


Firewall: Gauntlet
Version: v6.0, Solaris 8

The Internet firewall will act as the corporate gateway to the Internet for functions such as web browsing and FTPing files.

This firewall will have a standard multi-homed configuration with two interfaces. One interface will be considered untrusted. This interface will be connected to the border router. The other interface will be considered trusted and will be connected to the corporate backbone.

VPN Gateway



VPN: Nortel Contivity
Version: v3.5

The VPN device will serve the purpose of allowing the suppliers and partners access to the corporate backbone. Only specific resources that are needed by each party will be accessible. Split tunneling will not be allowed on the client.

The VPN device has two interfaces. One interface will be considered untrusted and will be connected to the Internet. The other interface will be considered trusted. This interface will be connected to the internal firewall.

Assignment 2 - Security Policy (25 Points)

Based on the security architecture that you defined in Assignment 1, provide a security policy for AT LEAST the following three components:

- Border Router
- Primary Firewall
- VPN

You may also wish to include one or more internal firewalls used to implement defense in depth or to separate business functions.

By 'security policy' we mean the specific ACLs, firewall ruleset, IPSec policy, etc. (as appropriate) for the specific component used in your architecture. For each component, be sure to consider internal business operations, customers, suppliers and partners. Keep in mind you are an E-Business with customers, suppliers, and partners - you MAY NOT simply block everything!

(Special note VPNs: since IPSec VPNs are still a bit flaky when it comes to implementation, that component will be graded more loosely than the border router and primary firewall. However, be sure to define whether split-horizon is implemented, key exchange parameters, the choice of AH or ESP and why. PPP-based VPNs are also fully acceptable as long as they are well defined.)

For each security policy, write a tutorial on how to implement each ACL, rule, or policy measure on your specific component. Please use screen shots, network traffic traces, firewall log information, and/or URLs to find further information as appropriate. Be certain to include the following:

1. The service or protocol addressed by the ACL or rule, and the reason these services might be considered a vulnerability.
2. Any relevant information about the behavior of the service or protocol on the network.
3. The syntax of the ACL, filter, rule, etc.
4. A description of each of the parts of the filter.
5. An explanation of how to apply the filter.
6. If the filter is order-dependent, list any rules that should precede and/or follow this filter, and why this order is important. (Note: instead of explaining order dependencies for each individual rule, you may wish to create a separate section of your practical that describes the order in which ALL of the rules should be applied, and why.)
7. Explain how to test the ACL/filter/rule.

Be certain to point out any tips, tricks, or "gotchas".

Internet Router

The Internet is our first piece of network hardware after coming off the Internet. Here is the configuration and access control lists to harden the router.

Configuration

To apply this configuration we first must be in privileged exec mode and global configuration mode. This will be applied to the running configuration. When finished with the configuration it must be saved into NVRAM so no changes will be lost. A copy run start will accomplish this.

```
banner motd #
Systems require authorization; Unauthorized access is illegal.
#
```

- This command will send out this message before login. It is good to have this so that it is clearly stated that it is unlawful to use these systems without authorization.

```
logging 10.10.10.200
```

- This command sends out logging to the syslog server residing on the corporate backbone.

```
service password-encryption
enable secret
```

- Enables a higher level of security on the router passwords.

```
no service tcp-small-servers
no service udp-small-servers
no service finger
no ip bootp server
no ip http server
```

- Disables unused services on the router.

```
no ip direct-broadcast
```

- Prevents malicious directed broadcasts from causing denial of service problems.

```
no ip unreachables
```

- Prevents the router from giving out network information from an ICMP error message.

```
no ip proxy-arp
```

- Disables the router from arp'ing on behalf of another device.

no snmp

- Disables the use of Simple Network Management Protocol.

no ip source-route

- This command disables the use of source routing. Source routing is not desired because we don't want packets "steering" themselves through our network.

Access Control Lists (ACL)

```
access-list 101 permit tcp any host 172.16.1.1 eq 80
access-list 101 permit tcp any host 172.16.1.1 eq 443
access-list 101 permit udp any host 172.16.1.2 eq 53
access-list 101 permit tcp any host 172.16.1.3 eq 25
access-list 101 permit udp any host 192.168.1.1 eq 500
access-list 101 permit esp any host 192.168.1.1
access-list 101 deny ip 10.0.0.0 0.255.255.255 any log
access-list 101 deny ip 172.16.0.0 0.15.255.255 any log
access-list 101 deny ip 192.168.0.0 0.0.255.255 any log
access-list 101 deny ip 127.0.0.0 0.255.255.255 any log
access-list 101 deny ip 224.0.0.0 31.255.255.255 any log
access-list 101 deny ip host 0.0.0.0 any log
access-list 101 deny ip any any
```

- The first two permit filters allows http and ssl access to the web server.
- The third permit filter allows DNS access to the DNS server.
- The fourth permit filter allows Internet key exchange access to the VPN device.
- The fifth permit filter allows IPSec encapsulated security payload access to the VPN device.
- The first three deny filters are to deny RFC1918 addresses from the Internet side of the router and log the connection. These addresses should never reach the external interface unless there is a misconfiguration on the Internet or someone is spoofing their address to try to gain unauthorized access into our network.
- The fourth deny filter is to deny anything coming from the loopback interface of 127.0.0.1 and log it. This address should never reach the external interface unless someone is spoofing.
- The fifth deny filter is to deny any kind of multi-cast and log it. Since we will not be using multi-cast we can filter these addresses out.
- The sixth deny filter is to block the invalid host of 0.0.0.0 and log it.
- The last deny filter is to block everything else. Although there is an implicit deny, this is a nice mental helper.

- These filters will be applied to incoming traffic on the external interface.
    - Order is not that important here except for the deny all. Although, we can start with the more likely to hit addresses and then end with the less likely to hit addresses. This might spare us a few extra CPU cycles on the router. The deny all must be at the end; else the router would be a black hole.

```
access-list 102 permit 222.222.222.0 0.0.0.255 any
access-list 102 deny any any log
```

- The first filter permits our external subnet to access any Internet resource.
- The second filter denies anything else and logs it.
- These filters will be applied to outgoing traffic on the external interface.
    - Order is important here. First we have to allow ourselves out before we apply the second rule, which will deny everything.

```
ip access-group 101 in
ip access-group 102 out
```

- These commands will be placed in the external interface section, most likely the serial interface.
- These commands tell the interface to use the 101 access-list as incoming filtering and to use the 102 access-list as outgoing filtering.
    - Order is not important here.

To make sure everything worked, we should first run the sh run command and make sure the entire configuration made it in. Now we can perform the sh start command and make sure that these two configurations are the same.

External Firewall

The external firewall is the firewall that serves the purpose of defending our DMZ equipment from the Internet. This includes the web server, DNS server, and mail server. IP forwarding will be disabled on the DMZ equipment to increase the DMZ's overall security effectiveness. Here is the script that will be run to setup the iptables (netfilter) ruleset.

Firewall Rules

```
#!/bin/sh
#
# /etc/rc.d/rc.firewall
#
# Firewall policy script
```

- First we start off with a shell script header that includes a description of this script.

```
#
```

```
#
FW=/usr/local/sbin/iptables
```

- This just gives a shortcut to the iptables command.

```
#
#
$FW -F INPUT
$FW -F OUTPUT
$FW -F FORWARD
$FW -F PREROUTING
$FW -F POSTROUTING
```

- These rules flush out the existing tables' rulesets. The INPUT table handles the incoming connections and the OUTPUT table handles the outgoing connections. The FORWARD, PREROUTING, and POSTROUTING deals with routing and NAT'ing.

```
#
#
$FW -P INPUT ACCEPT
$FW -P OUTPUT ACCEPT
$FW -P FORWARD DROP
$FW -P PREROUTING DROP
$FW -P POSTROUTING DROP
```

- These rules provide a default policy for the five built-in tables. If a packet does not hit one of the rules in these tables it will use the default policy to determine what to do.
- You would expect to see a default policy of drop on the INPUT table. However, there is a gotcha attached to this.
  - When you want to restart the rulebase remotely, you have to flush your INPUT table. Now there is no rule to allow your remote connection and the default policy is to drop. Your connection is now terminated and now the only way to re-install the rulebase is to go to the console.

```
#
#
$FW -A INPUT -i eth0 -s 192.168.0.0/16 -j LOG --log-prefix " --SPOOF-- "
$FW -A INPUT -i eth0 -s 192.168.0.0/16 -j DROP
$FW -A INPUT -i eth0 -s 172.16.0.0/12 -j LOG --log-prefix " --SPOOF-- "
$FW -A INPUT -i eth0 -s 172.16.0.0/12 -j DROP
$FW -A INPUT -i eth0 -s 10.0.0.0/8 -j LOG --log-prefix " --SPOOF-- "
$FW -A INPUT -i eth0 -s 10.0.0.0/8 -j DROP
```

- These rules will block any spoofing on the external interface of the firewall. These rules should not be hit because the router is doing anti-spoofing. However, there might be a machine between the router and the firewall that could be spoofing. We must account for this as well with these rules.

```
#
#
```

```
$FW -A INPUT -i eth1 -s 172.16.1.0/24 -d 0/0 -j ACCEPT
$FW -A INPUT -i lo -s 127.0.0.1/32 -d 0/0 -j ACCEPT
```

- These rules allow connections from our DMZ and our loopback device.

```
#
#
$FW -A FORWARD -j ACCEPT -m state --state ESTABLISHED,RELATED
```

- This rule allows connections to be stateful.

```
#
#
$FW -A INPUT -p udp -s 172.16.1.2/32 --sport 53 -d 0/0 -j ACCEPT
```

- This rule allows nameserver lookups to our DNS server.

```
#
#
$FW -A INPUT -i eth0 -p tcp -s 0/0 -d 0/0 --dport 80 -j ACCEPT
$FW -A INPUT -i eth0 -p tcp -s 0/0 -d 0/0 --dport 443 -j ACCEPT
$FW -A INPUT -i eth0 -p udp -s 0/0 -d 0/0 --dport 53 -j ACCEPT
$FW -A INPUT -i eth0 -p tcp -s 0/0 -d 0/0 --dport 25 -j ACCEPT
```

- These rules allow http, ssl, DNS, and mail to be accepted on the external interface. This will setup the first part of redirecting these services to the appropriate servers in our DMZ.

```
#
#
$FW -t nat -A PREROUTING -p tcp -d 222.222.222.110 --destination-port 80 -i eth0 -j DNAT --to 172.16.1.1
$FW -t nat -A PREROUTING -p tcp -d 222.222.222.110 --destination-port 443 -i eth0 -j DNAT --to 172.16.1.1
$FW -t nat -A PREROUTING -p udp -d 222.222.222.110 --destination-port 53 -i eth0 -j DNAT --to 172.16.1.2
$FW -t nat -A PREROUTING -p tcp -d 222.222.222.110 --destination-port 25 -i eth0 -j DNAT --to 172.16.1.3
```

- These rules redirect the packets into the appropriate servers in our DMZ.

```
#
#
$FW -A INPUT -i eth0 -s 0/0 -d 0/0 -j LOG --log-prefix " --DENY-- "
$FW -A INPUT -i eth0 -s 0/0 -d 0/0 -j DROP
```

- These rules are the last rules in the script and will deny everything else and log it. This also allows us to avoid the connection termination when re-installing the rulebase and still have a default deny all.

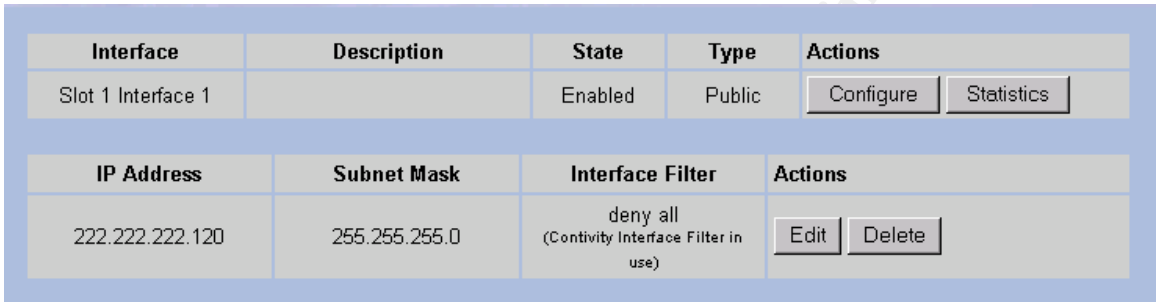
All these rules are order dependant. First, we start with flushing out our old rules so that we start with clean tables. Next, before anything else we want to filter out spoofed packets. Then we add rules in to allow valid packets to do certain tasks. Finally, we end up with a deny all.

To make sure everything worked we will run the iptables –L and iptables –t nat –L commands.

VPN Gateway

The VPN gateway will primarily be used to allow the suppliers and partners to access resources they need on the corporate backbone. A firewall module on this device will add security. We will utilize just the ESP protocol instead of AH so our payload is encrypted with 3DES. We will add SHA1 hashing so that we can verify the packets as they come into the device.

External Interface

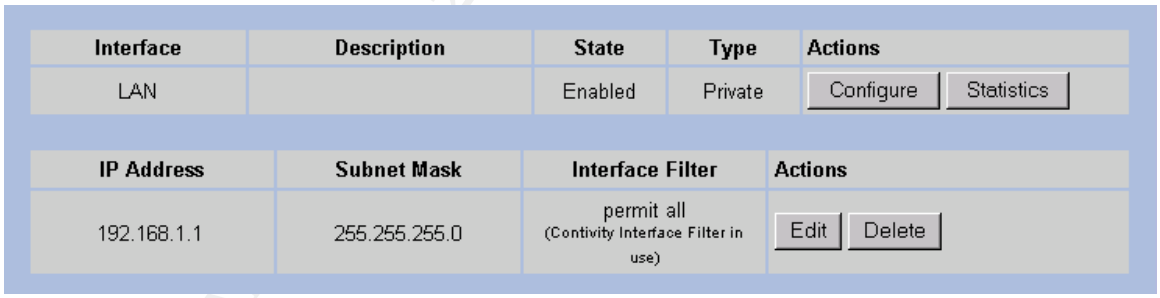


Interface	Description	State	Type	Actions
Slot 1 Interface 1		Enabled	Public	Configure Statistics

IP Address	Subnet Mask	Interface Filter	Actions
222.222.222.120	255.255.255.0	deny all (Contivity Interface Filter in use)	Edit Delete

- A deny all filter is placed on this interface to deny everything.
    - However, this does not include the IKE and IPSec protocols. These protocols are allowed by default.

Internal Interface



Interface	Description	State	Type	Actions
LAN		Enabled	Private	Configure Statistics

IP Address	Subnet Mask	Interface Filter	Actions
192.168.1.1	255.255.255.0	permit all (Contivity Interface Filter in use)	Edit Delete

- A permit all is placed on this interface to allow everything on the trusted side. We don't have to worry too much about this side because it is on the trusted side and there is a firewall in between.

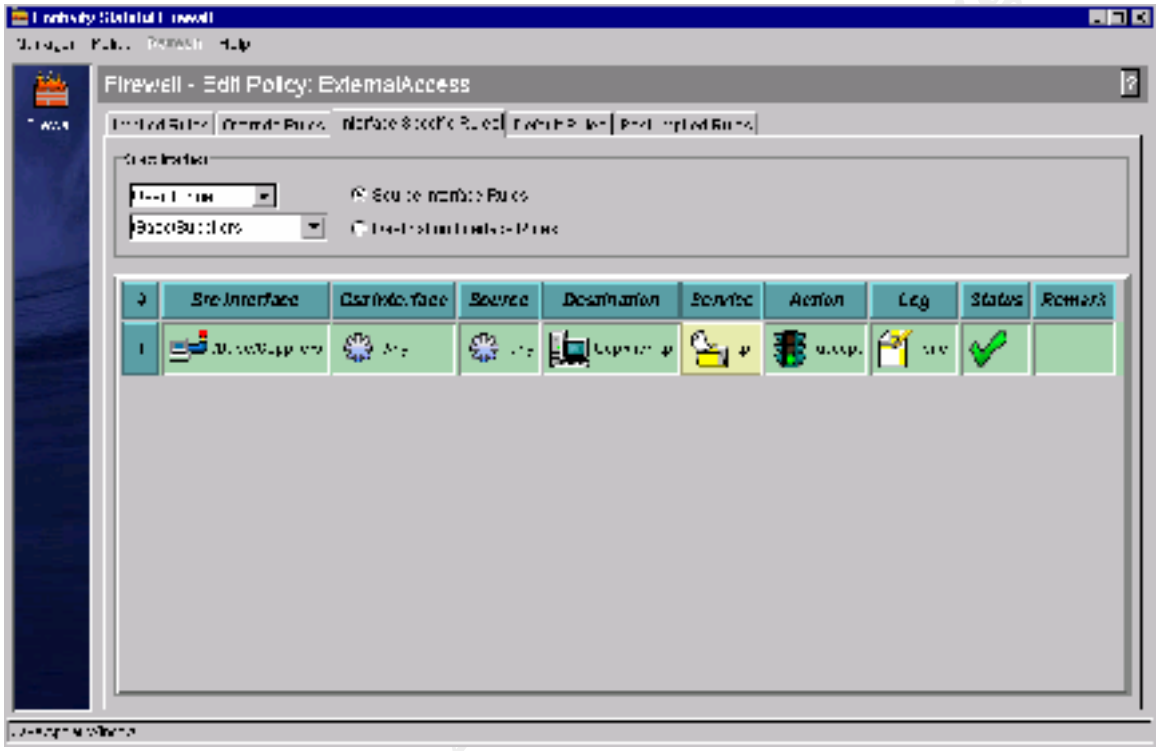
Ruleset

A group will be made for both the suppliers and the partners. Each group will have a different policy on what they are able to access.

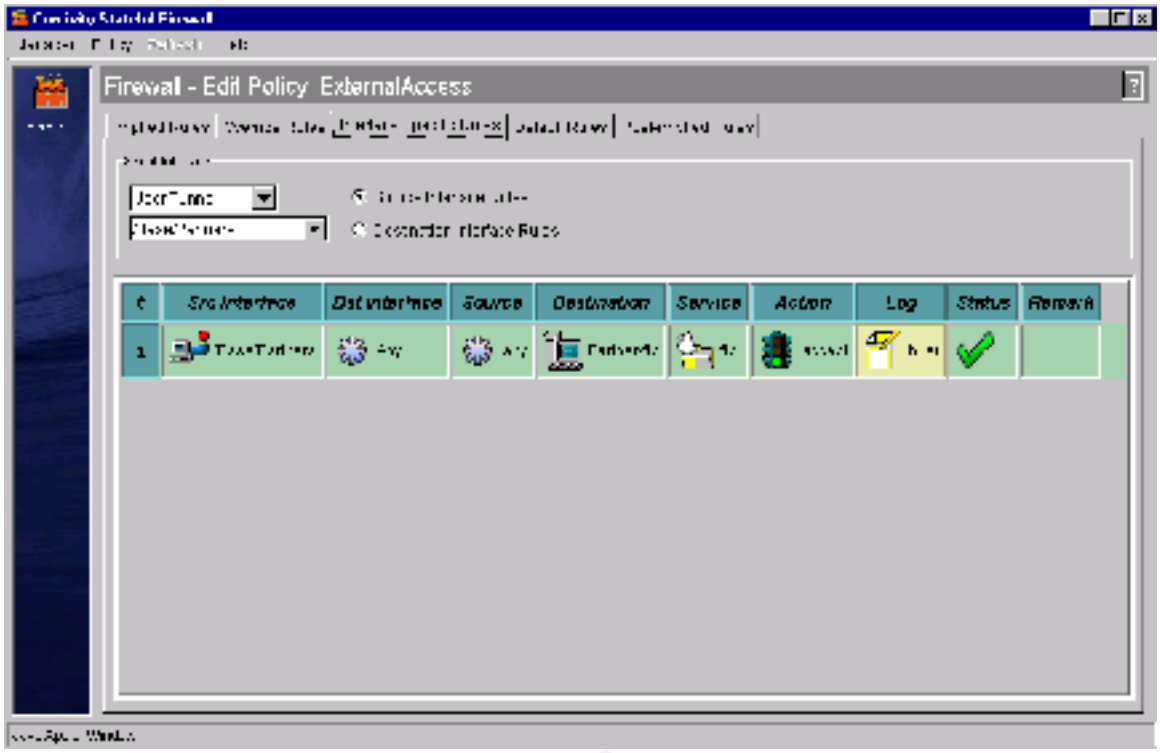
Suppliers will need to access an FTP server that resides on the corporate backbone. This is where the suppliers will transmit the fortunes for later retrieval by GIAC Enterprises.

Partners will need to access a different FTP server on the corporate backbone so they can retrieve fortunes that GIAC Enterprises is ready to sell in the different markets.

Here is the ruleset that will be applied to the firewall module on the VPN device.



- This rule will allow the suppliers' VPN tunnel to only access the supplier ftp server.



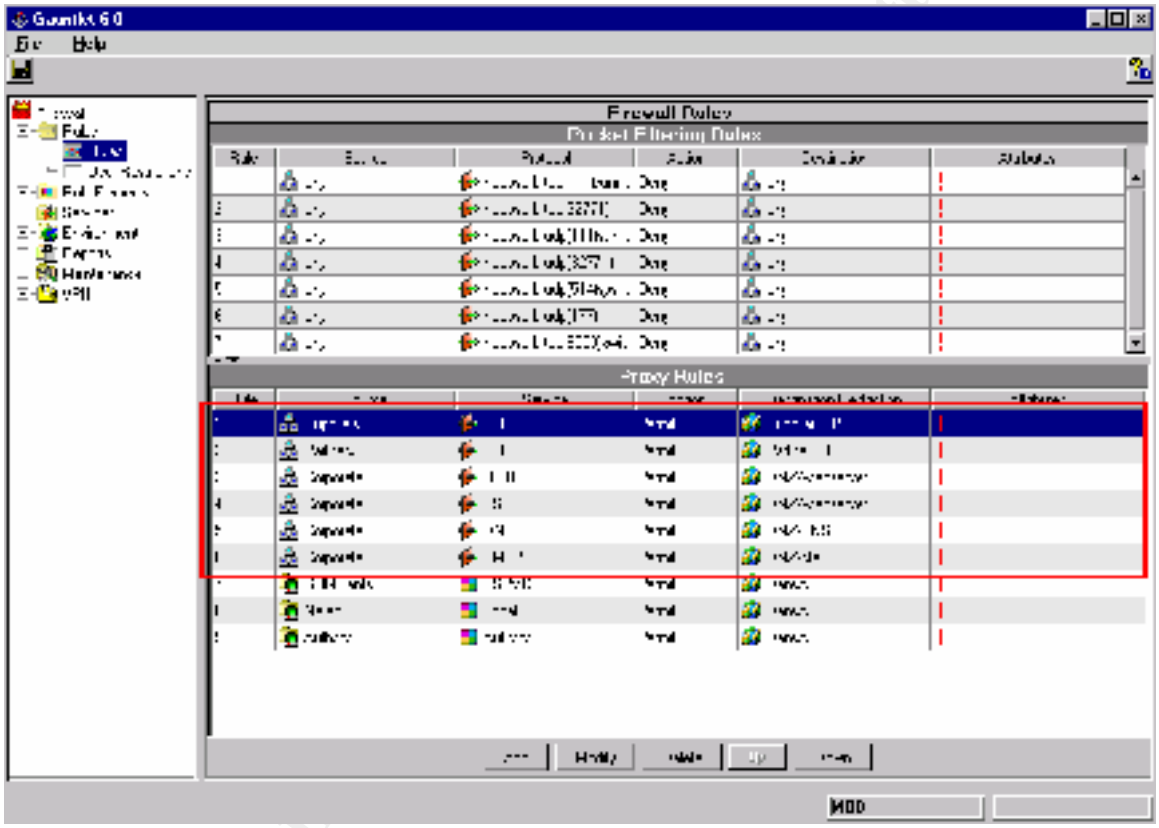
- This rule will allow the partners' VPN tunnel to only access the partner ftp server.
- A default deny is implied at the end of both of these rules.

To make sure everything worked we will need to add a test user to the suppliers' and partners' group and make sure the group can only access their respective ftp server from an Internet VPN tunnel. After the test we will remove this user to prevent unauthorized use.

Internal Firewall

The internal firewall will serve the primary purpose of defending our DMZ equipment from our corporate backbone. It will also defend our network from VPN users coming into the corporate network and DMZ. Here is the ruleset that will be applied to this firewall.

Firewall Rules



- The first six rules are the rules we want to implement, the last three are the ones that come defined with the product. The packet filters above are also all defaults, which block some well-known exploitable ports.
    - The first rule will allow the suppliers to access their ftp server once they come out of the VPN device.
    - The second rule will allow the partners to access their ftp server once they come out of the VPN device.
    - The third, fourth, fifth, and sixth rule will allow the corporate backbone to access the http/ssl, DNS, and mail servers in the DMZ, respectively.

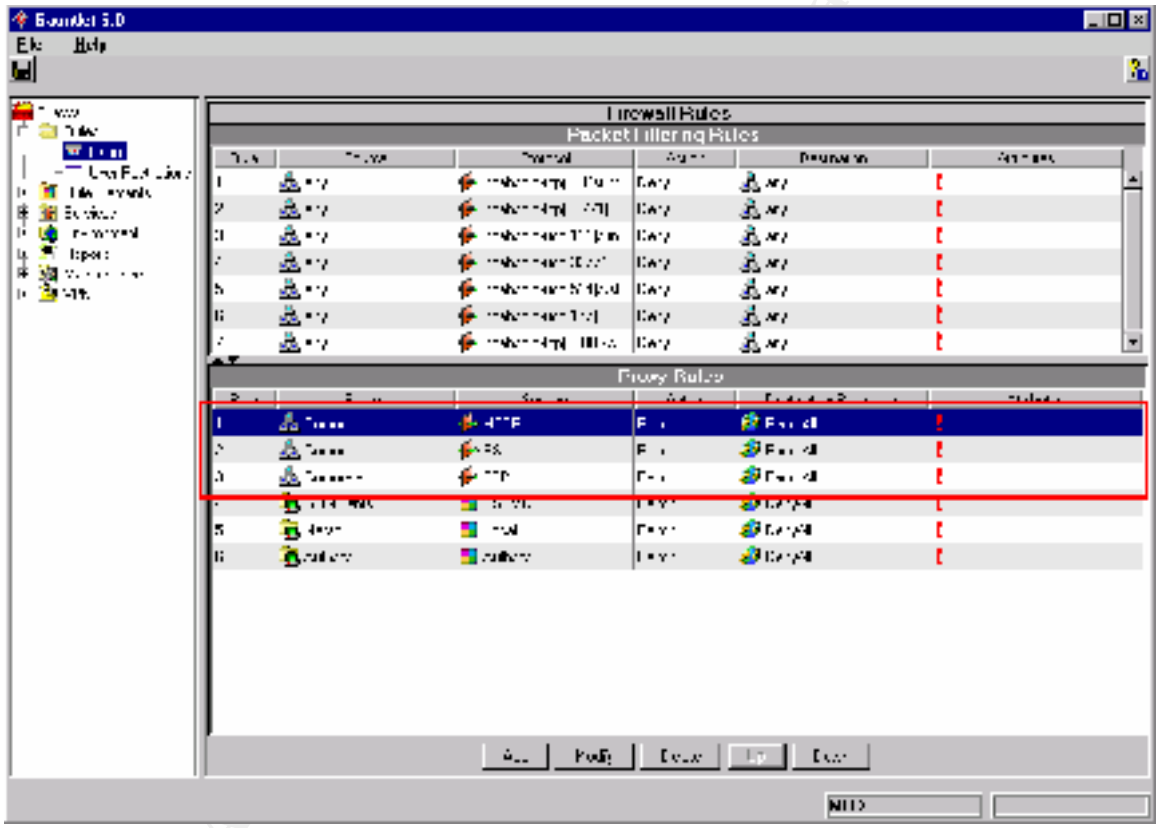
It is best to define our user-defined rules and place them at the top. However, these six rules are interchangeable.

To make sure everything worked we can run the netstat –an | grep LISTEN command to make sure proxy ports 80, 443, 21, 53, and 25 are listening.

Internet Firewall

The Internet firewall will serve the purpose of defending our corporate backbone from the Internet. This firewall will be the Internet gateway for the corporate backbone to access resources out on the Internet. Here is the ruleset that will be applied to this firewall.

Firewall Rules



- The first three rules are the rules we want to implement, the last three are the ones that come defined with the product.
    - The first rule will allow the corporate backbone of 10.* to access the Internet via http.
        - For simplicity sake we've defined 10.* for corporate, when in actuality we would need to define a subnet that we would want to access the Internet.
    - The second rule will allow the corporate backbone to access the Internet via ssl.

- The third rule will allow the corporate backbone to access the Internet via ftp.

It is best to define our user-defined rules and place them at the top. However, these three rules are interchangeable.

To make sure everything worked we can run the netstat –an | grep LISTEN command to make sure proxy ports 80, 443, and 21 are listening.

Assignment 3 - Audit Your Security Architecture (25 Points)

You have been assigned to provide technical support for a comprehensive information systems audit for GIAC Enterprises. You are required to audit the Primary Firewall described in Assignments 1 and 2. Your assignment is to:

1. Plan the assessment. Describe the technical approach you recommend to assess your perimeter. Be certain to include considerations such as what shift or day you would do the assessment. Estimate costs and level of effort. Identify risks and considerations.
2. Implement the assessment. Validate that the Primary Firewall is actually implementing the security policy. Be certain to state exactly how you do this, including the tools and commands used. Include screen shots in your report if possible.
3. Conduct a perimeter analysis. Based on your assessment (and referring to data from your assessment), analyze the perimeter defense and make recommendations for improvements or alternate architectures. Diagrams are strongly recommended for this part of the assignment.

Note: DO NOT simply submit the output of nmap or a similar tool here. It is fine to use any assessment tool you choose, but annotate the output.

Plan the assessment

To accomplish this assessment we will be utilizing the nmap tool. I believe this is one of the best, if not the best tool that can be used to identify vulnerabilities in systems. We want to determine that we have no more available services than we actually need. This is what we're limiting our assessment to. Of course, we could start trying out exploits against the systems. However, I believe as long as we keep up to date in our patches and limit the number of available services we can be secure as possible at that date in time. There WILL be exploits down the road and we need to limit the services that will be attacked as a result.

We will obtain the source code of nmap from *http://www.insecure.org/nmap*. The version that we will use will be v2.54BETA22.

We will be using a Linux notebook so we can have maximum mobility. This will allow us to move around to different systems in different areas easily. To perform this audit, it should only take one competent person and should not cost more than the salary that is paid to this person.

The time when we do this is very critical. Since we cannot afford to be down for any amount of time we need to pick a time when traffic is at a minimum. Usually this will be

Saturday night around midnight. Even though we probably won't take any systems down while performing the scan, it is best that we don't take a chance and do this on off-hours.

Implement the assessment

External Firewall

Nmap scan results

TCP

```
linux:~/nmap-2.54BETA22 # ./nmap -sS -P0 -v 222.222.222.110

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host (222.222.222.110) appears to be up ... good.
Initiating SYN Stealth Scan against (222.222.222.110)
Adding TCP port 443 (state open).
Adding TCP port 80 (state open).
Adding TCP port 25 (state open).
The SYN Stealth Scan took 733 seconds to scan 1542 ports.
Interesting ports on (222.222.222.110):
(The 1539 ports scanned but not shown below are in state: filtered)
Port       State       Service
25/tcp     open        smtp
80/tcp     open        http
443/tcp    open        https

Nmap run completed -- 1 IP address (1 host up) scanned in 734 seconds
```

UDP

```
linux:~/nmap-2.54BETA22 # ./nmap -sU -P0 -v 222.222.222.110

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host (222.222.222.110) appears to be up ... good.
Initiating UDP Scan against (222.222.222.110)
The UDP Scan took 1755 seconds to scan 1453 ports.
(no udp responses received -- assuming all ports filtered)
All 1453 scanned ports on (222.222.222.110) are: filtered
```

Nmap run completed -- 1 IP address (1 host up) scanned in 1755 seconds

- The TCP scan identified that the smtp, http, and https(ssl) services are available. This is the TCP ports that we want listening, and only these TCP ports.
- When performing the UDP scan, nmap said all udp ports are filtered. Well this is not what we wanted to see. We need to go deeper to make sure that UDP 53 and only UDP 53 is listening.
    - On the DNS server itself we can perform a netstat –an | grep udp command. This command produces this output…

```
udp     0    0 127.0.0.1:53        0.0.0.0:*
udp     0    0 172.16.1.2:53       0.0.0.0:*
udp     0    0 172.17.1.2:53       0.0.0.0:*
```

    - This output signifies that we only are using DNS on our loopback interface (127.0.0.1), our outside interface (172.16.1.2), and our inside interface (172.17.1.2).
- When performing these two scans log entries were generated on the firewall. Here is a sample output…

```
Mar 11 00:52:34 firewall kernel:  --DENY-- IN=eth0 OUT= MAC=00:e0:7d:00:94:5c:00:00:c0:d9:1a:74:08:00
SRC=222.222.222.254 DST=222.222.222.110 LEN=40 TOS=0x00 PREC=0x00 TTL=37 ID=18611 PROTO=TCP
SPT=46776 DPT=7597 WINDOW=2048 RES=0x00 SYN URGP=0

Mar 11 00:52:34 firewall kernel:  --DENY-- IN=eth0 OUT= MAC=00:e0:7d:00:94:5c:00:00:c0:d9:1a:74:08:00
SRC=222.222.222.254 DST=222.222.222.110 LEN=40 TOS=0x00 PREC=0x00 TTL=37 ID=61593 PROTO=TCP
SPT=46776 DPT=527 WINDOW=2048 RES=0x00 SYN URGP=0

Mar 11 00:52:34 firewall kernel:  --DENY-- IN=eth0 OUT= MAC=00:e0:7d:00:94:5c:00:00:c0:d9:1a:74:08:00
SRC=222.222.222.254 DST=222.222.222.110 LEN=40 TOS=0x00 PREC=0x00 TTL=37 ID=2104 PROTO=TCP
SPT=46776 DPT=2064 WINDOW=2048 RES=0x00 SYN URGP=0

Mar 11 00:52:34 firewall kernel:  --DENY-- IN=eth0 OUT= MAC=00:e0:7d:00:94:5c:00:00:c0:d9:1a:74:08:00
SRC=222.222.222.254 DST=222.222.222.110 LEN=40 TOS=0x00 PREC=0x00 TTL=37 ID=12792 PROTO=TCP
SPT=46776 DPT=85 WINDOW=2048 RES=0x00 SYN URGP=0

Mar 11 00:52:34 firewall kernel:  --DENY-- IN=eth0 OUT= MAC=00:e0:7d:00:94:5c:00:00:c0:d9:1a:74:08:00
SRC=222.222.222.254 DST=222.222.222.110 LEN=40 TOS=0x00 PREC=0x00 TTL=37 ID=9710 PROTO=TCP
SPT=46776 DPT=618 WINDOW=2048 RES=0x00 SYN URGP=0
```

- Interesting points to note about these logs.
    - Nmap by default doesn't change its source port (SPT).
    - Nmap doesn't do an ordinal scan. Meaning that the destination ports being tried are random rather than 1,2,3,etc…

Internet Firewall

Nmap scan results

TCP

```
linux:~/nmap-2.54BETA22 # nmap -sS -P0 -v 222.222.222.130

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host  (222.222.222.130) appears to be up ... good.
Initiating SYN Stealth Scan against  (222.222.222.130)
Adding TCP port 113 (state open).
The SYN Stealth Scan took 6 seconds to scan 1542 ports.
Interesting ports on  (222.222.222.130):
(The 1538 ports scanned but not shown below are in state: closed)
Port       State       Service
111/tcp    filtered    sunrpc
113/tcp    open        auth
6000/tcp   filtered    X11
32771/tcp  filtered    sometimes-rpc5


Nmap run completed -- 1 IP address (1 host up) scanned in 6 seconds
```

UDP

```
linux:~/nmap-2.54BETA22 # ./nmap -sU -P0 -v 222.222.222.130

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host  (222.222.222.130) appears to be up ... good.
Initiating UDP Scan against  (222.222.222.130)
Too many drops ... increasing senddelay to 50000
The UDP Scan took 176 seconds to scan 1453 ports.
Interesting ports on  (222.222.222.130):
(The 1448 ports scanned but not shown below are in state: closed)
Port       State       Service
111/udp    open        sunrpc
177/udp    open        xdmcp
514/udp    open        syslog
32771/udp  open        sometimes-rpc6
```

32772/udp open sometimes-rpc8

Nmap run completed -- 1 IP address (1 host up) scanned in 176 seconds

- This TCP scan shows that we have auth open and sunrpc, X11, and sometimes-rpc5 filtered.
    - Auth is used to get some identification from the device that is connecting. By default, Gauntlet leaves this open.
    - The other three services are being filtered by Gauntlet by default. These are actually being denied on all interfaces.
- The UDP scan shows that sunrpc, xdmcp, syslog, sometimes-rpc6, and sometimes-rpc8 are all available.
    - All these services have been known to be easily exploitable so special care has to be taken to filter these. Gauntlet has filters in place to deny these services on any interface.
        - The way UDP works makes nmap report that these ports are open instead of filtered.
- Here is a sample output of the logs that were generated…

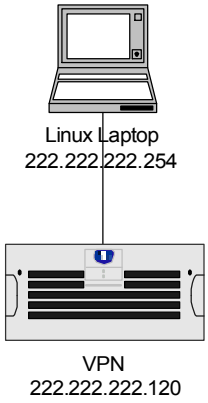
Mar 14 00:59:40 firewall gfw: [ID 702911 kern.info] securityalert: udp if=hme0 from 222.222.222.254:54578 to 222.222.222.130 on unserved port 1446

Mar 11 00:59:42 firewall gfw: [ID 702911 kern.info] securityalert: packet denied by local screen: UDP if=hme0 srcaddr=222.222.222.254 srcport=54579 dstaddr=222.222.222.130 dstport=111

Mar 11 00:59:43 firewall gfw: [ID 702911 kern.info] securityalert: packet denied by local screen: UDP if=hme0 srcaddr=222.222.222.254 srcport=54579 dstaddr=222.222.222.130 dstport=32771

- Some interesting notes about these log entries.
    - The first securityalert says that a device was trying to attach to an unserved or unavailable service.
    - The last securityalert's say that a device was trying to attach to a service that is currently being filtered.

VPN Gateway



Nmap scan results

TCP

```
linux:~/nmap-2.54BETA22 # ./nmap -sS -vv -P0 222.222.222.120

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host (222.222.222.120) appears to be up ... good.
Initiating SYN Stealth Scan against (222.222.222.120)
The SYN Stealth Scan took 2239 seconds to scan 1542 ports.
All 1542 scanned ports on (222.222.222.120) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 2239 seconds
```

UDP

```
linux:~/nmap-2.54BETA22 # ./nmap -sU -v -P0 222.222.222.120

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host (222.222.222.120) appears to be up ... good.
Initiating UDP Scan against (222.222.222.120)
The UDP Scan took 1755 seconds to scan 1453 ports.
(no udp responses received -- assuming all ports filtered)
All 1453 scanned ports on (222.222.222.120) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 1755 seconds
```

- These two scans show basically that no services were open on the VPN device. However, in order to do IPSec communication you have to do the Internet Key Exchange, which resides on UDP port 500. This did not show up in our scan because the VPN device thought we were performing a DoS (Denial of Service) so it discontinued communication with us.
- The only way we can make sure that only the IKE service is available is to get on the administration screens and disable all other services like ftp and telnet.
- Unfortunately, the Nortel VPN device doesn't log the anti-DoS event.

Conduct perimeter analysis

Security is very tight on our current infrastructure. There are no unneeded services available to the external interfaces of these three external devices.

On the external firewall only http, https(ssl), smtp were found to be available by nmap. Also, after performing a local command on the DNS server found its corresponding domain (UDP 53) port to be available.

On the Internet firewall only auth was found to be available. We can disable this service, but if we do, we need to make sure we make a filter to REJECT this connection so that our connections don't hang. All the other services were reported filtered, so Gauntlet is handling these with filters. We can disable these services on the firewall so we don't have to depend on the filters.

On the VPN device nothing was found to be open thanks to the Anti-DoS feature and filter that are implemented on the device. To better secure the VPN device we could think about placing this device behind a firewall, or even better, a VPN sandwich between two firewalls. Special care needs to be taken since sometimes the NAT'ing that needs to take place can break VPN connections.

Further securing of our Internet infrastructure can include implementing an Intrusion Detection System (IDS). If setup properly, it can give very valuable information about unauthorized connections within our Internet infrastructure.

Assignment 4 - Design Under Fire (25 Points)

The purpose of this exercise is to help you think about threats to your network and therefore develop a more robust design. Keep in mind that the next certification group will be attacking your architecture!

Select a network design from any previously posted GCFW practical (http://www.sans.org/giactc/gcfw.htm) and paste the graphic into your submission. Be certain to list the URL of the practical you are using. Design the following three attacks against the architecture:

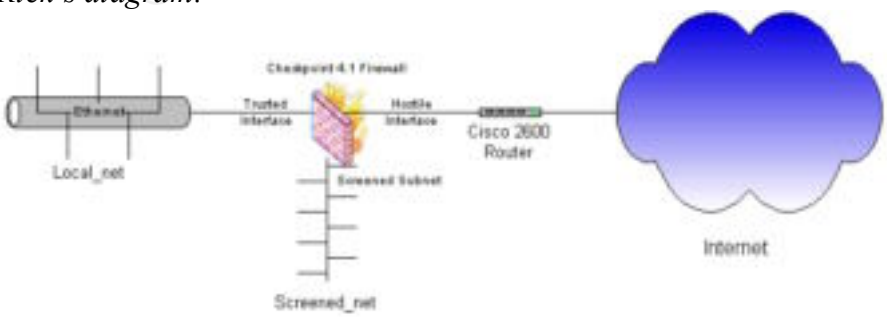
1. An attack against the firewall itself. Research vulnerabilities that have been found for the type of firewall chosen for the design. Choose an attack and explain the results of running that attack against the firewall.
2. A denial of service attack. Subject the design to a theoretical attack from 50 compromised cable modem/DSL systems using TCP SYN, UDP, or ICMP floods. Describe the countermeasures that can be put into place to mitigate the attack that you chose.
3. An attack plan to compromise an internal system through the perimeter system. Select a target, explain your reasons for choosing that target, and describe the process to compromise the target.

Note: this is the second time this assignment has been used. The first time, a number of students came up with magical "hand-waving" attacks. You must supply documentation (preferably a URL) for any vulnerability you use in your attack, and the exploit code that you use to accomplish the attack. The purpose of this exercise is for the student to clearly demonstrate they understand that firewall and perimeter systems are not magic "silver bullets" immune to all attacks

Attack the Firewall

For this attack, I chose Rick Dreger's design. His design can be found here: *http://www.sans.org/y2k/practical/Rick_Dreger.doc*. He uses an unspecified service pack level of CheckPoint Firewall-1 v4.1. We will assume he is using v4.1 with no service pack.

Rick's diagram:



To attack Rick's design we'll need to obtain the exploit code. It can be obtained here: *http://darknet.syntheticarmy.com/exploits/os/hardware/firewall-1/cpd.c*. Now we need to compile this with the command cc -o cpd cpd.c. This will create a binary that we can now execute and use.

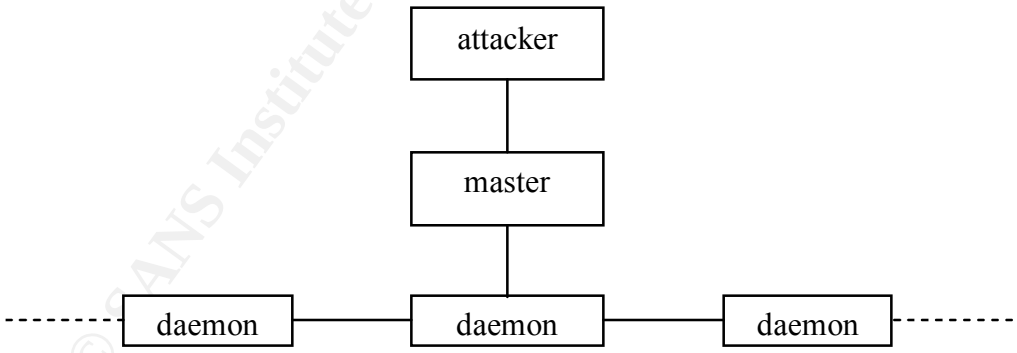
This exploit will concentrate on spoofing UDP packets sent to the firewall. Sending these spoofed packets where the source address equals the destination causes CheckPoint to panic and makes the system lock up. This can cause a lot of problems since this firewall is acting an Internet gateway. All Internet access has now just disappeared and there will be a lot of angry people complaining that the Internet is down.

Denial of Service

In this attack, we have compromised 50 cable modem/DSL systems. We'll assume these systems have been compromised and had trinoo installed on them. Source code for this exploit can be obtained here: *ftp://ftp.technotronic.com/denial/trinoo.tgz*. This is a distributed denial of service attack. It uses many clients' bandwidth to flood a target system and render its services and resources useless.

When you download this package you receive source code for machines that will be master(s) and daemons. A master will be what controls a specified group of daemons. Daemons are like drones awaiting orders from the master. For simplicity, we'll use a single master system.

Here is the logical diagram of what our trinoo network looks like:



Trinoo passwords have been compiled in to the daemon running on both the master and daemon systems. This protects us against other hackers, etc. from accessing our trinoo network.

To start the attack we authenticate to the master system and send the "dos <ip address>" command. This will cause the daemon systems to simultaneously send packets to the

target ip address. This will flood the target system and cause the services to be inaccessible.

To mitigate this attack, we must use a combination of rate limiting and emergency data backups. First, there is a feature on Cisco routers where you can limit the amount of network traffic being utilized. And secondly, an offsite data center can be brought online to keep resources available.

Attack an Internal System

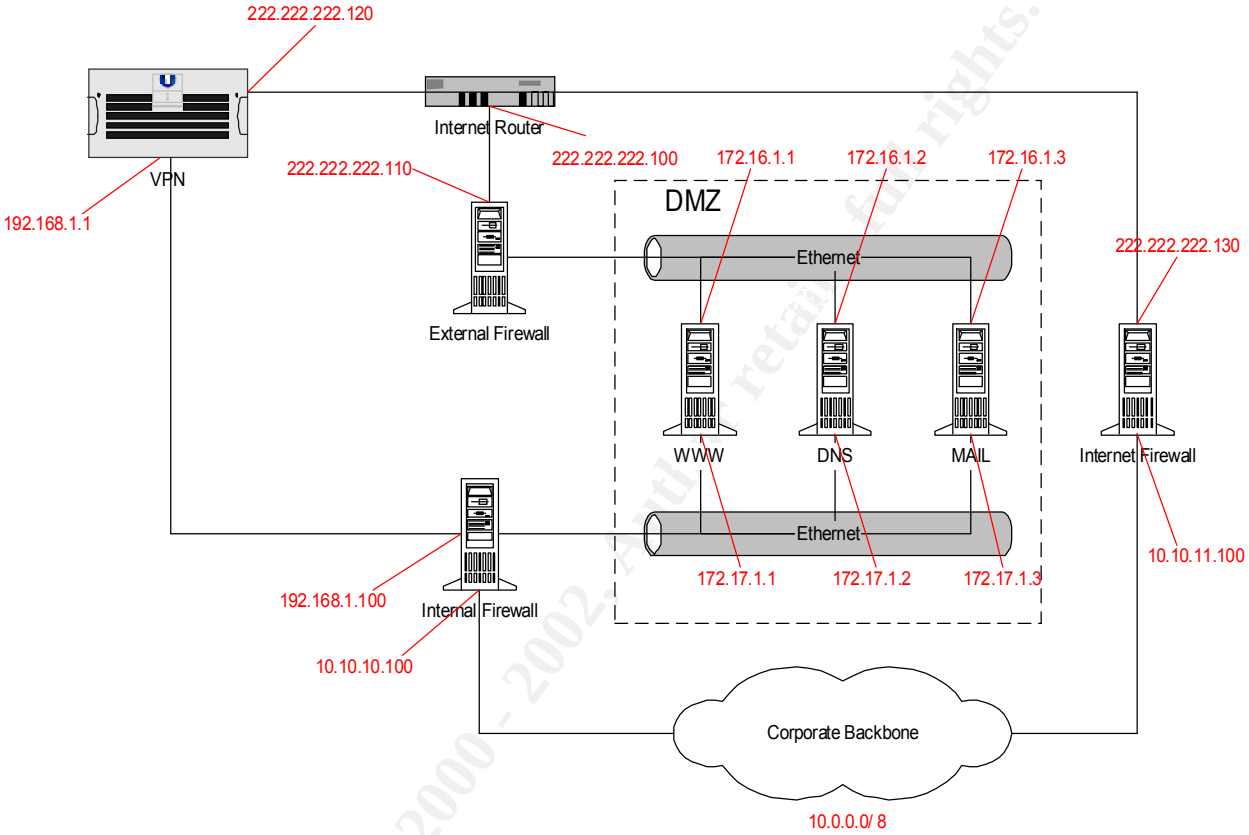
In this attack, we'll go after the web server. We'll assume the web server is Microsoft IIS 5. IIS is notorious for being vulnerable. Many businesses rely on web servers to distribute information about the company. If this is unavailable, customers might be hesitant to do business with this company.

To launch this attack we must obtain the source code for the exploit. The exploit can be obtained here: *http://www.securityfocus.com/data/vulnerabilities/exploits/vv5.pl.* Once this is obtained we can run it from any perl-enabled machine. This program just needs to be run with the ip and port of the web server.

This exploit will send malformed requests to the web server. IIS does not handle these requests very well. If this exploit is run against this server it causes the server to stop responding and could make it restart all the IIS services.

Appendix A

IP Addressing



References

1. Security Focus, http://www.securityfocus.com

2. Exploit Archive, http://www.hack.co.za

3. SANS Security Course Material [New Orleans 2001], http://www.sans.org

4. Trinoo Analysis, http://staff.Washington.edu/dittrich/misc/trinoo.analysis