Firewalls, Perimeter Protection, and VPNS

GCFW Practical Assignment Version 1.6a

San Diego SANS October 15, 2001

Asad Alsader

Table of Contents

Assignment 1 – Security Architecture

General Overview:

GIAC's security architecture is fairly simple but secured. There are two major ways for information to flow through the company:

1. Employees are allowed to access the Internet through the firewall GIACNET. They are also allowed to access servers in the DMZ through the firewall GIACIDZ. Telecommuters and administrators can access the internal network though the VPN device GIACVPN when they are off the premises.

2. Customers, Suppliers, and Partners are allowed access from the Internet to the DMZ through the firewall GIACEDZ. This access is needed to get to the web and FTP servers. One of the suppliers connects to GIAC's internal network through a private T1 line through the firewalls GIACEXV1 and GIACEXV2.

Detailed Design:

Following is a list of all the components defined in Figure 1 (on page 8) and explanation of each:

GIACBR:

This is the border router. It is a Cisco 3620 with IOS 12.2. An ACL on the router implements basic security features such as preventing spoofed traffic from entering GIAC.

GIACEDZ

This is the firewall that protects the DMZ from the Internet. This is a Gauntlet 5.5 firewall running on an R390 HP server. The operating system is HPUX 10.20. This firewall allows HTTP, HTTPS, FTP, DNS, and IPSEC traffic from the Internet to the DMZ. It also allows syslog traffic from the border router to the syslog server in the DMZ. The firewall has two interfaces, one is connected to the Internet and the other is connected to the DMZ. This is the firewall that allows customers, suppliers, and partners to gain access to the web and FTP servers. It also allows telecommuters, administrators, and partners to gain access to the internal network through the VPN device.

GIACNET

This is the firewall that allows GIAC's employees to access the Internet. This is a Gauntlet 5.5 firewall running on an R390 HP server. The operating system is HPUX 10.20. This firewall allows HTTP, HTTPS, SMTP, and SSH traffic to the Internet. The firewall has two interfaces, one is connected to the Internet and the other is connected to the internal network. Only GIACPRXY is allowed to talk to this firewall for HTTP and SSL traffic. Only GIACMAIL is allowed to talk to this firewall for SMTP traffic. The firewall also accepts mail from the Internet and relays it to GIACMAIL for final delivery. For security reasons, this firewall blocks Java and ActiveX. JavaScript is allowed. Anti-virus feature provided in the HTTP and SMTP proxies is enabled.

GIACWWW

This is the GIAC web server. It is a Netscape Enterprise Server version 3.61 running on an IBM RS/6000 model F-50 server. Customers, suppliers, and partners access this server through HTTP and SSH to execute transactions such as buying, supplying, or reselling fortunes. The web server in turns accesses the database server on the internal network to complete the transactions.

GIACFTP

This is the FTP server that customers, suppliers, and partners can ftp to and from it. It does not allow anonymous ftp. This is a D380 HP server running HPUX 11.0.

GIACEDNS

This is the External DNS server. This is an A180c HP server running a chrooted bind 9.1.3. The concept of split DNS is implemented. This DNS server is setup to answer to the world with information about the GIAC domain. It also resolves Internet names for the Internal DNS server. No internal information is available on this server.

GIACVPN

This is the VPN device. It is a Nortel Contivity 4600 version 3.65.05. The VPN has two interfaces; one connected to the DMZ and the other is connected to the service network on GIACIDZ.

GIACIDZ

This is the firewall that protects GIAC from the DMZ. This is a CheckPoint Firewall-1 version 4.1 firewall running on an R390 HP server. The operating system is HPUX 11.0. This firewall allows HTTP, HTTPS, FTP, and DNS traffic to the DMZ from the Internal network. It also allows traffic on port TCP 1521 so the web server GIACWWW can talk to the Oracle database server GIACDB which resides on the internal network. Specific admin workstations are allowed to SSH to DMZ servers. This firewall has three interfaces. One is connected to the DMZ, one is connected to the VPN device, and one is connected to the internal network.

GIACDB

This is the Oracle database server. It is an R390 HP server running HPUX 10.20. This is a database of all the fortune cookie sayings. Internal employees can access this server. Also, GIACWWW is the only server in the DMZ that is allowed to talk to this server.

GIACIDNS

This is the Internal DNS server. This is an A180c HP server with HPUX 10.20 running a chrooted bind 9.1.3. This DNS server holds the information for the internal GIAC network only. It contacts GIACEDNS server to resolve any Internet names.

GIACMAIL

This is the Internal Exchange mail server. It handles all the internal mail for GIAC employees. Any Internet mail is forwarded to GIACNET firewall for delivery.

GIACEXV1

This is the firewall that protects the internal network from the supplier network. This is a Gauntlet 5.5 firewall running on an R390 HP server. The operating system is HPUX 10.20. The firewall allows machines on the internal network of the supplier to talk to the database server GIACDB over port TCP 1521. The connection is going over a leased private T1 line. The firewall has two interfaces, one is connected to the internal network and the other is connected to the supplier network.

GIACEXV2

Same as GIACEXV1. This firewall is used for load balancing the traffic with GIACEXV1.

GIACFP1

This is the Fire Proof device from Radware version 1.32. It is a load balancing and fail over device used to load balance the traffic initiating from GIAC to the supplier. It divides the traffic evenly between the firewalls GIACEXV1 and GIACEXV2. It's also used to shift all traffic to one of the two firewalls when the other one is down. Please refer to Appendix B for explanation on how to setup the Fire Proof.

GIACFP2

Same as GIACFP1 except that it loads balance the traffic initiating from the supplier to GIAC.

GIACPRXY

This is a proxy server that GIAC employees use to access the Internet. It is a Network Appliance NetCache C760s appliance. All browsers at GIAC are setup to proxy to GIACPRXY on port 5000. The proxy server forwards the traffic to the firewall GIACNET on port 80. This is done for a couple of reasons:

a. Caching. The proxy server caches web pages, which improves the response time for the employees when surfing the net.

b. Filtering. The proxy server runs Websense software to minimize Internet abuse.

Miscellaneous Notes & Assumptions:

1. All the servers are installed with the most recent patches for the operating system and the application software.

2. All the servers are hardened. For example, direct ftp and login for generic IDs is disabled. Some processes are disabled such as tftp, klogin, ntalk, bootps, chargen, finger, and all RPC daemons. NFS mounts should not be used. All HPUX systems have been converted to a trusted system in order to get C2 level security. This allows GIAC to gain security features such as shadowed /etc/passwd file, the enforcement of strong passwords, password expiration and aging, auditing, terminal access control, and time-based access control.

3. Tripwire is installed on all servers to insure file integrity.

4. All servers are backed up daily.

5. All the equipment shown in Figure 1 is locked in a secured room. Only authorized personnel are allowed into this room.

6. Redundancy is an important feature in any design. The only reason I did not include redundancy in my design was to keep it simple and manageable. A more complete design would've included duplicate equipment everywhere. Multiple servers and a load-balancing device such as Alteon or Fire Proof from Radware are a must for any design.

   For illustration purposes, I did include redundancy in the connection between GIAC and the Supplier Network. I included 2 firewalls and a Fire Proof device on each side of the firewalls. Again, a more complete design should've included two Fire Proofs on each side of the firewalls.

7. It is assumed that internal routers at GIAC will take care of routing traffic between the internal network and the DMZ.

8. The router with the IP address 10.50.10.2, which resides on the supplier's premises, will take care of NATing the supplier network addresses into ones that are routed on GIAC network. The 192.168.24.0 network will be NATed to 10.129.101.0 network.

9. Firewall administrators are able to SSH to all the Gauntlet firewalls directly on port TCP 2222. The rules for this access are controlled by the SSH demon on the firewall and not by Gauntlet rules. The reason is that Gauntlet by default controls the traffic going through the firewall but not the traffic directed to the firewall (unless local packet filters are used). Port 2222 was used so GIAC employees can use SSH through the firewalls on the standard port 22.

   CheckPoint on the other hand controls the traffic going through the firewall and the traffic directed to the firewall. It will be necessary under CheckPoint to allow traffic directed to the firewall on port 2222.

10. GIAC's security policy is in line with the VISA 10 commandments:

    a. Install and maintain a working network firewall to protect data accessible via the Internet – This is accomplished through the GIACNET and GIACEDZ firewalls.

    b. Keep security patches up-to-date.

    c. Encrypt stored data – Sensitive data are encrypted through the use of PGP 7.0.

    d. Encrypt data sent across the network – This is accomplished through the VPN device GIACVPN and through the use of PGP 7.0.

    e. Use and regularly update anti-virus software – All workstations and servers are running Norton Antivirus 5.0 with the most recent DAT file.

    f. Restrict access to data by business "need to know". This is accomplished through the use of firewalls such as GIACEXV1 and GIACEXV2.

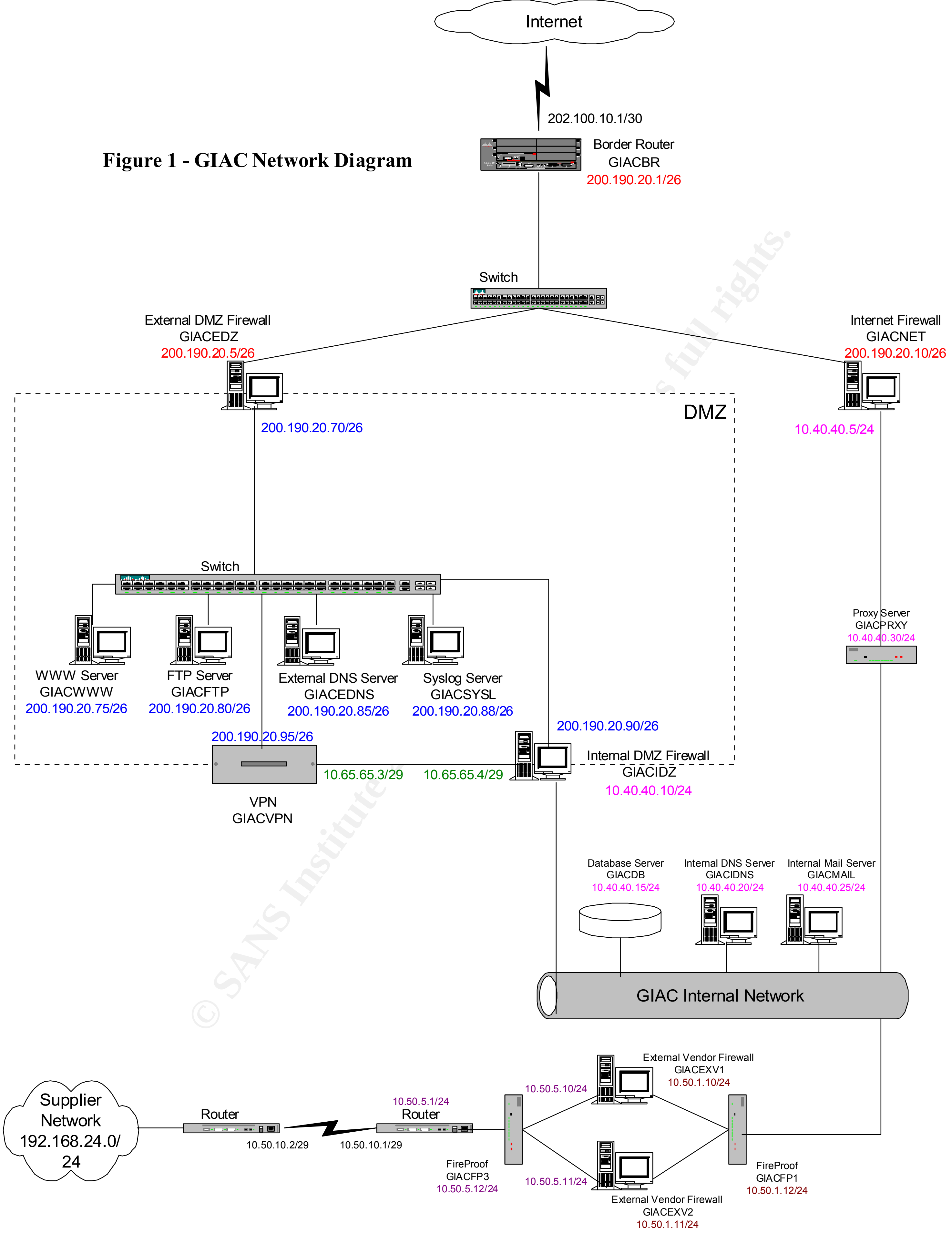
    g. Assign a unique ID to each person with computer access to data.

    h. Don't use vendor-supplied defaults for system passwords and other security parameters. This is accomplished by converting all servers to a trusted system with C2 level security, which will expire all passwords and forces users to choose new ones.

    i. Track access to data by unique ID.

    j. Regularly test security systems and processes. A good auditing policy as explained in Assignment 3 will help accomplishing this goal.

Figure 1 - GIAC Network Diagram



Assignment 2 – Security Policy

Security Policy for the Border Router (GIACBR)

Global Configurations:

The border router uses static routes only. Routing protocols are disabled. In order to enter the necessary global configurations, we need to be in global config mode. To do so enter the following commands:

Router> **enable**
Router# **config term**

Router(config)# **hostname giacbr**

giacbr(config)# **no ip source-route**

Disables loose source routing. Prevents an attacker from delivering harmful packets to destinations that cannot be reached due to ACLs.

giacbr(config)# **enable secret**

Sets the password that grants privileged administrative access to the IOS system. Enable secret should be used instead of enable password. Enable password uses a weak encryption algorithm. The enable secret command uses MD5 for password hashing, but it can be subject to dictionary attacks.

giacbr(config)# **service password-encryption**

Encrypts the passwords in the configuration files.
Algorithm used is simple Vigenere cipher, which could easily be compromised.
Not intended for protection against serious attacks.

giacbr(config)# **banner motd # Systems require authorization; Unauthorized access is illegal.#**

Informs anyone accessing the router that unauthorized access is illegal.
Needed for some jurisdictions to make it easier to prosecute hackers.

giacbr(config)# **no service tcp-small-servers**
giacbr(config)# **no service udp-small-servers**

The above 2 commands disable services that are hardly ever used.

giacbr(config)# **no service finger server**

Disables finger server so hackers cannot get any information about users logged into the router.

giacbr(config)# **no ip http server**
giacbr(config)# **no ip bootp server**

The above 2 commands disable unused services on the router.

```
giacbr(config)# no ip direct-broadcast
```

Protects against DoS attacks.

```
giacbr(config)# no ip unreachables
```

Prevents the router from sending ICMP unreachable messages, which prevents giving out network information.

```
giacbr(config)# logging 200.190.20.88
```

Sends router logging to the syslog server in the DMZ

```
giacbr(config)# no ip proxy-arp
```

Disables the router from arping on behalf of another device

```
giacbr(config)# no snmp
```

Disables SNMP

```
giacbr(config)# no cdp running
```

Prevents the router from releasing information about itself to devices directly connected to it.

```
giacbr(config)# service timestamps log datetime msecs
```

Time-stamps all log entries instead of the usual "time since restart" Cisco time-stamp.

Order is not important for the above commands.

Interface Configurations:

The router has 2 interfaces. The external serial interface facing the internet, and the internal Ethernet interface facing GIAC.

ACLs are used to secure both interfaces. The ACL implemented on the serial interface is used to control incoming traffic from the Internet. ACL implemented on the Ethernet interface is used to control traffic leaving GIAC.

Note: There is no need to use the "no ip directed-broadcast" command since it is the default in Cisco IOS 12.0 and later. GIACBR is running IOS 12.2.

Serial Interface configurations:

Interface Commands:

```
giacbr(config)# interface serial 0
giacbr(config-if)# ip address  202.100.10.1 255.255.255.252
giacbr(config-if)#  ip access-group 101 in
```

Serial Interface Inbound ACL:

```
giacbr(config)# access-list 101 permit tcp any host 200.190.20.75 eq 80
giacbr(config)# access-list 101 permit tcp any host 200.190.20.75 eq 443
```

These 2 rules allow incoming HTTP and SSL traffic to the web server

```
giacbr(config)# access-list 101 permit tcp any host 200.190.20.80 eq 20
giacbr(config)# access-list 101 permit tcp any host 200.190.20.80 eq 21
```

These 2 rules allow incoming FTP traffic to the FTP server

```
giacbr(config)# access-list 101 permit tcp any host 200.190.20.10 eq 25
```

This rule allows incoming smtp traffic to the internet firewall

```
giacbr(config)# access-list 101 permit udp any host 200.190.20.85 eq 53
```

This rule allows incoming DNS traffic to the DNS server

```
giacbr(config)# access-list 101 permit udp any host 200.190.20.95 eq 500
```

This rule allows incoming Internet Key Exchange traffic to the VPN device

```
giacbr(config)# access-list 101 permit esp any host 200.190.20.95
```

This rule allows incoming IPSec Encapsulated Security Payload traffic to the VPN device

```
giacbr(config)# access-list 101 deny ip 10.0.0.0 0.255.255.255 any log
giacbr(config)# access-list 101 deny ip 172.16.0.0 0.15.255.255 any log
giacbr(config)# access-list 101 deny ip 192.168.0.0 0.0.255.255 any log
```

These 3 rules drop and log incoming traffic from private addresses as defined in RFC1918. This is done to prevent spoofing.

```
giacbr(config)# access-list 101 deny ip 127.0.0.0 0.255.255.255 any log
```

This rule drops and logs incoming traffic from the loopback interface. This is done to prevent spoofing.

```
giacbr(config)# access-list 101 deny ip 224.0.0.0 31.255.255.255 any log
```

This rule drops and logs incoming multi-cast traffic.

```
giacbr(config)# access-list 101 deny ip host 0.0.0.0 any log
```

This rule drops and logs incoming traffic from the invalid host of 0.0.0.0

```
giacbr(config)# access-list 101 deny ip any  host 202.100.10.1 log
giacbr(config)# access-list 101 deny ip any  host 200.190.20.1 log
```

These 2 rules prevent and log any external hosts from connecting directly to the router

```
giacbr(config)# access-list 101 deny ip 200.290.20.0 0.0.0.63 log
```

This rule drops and logs incoming traffic from the Internet using GIAC's public IP addresses. This is done to prevent spoofing.

```
giacbr(config)# access-list 101 deny tcp any any range 135 139
giacbr(config)# access-list 101 deny udp any any range 135 139
```

These 2 rules drop incoming NetBIOS traffic. No need to log that since it is so common.

```
giacbr(config)# access-list 101 deny udp any any range 69 log
```

This rule drops and logs incoming TFTP traffic

```
giacbr(config)# access-list 101 deny udp any any range 514 log
```

This rule drops and logs incoming Syslog traffic

```
giacbr(config)# access-list 101 deny udp any any range 161 162 log
```

This rule drops and logs incoming SNMP traffic

```
giacbr(config)# access-list 101 deny ip any any log
```

This rule drops everything else. Even though there is an implicit deny, it is nice to have this explicit filter. This rule should be monitored carefully. Any traffic that gets dropped by this rule should be researched to find out why it was not dropped by an earlier rule.

Order of the above rules is not that important except for the deny any rule which must be the last one. We might need in the future to re-arrange some of these rules to bring the ones hit the most to the top. Also, if we have experience with spoofing in the future, we might then bring the anti-spoofing rules to the top.

Ethernet Interface configurations:

Interface Commands:

```
giacbr(config)# interface ethernet 1/0
giacbr(config-if)# ip address  200.190.20.1 255.255.255.192
giacbr(config-if)# no ip redirects
giacbr(config-if)# no ip mroute-cache
giacbr(config-if)# no keepalive
giacbr(config-if)#  ip access-group 110 in
```

Ethernet Interface Outbound ACL:

giacbr(config)# **access-list 110 permit 200.290.20.0 0.0.0.63**

This rule allows GIAC's public subnet to access anything on the Internet

giacbr(config)# **access-list 110 deny any any log**

This rule drops any other traffic that tries to leave GIAC. This is important so machines on the inside cannot participate in denial of service against other machines on the outside. This could happen as a result of internal machines being infected by warms such as Nimda or Code Red.

Order is very important here. We have to first allow needed traffic to get out and then block everything else.

Virtual Terminal Interface (VTY) configurations:

Interface Commands:

giacbr(config)# **line vty 0 4**
giacbr(config)# **login**
giacbr(config-line)# **access-class 5**
giacbr(config-line)# **transport input ssh**

Accepts only SSH connections

VTY Interface ACL:

giacbr(config)# **access-list 5 permit 200.190.20.10**

Allows only the external interface of the Internet firewall to talk to the router. This works since the firewall is proxying the connection. Anyone in the internal network that needs to SSH to the router has to go through the Internet firewall.

Make sure to save the configurations when done:

giacbr(config)# **copy running-config startup-config**

This command saves the configuration into NVRAM

giacbr(config)# **sh start**

This command lists the configuration so it can be checked

When all is done, nmap must be run against the router from the Internet and from the DMZ to make sure nothing else needs to be blocked.

Security Policy for the Internet Firewall (GIACNET)

Please refer to appendix A for instruction on how to configure Gauntlet 5.5 firewall.

The rules for this firewall are:

Rule#	Network Group	Network Group Members	Service Group	Service Group Members	Destination	Action
1	ESPMD	10.40.40.40 10.40.40.41 10.40.40.42	ESPMD	espmd	10.40.40.5	Allow
2	Authsrv	127.0.0.1	Authsrv	authsrv	10.40.40.5	Allow
3	Proxy	10.40.40.30	Web	http-gw ssl-gw	*	Allow
4	MailSrv	10.40.40.25	MailOut	smap smapd	*	Allow
5	GIAC	10.40.40.*	MailOut	smap smapd	*	Deny
6	ANY	*	MailIn	smap spamd	10.40.40.25	Allow
7	GIAC	10.40.40.*	SSH	ssh	*	Permit

Rules #1 and #2 are common to all Gauntlet 5.5 firewalls. I'm going to mention them once, but they apply to all Gauntlet 5.5 firewalls. Rule #1 allows the network group ESPMD to be able to connect to the firewall through the GUI. Only 3 workstations on the internal network are allowed to do that. Rule #2 is for authentication to the firewall.

Rule number #3 allows the proxy server to use the http and ssl gateways to go anywhere. Remember that all internal workstations' browsers are configured to proxy off the proxy server, which is the only host, allowed to use the Web service group. The anti-virus feature provided in the proxy is enabled.

Rule #4 allows the mail server to use the MailOut service group which contains smap and smapd services. The mail server is allowed to go anywhere. Smap service listens on port TCP 25 for mail. It receives the mail and after checking it for basic rules, it puts it in a directory called /var/spool/smap. Smapd then picks it up and puts it in a directory called /var/spool/mqueue. Smapd then spawns Sendmail to deliver the mail to its destination. The anti-virus feature provided in the proxy is enabled.

Rule #5 denies any other host on the GIAC network from relaying mail off the firewall. This insures that only the mail server is allowed to relay mail off the firewall.

Rule #6 is for incoming mail. It allows any host in the world to deliver mail to the firewall. The firewall then relays it to the mail server on the internal network using the same process as explained in Rule #4.

Rule #7 allows anyone on the GIAC internal network to be able to SSH through the firewall to any host on the internet. The service SSH is a TCP plug listening on port TCP 22.

Order of the above rules is not important except for rules 4,5, and 6. They have to be in this order to allow only the mail server to relay off the firewall, deny all other hosts on the internal GIAC network, and to allow any host on the Internet to deliver mail to the firewall.

We could test the above by logging to a workstation on the GIAC internal network and doing the following:

1. Surf the Internet. Make sure the browser is set to proxy off the proxy server. This should work.
2. Change the browser to proxy off the firewall directly. This should fail.
3. Send an email to someone on the Internet and make sure they receive it. This should work.
4. Receive an email from the administrator account at yahoo.com. This should work.
5. SSH to the administrator box at home. This should work.

Monitor Gauntlet logs by "tail –f /var/log/messages" during these tests. And finally, we should run nmap against the internal and external interfaces of the firewall to make sure no other services are allowed.

Security Policy for the External DMZ Firewall (GIACEDZ)

The rules for this firewall are:

Rule#	Network Group	Network Group Members	Service Group	Service Group Members	Destination	Action
1	ANY	*	Web	http ssl-gw	200.190.20.75	Allow
2	ANY	*	FTP	ftp-gw	200.190.20.80	Allow

Rule #1 allows any host on the Internet to reach only GIAC's web server through ports 80 and 443. Note here that we used a plug called http listening on port 80 instead of using the HTTP proxy. This was done for a couple of reasons:

1. To avoid security risks inherent into the HTTP protocol such as port re-direction. Since the HTTP proxy understands the HTTP protocol, a hacker can exploit some features in the HTTP protocol that could be harmful. A plug on the other hand will be able to handle traffic on port 80, but it does not fully understand the HTTP protocol.

2. Network Associates specifically states that they do not support the use of the HTTP proxy for inbound traffic.

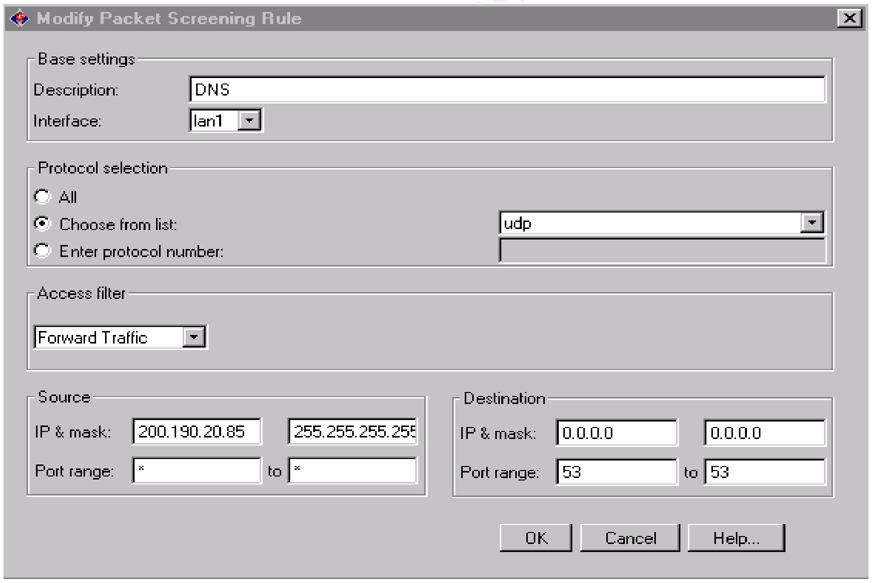
Rule #2 allows any host on the Internet to reach only GIAC's FTP server. The ftp-gw proxy will handle the connection. The anti-virus feature provided in the proxy is enabled.

The rest of the rules are handled through packet filters since they pass UDP traffic. I'm going to explain these through screen shots.

Note: Gauntlet 6.0 provides a UDP proxy that could be used instead of packet filters. This is a great security improvement feature.

Screen 1 shows a packet filter for handling DNS traffic. It allows GIAC's DNS server to talk to any server on the Internet on port UDP 53. Since the connection initiates from the trusted side of the firewall, Interface is set to lan1, which is the internal interface of the firewall.

Screen 1



Modify Packet Screening Rule

Base settings
Description: DNS
Interface: lan1

Protocol selection
○ All
● Choose from list: udp
○ Enter protocol number:

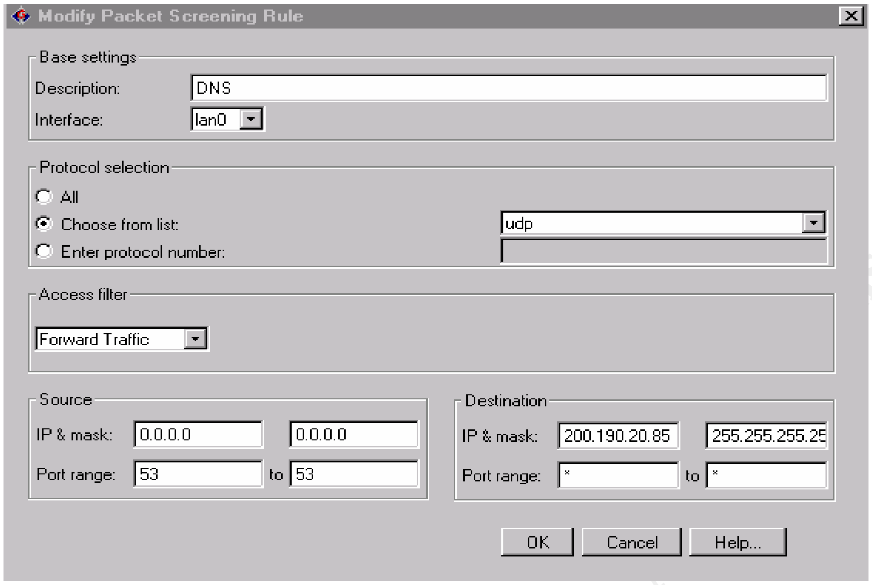
Access filter
Forward Traffic

Source
IP & mask: 200.190.20.85 255.255.255.25
Port range: * to *

Destination
IP & mask: 0.0.0.0 0.0.0.0
Port range: 53 to 53

Screen 2 shows a packet filter for handling the DNS reply back to GIAC's DNS server. Interface is set to lan0, which is the external interface of the firewall.

Screen 2



Modify Packet Screening Rule

Base settings

Description: DNS

Interface: lan0

Protocol selection

○ All

● Choose from list: udp

○ Enter protocol number:

Access filter

Forward Traffic

Source

IP & mask: 0.0.0.0 0.0.0.0

Port range: 53 to 53

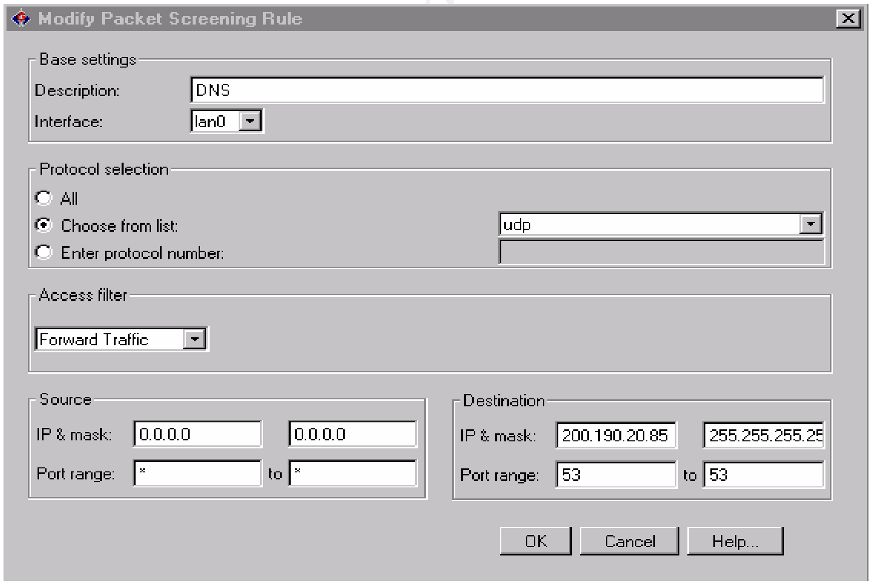
Destination

IP & mask: 200.190.20.85 255.255.255.25

Port range: * to *

Screen 3 shows a packet filter for handling DNS traffic. It allows any host on the Internet to query GIAC's DNS server on port UDP 53. This is needed for other DNS servers to obtain information about the GIAC domain. Since the connection initiates from the untrusted side of the firewall, Interface is set to lan0.

Screen 3



Modify Packet Screening Rule

Base settings

Description: DNS

Interface: lan0

Protocol selection

○ All

● Choose from list: udp

○ Enter protocol number:

Access filter

Forward Traffic

Source

IP & mask: 0.0.0.0 0.0.0.0

Port range: * to *

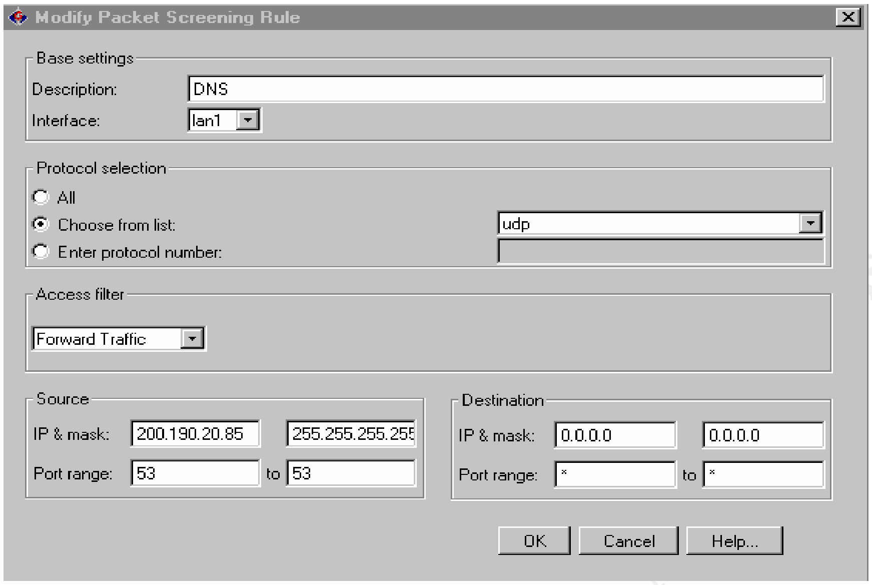
Destination

IP & mask: 200.190.20.85 255.255.255.25

Port range: 53 to 53

Screen 4 shows a packet filter for handling the DNS reply from GIAC's DNS server to the servers that made the query. Interface is set to lan1.

Screen 4



Modify Packet Screening Rule

Base settings
Description: DNS
Interface: lan1

Protocol selection
○ All
● Choose from list: udp
○ Enter protocol number:

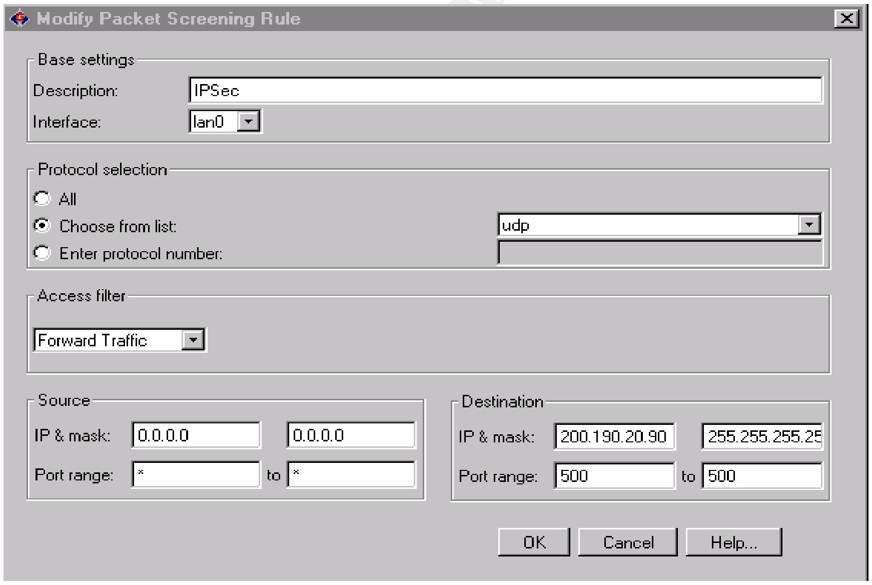
Access filter
Forward Traffic

Source
IP & mask: 200.190.20.85 255.255.255.255
Port range: 53 to 53

Destination
IP & mask: 0.0.0.0 0.0.0.0
Port range: * to *

Screen 5 shows a packet filter for handling IPSec traffic. It allows any host on the Internet to reach the VPN device over port UDP 500 (IKE). Since the connection initiates from the untrusted side of the firewall, Interface is set to lan0.

Screen 5



Modify Packet Screening Rule

Base settings
Description: IPSec
Interface: lan0

Protocol selection
○ All
● Choose from list: udp
○ Enter protocol number:

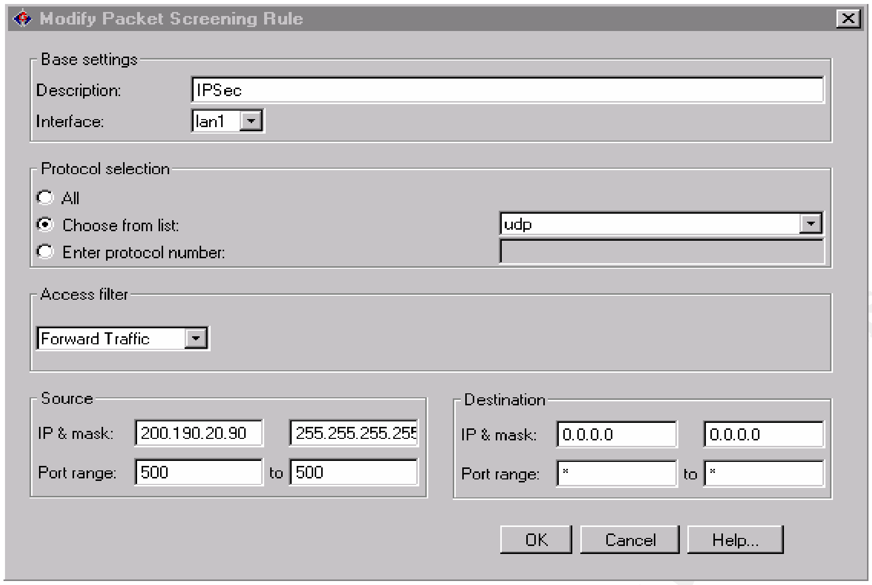
Access filter
Forward Traffic

Source
IP & mask: 0.0.0.0 0.0.0.0
Port range: * to *

Destination
IP & mask: 200.190.20.90 255.255.255.25
Port range: 500 to 500

Screen 6 shows a packet filter for handling the IKE reply. Interface is set to lan1.

Screen 6



Modify Packet Screening Rule

Base settings

Description: IPSec

Interface: lan1

Protocol selection

○ All

● Choose from list: udp

○ Enter protocol number:

Access filter

Forward Traffic

Source

IP & mask: 200.190.20.90 255.255.255.255

Port range: 500 to 500

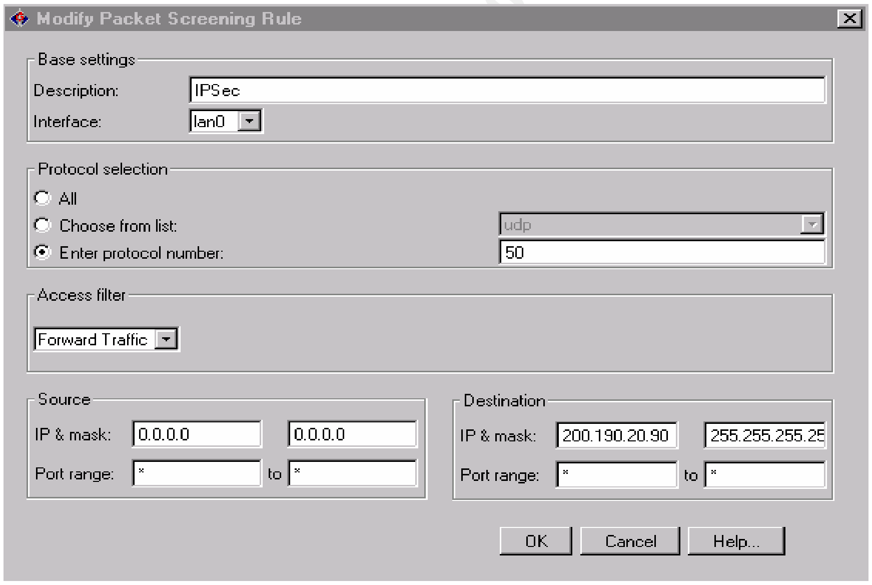
Destination

IP & mask: 0.0.0.0 0.0.0.0

Port range: * to *

Screen 7 shows a packet filter for handling IPSec traffic. It allows any host on the Internet to reach the VPN device over protocol 50 (ESP). Since the connection initiates from the untrusted side of the firewall, Interface is set to lan0.

Screen 7



Modify Packet Screening Rule

Base settings

Description: IPSec

Interface: lan0

Protocol selection

○ All

○ Choose from list: udp

● Enter protocol number: 50

Access filter

Forward Traffic

Source

IP & mask: 0.0.0.0 0.0.0.0

Port range: * to *

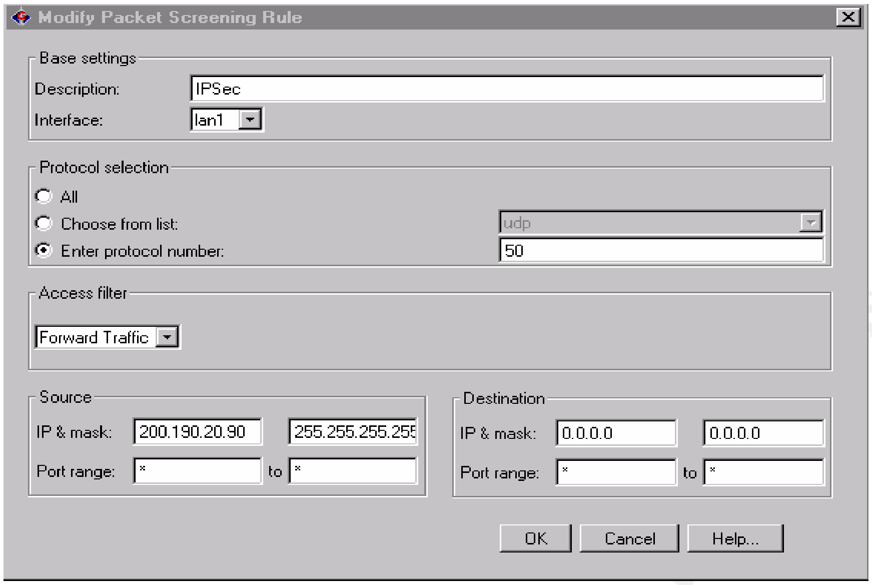
Destination

IP & mask: 200.190.20.90 255.255.255.255

Port range: * to *

Screen 8 shows a packet filter for handling the ESP reply. Interface is set to lan1.

Screen 8



Base settings

Description: IPSec

Interface: lan1

Protocol selection

○ All

○ Choose from list: udp

● Enter protocol number: 50

Access filter

Forward Traffic

Source

IP & mask: 200.190.20.90 255.255.255.25

Port range: * to *

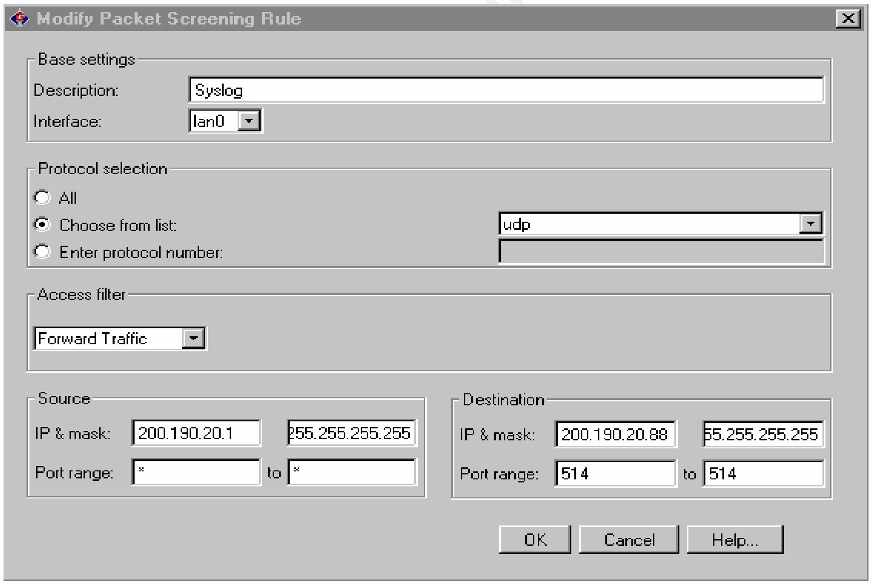
Destination

IP & mask: 0.0.0.0 0.0.0.0

Port range: * to *

Screen 9 shows a packet filter for handling Syslog traffic. It allows GIAC's border router to reach the Syslog server over port UDP 514. Since the connection initiates from the untrusted side of the firewall, Interface is set to lan0.

Screen 9



Base settings

Description: Syslog

Interface: lan0

Protocol selection

○ All

● Choose from list: udp

○ Enter protocol number:

Access filter

Forward Traffic

Source

IP & mask: 200.190.20.1 255.255.255.255

Port range: * to *

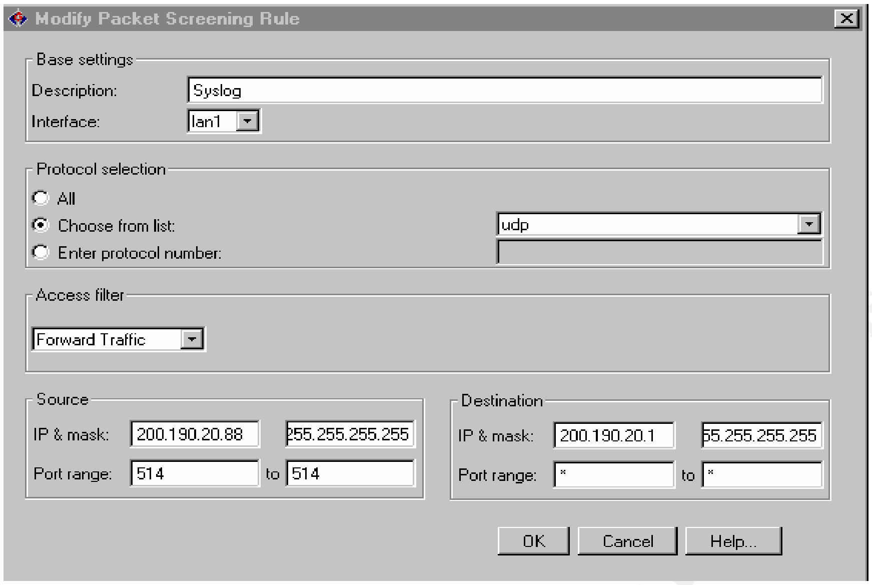
Destination

IP & mask: 200.190.20.88 55.255.255.255

Port range: 514 to 514

Screen 10 shows a packet filter for handling the Syslog reply. Interface is set to lan1.

Screen 10



Base settings

Description: Syslog

Interface: lan1

Protocol selection

○ All

● Choose from list: udp

○ Enter protocol number:

Access filter

Forward Traffic

Source

IP & mask: 200.190.20.88 255.255.255.255

Port range: 514 to 514

Destination

IP & mask: 200.190.20.1 55.255.255.255

Port range: * to *

Order of the above rules is not important. Remember that Gauntlet denies anything that is not explicitly permitted. So there is always an implicit deny rule at the end.

We could test the above by logging to a workstation outside the GIACEDZ firewall and inside the GIACBR router and doing the following:

1. Browse the GIAC web site. This should work.
2. FTP to the FTP server. This should work.
3. Establish a connection with the VPN device. This should work.

Monitor Gauntlet logs by "tail –f /var/log/messages" during these tests. Gauntlet does not log by default traffic from packet filters. It is important to turn IPFS on by issuing "ipfs –t on" command at the firewall. Gauntlet then logs packet filter traffic in /var/log/messages. IPFS generates a lot of traffic so don't forget to turn it off when done (ipfs –t off). And finally, we should run nmap against the internal and external interfaces of the firewall to make sure no other services are allowed.

Security Policy for the Internal DMZ Firewall (GIACIDZ)

The rules for this firewall are:

No.	Source	Destination	Service	Action
1	10.40.40.40 10.40.40.41 10.40.40.42	10.65.65.3	http	Accept
2	10.40.40.0/24	200.190.20.75	http https	Accept
3	10.40.40.0/24	200.190.20.80	ftp	Session Auth
4	10.40.40.40 10.40.40.41 10.40.40.42	200.190.20.0/26	ssh	Session Auth
5	10.40.40.40 10.40.40.41 10.40.40.42	10.40.40.10	ssh-2222	Session Auth
6	10.40.40.20	200.190.20.85	dns-udp	Accept
7	200.190.20.75	10.40.40.15	oracle	Accept
8	10.50.50.0/24	10.40.40.0/24	Any	Session Auth
9	Any	Any	Any	Drop

Rule#1 allows three of the administrators' workstations the ability to connect to the VPN device over port 80. This is needed for managing the VPN device.

Rule #2 allows any host on the Internal GIAC network the ability to connect to the web server in the DMZ. The services allowed are http and https.

Rule #3 allows any host on the Internal GIAC network the ability to connect to the FTP server in the DMZ. Authentication is required for this connection.

Rule #4 allows three of the administrators' workstations the ability to connect to any server in the DMZ by using SSH on the standard port 22. This is needed for maintenance and support. Authentication is required for this connection.

Rule #5 allows three of the administrators' workstations the ability to SSH to the firewall itself. SSH daemon on the firewall was configured to listen on port 2222.

Rule #6 allows the internal DNS server the ability to connect to the DNS server in the DMZ over port UDP 53. This is needed to resolve names outside GIAC's domain.
Rule #7 allows the web server in the DMZ the ability to connect to the database server over port TCP 1521. This is needed so the web server can retrieve and update information in the Oracle database.

Rule #8 allows any IP address assigned by the VPN device the ability to connect to any host on the Internal GIAC network over any protocol. When the connection comes from the Internet to the VPN device, it gets decrypted and an IP from the pool 10.50.50.0/24 gets assigned as the source address. This is done to allow telecommuters and administrators the ability to connect to servers on the internal network. Anyone connecting has to authenticate at the VPN and at the firewall. Also, Nortel provides a firewall module that could be used to decide who can go where on the internal network.

Rule #9 drops anything else.

Order of the above rules is not important except for rule #9 which must be the last one.

We could test the above by logging to a workstation on the GIAC internal network and doing the following:

1. Connect to the Web Server from a browser. This should work.
2. FTP to the FTP server. This should work.
3. From one of the three administrator workstations, SSH to the web server and then to the FTP server. This should work.
4. From a workstation outside the GIACEDZ firewall, establish a connection to the VPN and then connect to the database server. This should work.

Monitor CheckPoint logs during these tests. And finally, we should run nmap against the internal and external interfaces of the firewall to make sure no other services are allowed.

Security Policy for the External Vendor Firewall (GIACEXV1)

The rules for this firewall are:

Rule#	Network Group	Network Group Members	Service Group	Service Group Members	Destination	Action
1	Supplier	10.129.101.10 10.129.101.11	Supplier	Oracle	10.40.40.15	Allow

There is only one rule of interest in the GIACEXV1 firewall. It permits 2 hosts from the Supplier network to connect to GIAC's database server. The connection is accomplished through a TCP plug (called oracle) listening on port TCP 1521. This is the port required for communication with Oracle database.

We should run nmap against the internal and external interfaces of the firewall to make sure no other services are allowed.

Security Policy for the VPN (GIACVPN)

This is the Nortel Contivity 4600 VPN device. It implements IPSec in a tunnel mode for gateway-to-gateway connections with business partners. The values used to setup IPSec are:

1. Triple DES with SHA1 and MD5 Integrity for ESP
2. 56-bit DES with Group 1 and Triple DES with Group 2 for IKE encryption and Diffie-Hellman group
3. LDAP for authentication

The Nortel VPN comes with a few built-in filters that are not needed. A deny all filter is placed on the VPN's interface facing the DMZ. IKE and ESP are allowed by default. A permit all filter is placed on the interface facing the firewall GIACIDZ.

The primary use of the VPN is to allow telecommuters, administrators, and partners the ability to access the Internal GIAC network. Only ESP protocol is utilized. AH protocol is not used.

Nortel provides a statefull firewall module that is used to provide granular security policies. Three groups are created, one for the telecommuters, another for the administrators, and another for the partners. Each group will have its own policy rules controlling what each group can access on the Internal GIAC network.

To test the configurations, create a test account in each group. Login with the test account from each group and make sure you can only get to the servers allowed by each group. We should also run nmap against the internal and external interfaces of the VPN to make sure no other services are allowed.

Assignment 3 – Auditing the Security Architecture

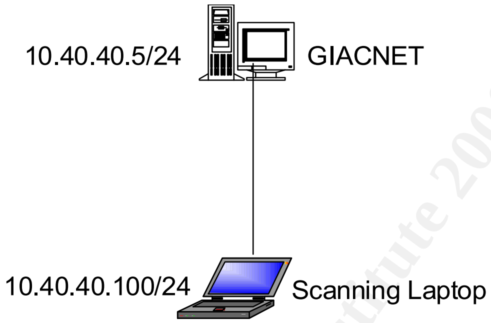
Plan the Audit

It is important to conduct an audit on regular basis to verify systems integrity. For GIAC, we are going to audit some of the firewalls and the VPN device to ensure that security policies are implemented as intended.

We will scan one of the Gauntlet firewalls (GIACNET), the CheckPoint firewall (GIACIDZ), and the VPN device (GIACVPN) using nmap version 2.54BETA25 and 2.54BETA30 installed on two Red Hat Linux laptops. The audit will require the time of a full time employee for about a week. The cost of the audit is the employee's salary for a week.

The time when the scans are done is very critical. We will do it on third shift to minimize the number of affected people in case of problems caused by the scan. We will also make sure the appropriate people are notified of the time and duration of the scan. Systems and firewalls administrators should be notified.

GIACNET Scan



Note: Scanning the internal and external interfaces produced the same exact results. Showing the results from the internal interface only.

TCP Scan Results:

```
# nmap (V. 2.54BETA30) scan initiated Tue Nov 27 12:16:33 2001 as: nmap -
sS -O -v -P0 -p 1-65535 -oN /root/giacnet_internal_TCP 10.40.40.5
Interesting ports on giacnet.giac.com (10.40.40.5):
(The 65525 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp
80/tcp     open        http
113/tcp    open        auth
443/tcp    open        https
2222/tcp   open        unknown
```

```
8004/tcp   open        unknown
34604/tcp  filtered    unknown

No exact OS matches for host (If you know what OS is running on it, see
http://www.insecure.org/cgi-bin/nmap-submit.cgi).
TCP/IP fingerprint:
SInfo(V=2.54BETA30%P=i686-pc-linux-gnu%D=11/27%Time=3C03D8AF%O=23%C=1)
TSeq(Class=RI%gcd=1%SI=41EF1%TS=U)
TSeq(Class=RI%gcd=1%SI=3D225%TS=U)
TSeq(Class=RI%gcd=1%SI=3BA7E%TS=U)
T1(Resp=Y%DF=Y%W=8000%ACK=S++%Flags=AS%Ops=M)
T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=Y%W=8000%ACK=O%Flags=A%Ops=)
T4(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
PU(Resp=N)

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=244350 (Good luck!)
IPID Sequence Generation: Busy server or unknown class

# Nmap run completed at Tue Nov 27 12:17:19 2001 -- 1 IP address (1 host
up) scanned in 45 seconds
```

UDP Scan Results:

```
# nmap (V. 2.54BETA30) scan initiated Mon Nov 26 15:26:14 2001 as: nmap -sU
-O -v -P0 -p 1-65535 -oN /root/giacnet_internal_UDP 10.40.40.5
Warning:  OS detection will be MUCH less reliable because we did not find
at least 1 open and 1 closed TCP port
Interesting ports on giacnet.giac.com (10.40.40.5):
(The 65531 ports scanned but not shown below are in state: closed)
Port       State       Service
123/udp    open        ntp
34604/udp  open        unknown

Too many fingerprints match this host for me to give an accurate OS guess
TCP/IP fingerprint:
SInfo(V=2.54BETA30%P=i686-pc-linux-gnu%D=11/26%Time=3C02B38D%O=-1%C=-1)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%RIPCK=0%UCK=E%ULEN=134%DAT=E)

# Nmap run completed at Mon Nov 26 15:26:37 2001 -- 1 IP address (1 host
up) scanned in 22 seconds
```

Scan Analysis:

1. The scan did not reveal any surprises.
2. Nmap could not detect the OS.
3. Remember that services open by Gauntlet are "listening services" and will show up on the scan (such as HTTP proxy). This is in addition to any daemons listening on the machine itself (such as SSH).
4. The firewall sent resets for closed TCP ports and unreachable for closed UDP ports.
5. Here is an explanation of the reported ports:

   - Port 22 is open through a Gauntlet plug for allowing SSH traffic through the firewall.
   - Port 25 is for handling mail from and to GIAC.
   - Ports 80 and 443 are for allowing HTTP and SSL traffic from GIAC to the Internet.
   - Port 113 is for Gauntlet authentication.
   - Port 2222 is open through the SSH daemon running on the firewall. It is used for allowing administrators to SSH to the firewall.
   - Port 8004 is for allowing the GUI client to connect to the firewall.
   - Port 34604 TCP is for HP AutoRAID management. This port is filtered.
   - Port 123 UDP is xntpd. ntp.conf has the rules for specifying the servers the firewall can sync from, and for specifying who can sync from the firewall.
   - Port 34604 UDP is for HP AutoRAID management.

Gauntlet detected the scan. Here is a sample from the logs. Note how nmap randomizes the source port number:

```
Nov 27 12:16:34 giacnet vmunix: securityalert: tcp if=lan1 from
10.40.40.100:43716 to 10.40.40.5 on unserved port 48020
Nov 27 12:16:34 giacnet vmunix: securityalert: tcp if=lan1 from
10.40.40.100:43716 to 10.40.40.5 on unserved port 27657
Nov 27 12:16:34 giacnet vmunix: securityalert: tcp if=lan1 from
10.40.40.100:43716 to 10.40.40.5 on unserved port 47028

Nov 26 15:26:15 giacnet vmunix: securityalert: udp if=lan1 from
10.40.40.100:63561 to 10.40.40.5 on unserved port 42655
Nov 26 15:26:15 giacnet vmunix: securityalert: udp if=lan1 from
10.40.40.100:63561 to 10.40.40.5 on unserved port 26029
Nov 26 15:26:15 giacnet vmunix: securityalert: udp if=lan1 from
10.40.40.100:63561 to 10.40.40.5 on unserved port 15811
```

During the scan, tcpdump ran on the scanning laptop which confirmed the findings of nmap:

Open TCP Port (standard handshake):

```
12:16:34.905303 eth0 > laptop.giac.com.45825 > giacnet.giac.com.80: S
1837860319:1837860319(0) win 3072

12:16:34.905303 eth0 < giacnet.giac.com.80 > laptop.giac.com.45825: S
3773813941:3773813941(0) ack 1837860320 win 32768 <mss 536> (DF)

12:16:34.905303 eth0 > laptop.giac.com.45825 > giacnet.giac.com.80: R
1837860320:1837860320(0) win 0 (DF)
```

Filtered TCP Port (no response to the SYN):

```
12:16:45.225303 eth0 > laptop.giac.com.50665 > giacnet.giac.com.34604:
S 4138886049:4138886049(0) win 4096
12:16:51.245303 eth0 > laptop.giac.com.50666 > giacnet.giac.com.34604:
S 593002241:593002241(0) win 4096
12:16:57.265303 eth0 > laptop.giac.com.50667 > giacnet.giac.com.34604:
S 2844053899:2844053899(0) win 4096
12:17:03.285303 eth0 > laptop.giac.com.50668 > giacnet.giac.com.34604:
S 4138886049:4138886049(0) win 4096
12:17:09.305303 eth0 > laptop.giac.com.50669 > giacnet.giac.com.34604:
S 593002241:593002241(0) win 4096
12:17:09.325303 eth0 > laptop.giac.com.50670 > giacnet.giac.com.34604:
S 2844053899:2844053899(0) win 4096
```

Closed TCP Port (Host sends a reset immediately after receiving a SYN):

```
12:53:19.715303 eth0 > laptop.giac.com.40973 > giacnet.giac.com.ntp: S
3991040088:3991040088(0) win 3072
12:53:19.715303 eth0 < giacnet.giac.com.ntp > laptop.giac.com.40973: R
0:11(11) ack 3991040089 win 0 (DF)
```

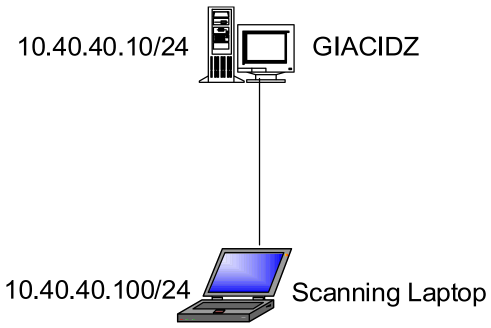
Open UDP Port (no response back):

```
15:26:37.565303 eth0 > laptop.giac.com.55161 > giacnet.giac.com.123:
udp 0
15:26:43.585303 eth0 > laptop.giac.com.55162 > giacnet.giac.com.123:
udp 0
```

Closed UDP Port (host sends back unreachable):

```
15:26:55.655303 eth0 > laptop.giac.com.63809 > giacnet.giac.com.2:udp 0
15:26:55.655303 eth0 < giacnet.giac.com > laptop.giac.com: icmp:
giacnet.giac.com udp port 2 unreachable
```

GIACIDZ Scan



Note: Scanning the internal and external interfaces produced the same exact results. Showing the results from the internal interface only.

TCP Scan Results:

```
# nmap (V. 2.54BETA30) scan initiated Tue Nov 27 12:20:37 2001 as: nmap -
sS -O -v -P0 -p 1-65535 -oN /root/giacidz_internal_TCP 10.40.40.10
Interesting ports on giacidz.giac.com (10.40.40.10):
(The 65530 ports scanned but not shown below are in state: filtered)
Port       State       Service
264/tcp    open        bgmp
265/tcp    open        unknown
2222/tcp    open        passgo

Remote operating system guess: Apple MacOS 9.04 (Powermac or G4)
Uptime 196.238 days (since Tue May 15 09:26:01 2001)

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=55429 (Worthy challenge)
IPID Sequence Generation: Incremental

# Nmap run completed at Tue Nov 27 14:09:27 2001 -- 1 IP address (1 host
up) scanned in 6529 seconds
```

UDP Scan Results:

```
# nmap (V. 2.54BETA30) scan initiated Tue Nov 27 12:21:06 2001 as: nmap -
sU -O -v -P0 -p 1-65535 -oN /root/giacidz_internal_UDP 10.40.40.10
Warning:  OS detection will be MUCH less reliable because we did not find
at least 1 open and 1 closed TCP port
Interesting ports on giacidz.giac.com (10.40.40.10):
Port       State       Service
1/udp      open        tcpmux
2/udp      open        compressnet
3/udp      open        compressnet
.
.
.
65533/udp  open        unknown
65534/udp  open        unknown
65535/udp  open        unknown

Too many fingerprints match this host for me to give an accurate OS guess
TCP/IP fingerprint:
SInfo(V=2.54BETA30%P=i686-pc-linux-gnu%D=11/27%Time=3C042A06%O=-1%C=-1)
T5(Resp=N)
T6(Resp=N)
T7(Resp=N)
PU(Resp=N)

# Nmap run completed at Tue Nov 27 18:04:22 2001 -- 1 IP address (1 host
up) scanned in 20595 seconds
```

Scan Analysis:

1. The scan did not reveal any surprises.
2. Nmap detected the wrong OS. The machine is running HPUX 11.0 and not Apple MacOS 9.04.
3. Remember that services open by CheckPoint are not "listening services" and will not show up on the scan (such as port 80). Only daemons listening on the machine itself will show up on the scan (such as SSH).
4. The firewall did not respond for closed TCP ports so nmap reported them as filtered. Also, there was no response on any of the UDP ports so nmap reported them as open.
5. Here is an explanation of the reported ports:

    - Ports 264 and 265 are reported as open. Here is a direct quote from www.phoneboy.com to explain these two ports:

      "TCP Port 264 is used for Secure Client (SecuRemote) build 4100 and later to fetch network topology and encryption keys from a FireWall-1 Management Console

TCP port 265, according to my 4.1SP1 objects.C, is labeled "Check Point VPN-1 Public Key Transfer Protocol." I'm guessing this is used by FireWall-1 to exchange public keys with other hosts. "

- Port 2222 is used for accessing the firewall through SSH.

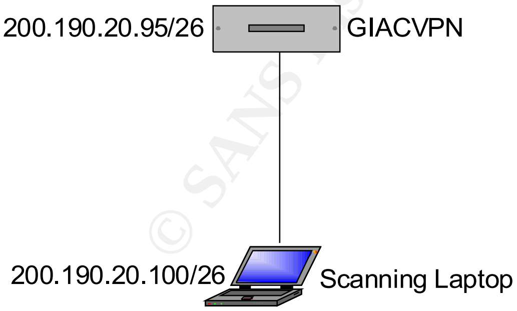
CheckPoint detected the scan. Here is a sample from the logs. Note how nmap randomizes the source port number:

```
Nov 27 12:30:00 giacidz syslog: 12:29:59 drop   giacidz.giac.com >lan0
proto tcp src 10.40.40.100 dst 10.40.40.10 service 13081 s_port 45677
len 40 rule 37
Nov 27 12:30:00 giacidz syslog: 12:29:59 drop   giacidz.giac.com >lan0
proto tcp src 10.40.40.100 dst 10.40.40.10 service 49130 s_port 45677
len 40 rule 37
Nov 27 12:30:00 giacidz syslog: 12:29:59 drop   giacidz.giac.com >lan0
proto tcp src 10.40.40.100 dst 10.40.40.10 service 57162 s_port 45677
len 40 rule 37

Nov 27 12:30:04 giacidz syslog: 12:30:03 drop   giacidz.giac.com >lan0
proto udp src 10.40.40.100 dst 10.40.40.10 service 55630 s_port 35019
len 28 rule 37
Nov 27 12:30:04 giacidz syslog: 12:30:03 drop   giacidz.giac.com >lan0
proto udp src 10.40.40.100 dst 10.40.40.10 service 20425 s_port 35019
len 28 rule 37
Nov 27 12:30:04 giacidz syslog: 12:30:03 drop   giacidz.giac.com >lan0
proto udp src 10.40.40.100 dst 10.40.40.10 service 60734 s_port 35019
len 28 rule 37
```

tcpdump confirmed the findings of nmap. I'm not going to list the output of tcpdump since they are similar to the ones listed above for GIACNET.

GIACVPN Scan



Note: Scanning the internal and external interfaces produced the same exact results. Showing the results from the external interface only.

TCP Scan Results:

```
# nmap (V. 2.54BETA25) scan initiated Mon Nov 26 18:24:19 2001 as: nmap -
sS -O -v -P0 -p 1-65535 -oN /root/giacvpn_external_TCP 200.190.20.95
Warning:  OS detection will be MUCH less reliable because we did not find
at least 1 open and 1 closed TCP port
All 65535 scanned ports on giacvpn.giac.com (200.190.20.95) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess
TCP/IP fingerprint:
SInfo(V=2.54BETA25%P=i686-pc-linux-gnu%D=11/27%Time=3C035CFB%O=-1%C=-1)
T5(Resp=N)
T6(Resp=N)
T7(Resp=N)
PU(Resp=N)

# Nmap run completed at Tue Nov 27 03:29:31 2001 -- 1 IP address (1 host
up) scanned in 32711 seconds
```

UDP Scan Results:

```
# nmap (V. 2.54BETA30) scan initiated Tue Nov 27 12:19:37 2001 as: nmap -
sU -O -v -P0 -p 1-65535 -oN /root/giacvpn_external_UDP 200.190.20.95
Warning:  OS detection will be MUCH less reliable because we did not find
at least 1 open and 1 closed TCP port
All 65535 scanned ports on giacvpn.giac.com (200.190.20.95) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess
TCP/IP fingerprint:
SInfo(V=2.54BETA30%P=i686-pc-linux-gnu%D=11/28%Time=3C050DE3%O=-1%C=-1)
T5(Resp=N)
T6(Resp=N)
T7(Resp=N)
PU(Resp=N)

# Nmap run completed at Wed Nov 28 10:16:35 2001 -- 1 IP address (1 host
up) scanned in 79016 seconds
```

Scan Analysis:

1. This scan was the trickiest and took the longest. The TCP scan took about 9 hours and the UDP scan took about 22 hours.
2. Nmap could not detect the OS.
3. The VPN did not respond for closed TCP ports so nmap reported them as filtered. Also, there was no response on any of the UDP ports so nmap also reported them as filtered.
4. No matter how many TCP ports nmap scanned, it always reported them as filtered. UDP ports were a different story. I noticed that when nmap scanned 25 ports or less, it reported them as open. But when it scanned more than 25 ports, it reported them as filtered. Tcpdump showed the same results in either case. This must be a result of the Anti-DoS feature in the VPN device. When I scanned exactly 25 ports, the output from nmap looked like this:

```
[root@localhost]# nmap -sU -P0 -p 333-357 200.190.20.95

Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
Interesting ports on giacvpn.giac.com (200.190.20.95):

Port       State       Service
333/udp    open        unknown
334/udp    open        unknown
335/udp    open        unknown
336/udp    open        unknown
337/udp    open        unknown
338/udp    open        unknown
339/udp    open        unknown
340/udp    open        unknown
341/udp    open        unknown
342/udp    open        unknown
343/udp    open        unknown
344/udp    open        pdap
345/udp    open        pawserv
346/udp    open        zserv
347/udp    open        fatserv
348/udp    open        csi-sgwp
349/udp    open        mftp
350/udp    open        matip-type-a
351/udp    open        matip-type-b
352/udp    open        dtag-ste-sb
353/udp    open        ndsauth
354/udp    open        bh611
355/udp    open        datex-asn
356/udp    open        cloanto-net-1
357/udp    open        bhevent

Nmap run completed -- 1 IP address (1 host up) scanned in 36
seconds
```

But when I scanned 26 ports, the output from nmap looked like this:

```
[root@localhost]# nmap -sU -P0 -p 333-358 200.190.20.95

Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ ) All 26
scanned ports on Giacvpn.giac.com (200.190.20.95) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 36
seconds
```

Audit Analysis and Recommendations:

1. Security is pretty good on the GIAC infrastructure. There are no unneeded services available on the firewalls or the VPN device.

2. Implementing an Intrusion Detection System (IDS) is needed to detect any unauthorized activities. Century Taps from Finisar Systems could be used. They send a copy of all the traffic to an ISS RealSecure collector station on the internal network. The collector will analyze the traffic for known signatures. Please see appendix C for the GIAC network diagram with IDS taps placed on it.

3. GIAC's network design needs to be improved to include more redundancy. The redundancy implemented for the Supplier network needs to be implemented everywhere else. Appendix B can be used to help in doing that.

4. The VPN is being managed through http. That is not good because it is not encrypted. A better way is to manage it through an IPSEC tunnel.

5. A firewall needs to be implemented to segment out the database server from the rest of the GIAC network to protect it against internal threats.

6. The VPN device allows access to anything on the GIAC internal network. The VPN firewall module should be used to limit access (by user) to needed hosts and services only.

7. A process must be implemented to monitor who has access to the firewalls. The following shell script can be scheduled in cron on each firewall to monitor /etc/passwd file for unauthorized user IDs. The file good_ids contains a list of the IDs allowed to be in the /etc/passwd file. Only root should be able to modify the good_ids file.

```
#!/bin/sh
#
#  This script simply reads the passwd file and compares the IDs to a list in a
#  second file.  If the ID exists in the second file, nothing is to be done.  Otherwise
#  the ID will be reported.
#
passwd=/etc/passwd
input_file=/etc/good_ids
server=$(hostname)
subject="Security Report for: $server"
email_list=administrator@giac.com
output_file=/tmp/illegal_ids
flag=0
#
echo "The following IDs are illegal on: $server" > $output_file
echo "" >> $output_file
```

```
while read passwd_line
do
        id=$(echo $passwd_line |cut -f1 -d: )
        if ! grep -qe $id $input_file
        then
              flag=1
              echo $passwd_line >> $output_file
        fi
done < $passwd
if [ $flag -eq 1 ]
then
        cat $output_file | mailx -s "$subject" $email_list
fi
exit 0
```

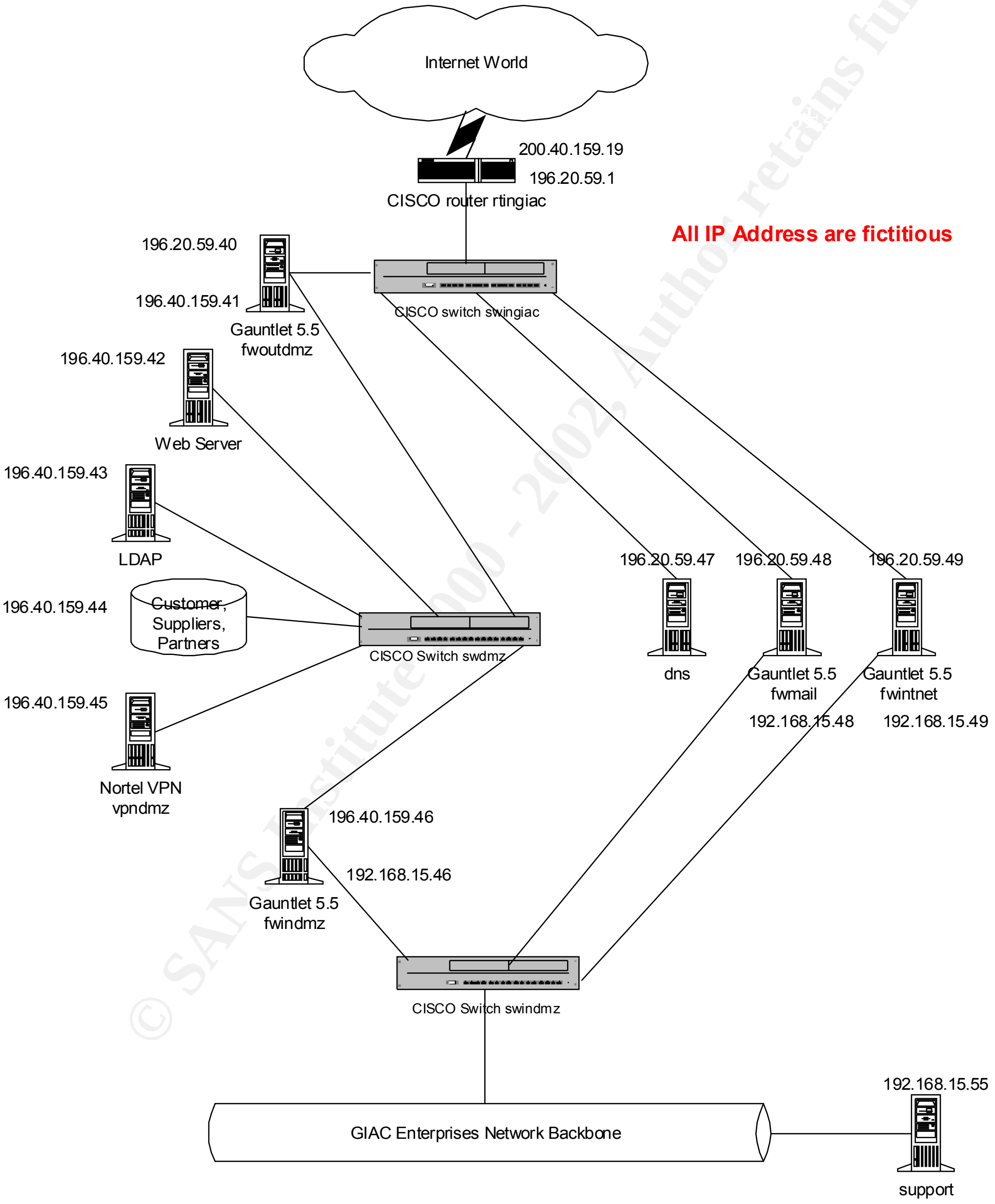
8. A procedure needs to be put in place to accomplish the following:

    a. Review all audit logs. Audit trails and logs need to be protected from alteration. This can be achieved by having all firewalls and servers log to a secured centralized logging server.
    b. Report on all security changes.
    c. Do regular and documented self-audits.
    d. Test all backups on regular basis.
    e. Periodically run vulnerability assessment software on all firewalls and servers to detect system configuration weaknesses and the presence of Trojan horses.

Assignment 4 – Design Under Fire

I chose Kelvin Tarrance's practical for this assignment. It can be found at: http://www.sans.org/y2k/practical/Kelvin_Tarrance_GCFW.zip

Here is the network diagram:



Information Gathering:

The first tool we can use to gather some information is nslookup:

```
> set q=ns
> giac.com
Server:  A.GTLD-SERVERS.NET
Address:  192.5.6.30

Non-authoritative answer:
giac.com   nameserver = NS1.giac.com

Authoritative answers can be found from:
NS1.giac.com       internet address = 196.20.59.47
```

At this point we know the name and address of the GIAC DNS server. We can query it for further information.

```
> server ns1.giac.com
Default Server:  ns1.giac.com
Address:  196.20.59.47
>
> set q=mx
> giac.com
Server:  ns1.giac.com
Address:  196.20.59.47

giac.com   preference = 5, mail exchanger = fwmail.giac.com
giac.com   nameserver = ns1.giac.com
mail.giac.com      internet address = 196.20.59.48
```

At this point we know the name of the server handling mail for GIAC. This is probably a firewall.

Next thing we can try is a zone transfer. If we are lucky we get the following:

```
> ls -d giac.com
[ns1.giac.com]
$ORIGIN giac.com.
@               15M IN SOA             @ hostmaster (
                                       2001120501    ; serial
                                       1H            ; refresh
                                       2H            ; retry
                                       1W            ; expiry
                                       15M )         ; minimum

                15M IN NS     ns1
                15M IN MX     5 fwmail
fwmail          15M IN A      196.20.59.48
fwintnet        15M IN A      196.20.59.49
dns             15M IN A      196.20.59.47
fwoutdmz        15M IN A      196.20.59.40
```

Now we can try to telnet to the mail firewall on port 25 and type a helo command:

```
# telnet 196.20.59.48 25
Trying...
Connected to 196.20.59.48.
Escape character is '^]'.
220 fwmail.giac.com SMTP/smap Ready.
helo
250 Charmed, Im sure.
```

There is a very good possibility that the mail server is a Gauntlet firewall because of the smap line and because of the phrase "Charmed, Im sure". These are "signatures" of Gauntlet.

At this point we could run scans against the firewalls and the DNS server. Nmap could be use for that.

We have to be careful doing any of the above because the firewall or IDS might detect us. A good hacker would do the above from a compromised host on a different network.

Attacking the Firewall

Here is a list of Gauntlet vulnerabilities obtained from www.securiteam.com. It is amazing that there are not very many of them, especially when compared to other firewalls such as CheckPoint:

- Buffer overflow with the CyberPatrol daemon that can be used to cause a Denial of Service. It also can be used to remotely execute arbitrary shell commands as root on the firewall.
  http://www.securiteam.com/securitynews/5MP080A1QK.html

- Weak random seed attack. This problem affects all random challenge authentication methods in Unix platform release of both FWTK and Gauntlet 4.x.
  http://www.securiteam.com/unixfocus/2ZUQ0QAQPI.html

- Smap/smapd (Gauntlet 5.x) and CSMAP (Gauntlet 6.x) Buffer overflow. This vulnerability can be used to execute arbitrary shell commands with the privileges of the owner of the corresponding daemon.
  http://www.securiteam.com/unixfocus/5HP041F5FA.html

- NAT mishandling. This vulnerability causes incorrect routable IP addresses to be generated leading to DoS of systems that legitimately own the routable addresses.
  http://www.securiteam.com/securitynews/5GP000A1SE.html

- Gauntlet 5.0 Firewall running under Windows NT server can be made to be inoperable using a simple ftp proxy redirection technique.
  http://www.securiteam.com/windowsntfocus/3W5QCQKPPW.html

  Author's Note: NAI stopped offering Gauntlet for NT in the 6.0 release.

- Gauntlet 5.0 Firewall under BSDI can be bypassed
  http://www.securiteam.com/unixfocus/3P5QEQAPQA.html

To protect against any of the above vulnerabilities, the appropriate patches need to be installed. These patches are available at ftp://ftp.nai.com/pub/security/gauntlet/patches/

The first thing I would try is to attack the email firewall in Kelvin's practical (fwmail). A smap/smpad (Gauntlet 5.5) and CSMAP (Gauntlet 6.0) vulnerability was announced in May, 2001. Since fwmail is a Gauntlet 5.5 firewall and Kelvin's practical was written in July 2001, there is a chance the firewall is not patched and my attack will succeed.

Smap service listens on port TCP 25 for mail. It receives the mail and after checking it for basic rules, it puts it in a directory called /var/spool/smap. Smapd then picks it up and puts it in a directory called /var/spool/mqueue. Smapd then spawns Sendmail to deliver the mail to its destination. PGP combined smap/smpad into one program called CSMAP, which is included in Gauntlet 6.0.

I spent hours and hours looking for an exploit for this vulnerability, and I could not find one. Had I found one, I would've been able to attack the firewall through port 25 and exploit the buffer overflow vulnerability. I would've also been able to execute arbitrary shell commands to do damage to the firewall itself and possibly to hosts on the internal network.

Another possible attack is against the Internet firewall fwintnet. Assuming CyberPatrol daemon is active on this firewall, I could exploit the buffer overflow vulnerability to cause a Denial of Service. Since the CyberPatrol daemon is used in conjunction with the HTTP proxy, crashing the CyberPatrol daemon would cause the HTTP proxy to stop accepting traffic. Also, it is possible to remotely execute arbitrary shell commands as root on the firewall. A lot of damage can be done to the firewall and access to hosts on the internal network is possible.

A proof of concept exploit I could find for this vulnerability is available at http://www.securiteam.com/exploits/5IP020A1SU.html. It is designed to execute as root a shell script called /bin/zz. This exploit was written specifically for BSDI system.

Denial of Service Attack

Tool Choice and Description

The tool I will use for this attack is Shaft. It can be obtained from www.technotronic.com. Interesting enough this site was hacked when I was writing this section.

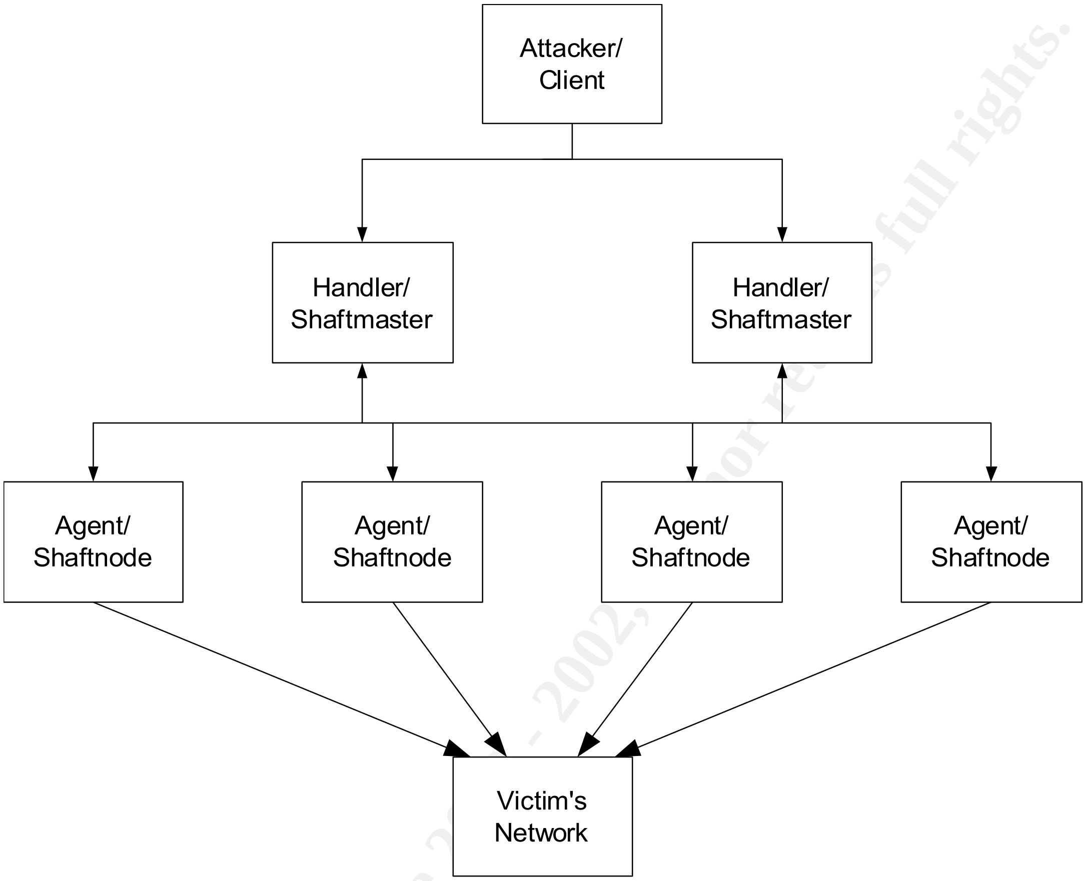
The Shaft DDoS tool was first introduced in November 1999. It belongs in the family of tools such as Trinoo, TFN, Stacheldraht, and TFN2K. Shaft is a packet-flooding tool and the client controls the size of the flooding packets and the duration of the attack. Interesting signature of this tool is that the sequence number for all TCP packets is 0x28374839.

The following pieces are needed to accomplish a Shaft attack:

1. One or more handler programs (shaftmaster)
2. A large number of agents (shaftnode)
3. An attacker who uses a telnet program (client) to communicate with the handlers to launch the attack.

Interesting feature of Shaft is that it collects statistics about the attacked network. It does that by keeping track of the packet generation rate of the agents. This allows the attacker to stop adding hosts to the attack when no more hosts are needed to bring the network down. Why waste more resources when the goal has been achieved! It also allows the attacker to keep adding necessary agents as discovered agents have been taken off line.

A Shaft network would look like this:



Network communications between the pieces occur as follows:

Client to handler(s): 20432/tcp
Handler to agent(s): 18753/udp
Agent to handler(s): 20433/udp

Conducting the Attack

In our case we have already compromised 50 cable modem/DSL systems. Using certain hacking techniques, we have already installed the Shaft handler on two of these machines. The Shaft agent has already been installed on the other 48 machines. 24 agents report to the first handler, and the other 24 agents report to the second handler. We will start the attack by using the first handler. We will use the second handler if we need more agents or if the first handler was taken "off line".

Remote control of the handler is done through a telnet connection. Shaft uses "tickets" for keeping track of the agents. Both passwords and ticket numbers have to match for the agent to execute a request.

When the attack is launched, the agent reports back to its default handler by sending a "new <upshifted password>" command. The default password is shift so the agent would send out the command "new tijgu". All subsequent messages would carry this password in it.

After the agents send the password, the handler and agents enter into a 2-way conversation. Handler sends commands, and agents send back acknowledgements.

Commands from the handler to the agents are in this format:
"cmd password [args] Na Nb"

Acknowledgements from the agents to the handler are in this format:
"cmdrep password Na Nb [args]"

Where:

- password is the upshifted password.
- Na is the socket number. This number remains the same during the attack.
- Nb is the ticket number. This number keeps changing during the attack
- cmd is the command from the handler to the agents
- cmdrep is the command acknowledgement from the agents to the handler
- args are command arguments

We will issue the following commands to the first handler. For illustration purposes we will use the following values:

- "tijgu" for the password
- The number 5 for the socket number
- 198.20.64.20 for the handler
- 197.30.24.20 for one of the agents

+node 197.30.24.20

This command instructs the handler to add new agents. We can use this command to add the 24 agents reporting to this handler.

switch

This command instructs the handler to become the handler for the specified agents. It sends to all agents the command "switch tijgu 198.20.64.20 20433 5 3434". The agents will reply with the acknowledgment "switching tijgu 5 3434 198.20.64.20 20433".

alive

This command instructs the handler to send “alive” to all agents. A possible argument to alive is hi. It sends to all agents the command “alive tijgu hi 5 7120”. The agents will reply with the acknowledgment “alive tijgu 5 7120 blah”.

time 500

This command instructs the handler to set the duration of the attack to 500 seconds. It sends to all agents the command “time tijgu 700 5 8756”. The agents will reply with the acknowledgement “time tijgu 5 8756 700”.

size 8192

This command instructs the handler to set the packet size to the maximum allowed which is 8k. It sends to all agents the command “size tijgu 8192 5 6231”. The agents will reply with the acknowledgment “size tijgu 5 6231 8192”.

type UDP TCP ICMP

This command instructs the handler to set type of attack to UDP, TCP, and ICMP. It is also possible to use only one of these types, or a combination of 2. It sends to all agents the command “type tijgu 2 5 8901”. The agents will reply with the acknowledgment “type tijgu 5 8901 2”. The value 2 means UDP, TCP, and ICMP. 0 means UDP, 1 means TCP, and 3 means ICMP.

mdos <host list>

This command instructs the handler to start the DDoS attack. It sends “own host” messages to all agents reporting to this handler. To attack the border router in Kelvin’s practical we can send to the handler the command “mdos 200.40.159.19”. The handler sends to all agents the command “own tijgu 200.40.159.19 5 5466”. The agents will reply with the acknowledgment “owning tijgu 5 5466 200.40.159.19”.

Detecting the Attack

The following can be done to detect a Shaft attack on the network:

- Scan the network for open port 20432 to detect the presence of a handler.
- Scan the network for open port 18753 to detect the presence of an agent.
- Write a program to send out “alive” messages to all nodes on the network on UDP port 18753. This will help in detecting any agent that responds to the “alive” command.
- Since the traffic between the handler and the agents is not encrypted, an attack can be detected based on certain keywords such as alive and pktstat (these are commands the handler sends to the agents).
- Look for TCP packets with sequence number of 0x28374839.
- Use DDoS scanners such dds by Dave Dittrich (http://staff.washington.edu/dittrich/misc/ddos_scan.tar) or RID by David Brumley (http://www.theorygroup.com/Software/RID/).

Protecting Against the Attack

- Use rate-limiting feature on the border router to protect against ICMP packet flooding attacks.
- Use anti-spoofing filters at the border router.
- Use an offsite data center during an attack to keep resources available.
- Always keep systems up to date on patches.
- Follow CERT/CC and SANS best practices.

.

Appendix A – Gauntlet 5.5 Tutorial

Introduction:

Gauntlet 5.5 is a proxy firewall. Gauntlet itself ships with many proxies such as HTTP, FTP, Telnet, and many more. It also contains a feature called plugs where you can define your own services. A plug is not a full-blown proxy since it is generic and does not fully understand a specific protocol. The plug goes as high as layer 4 on the OSI model where a proxy goes as high as layer 7. Both proxies and plugs provide a built in NAT. All packets leave the firewall with the firewall's external IP address as the source address.

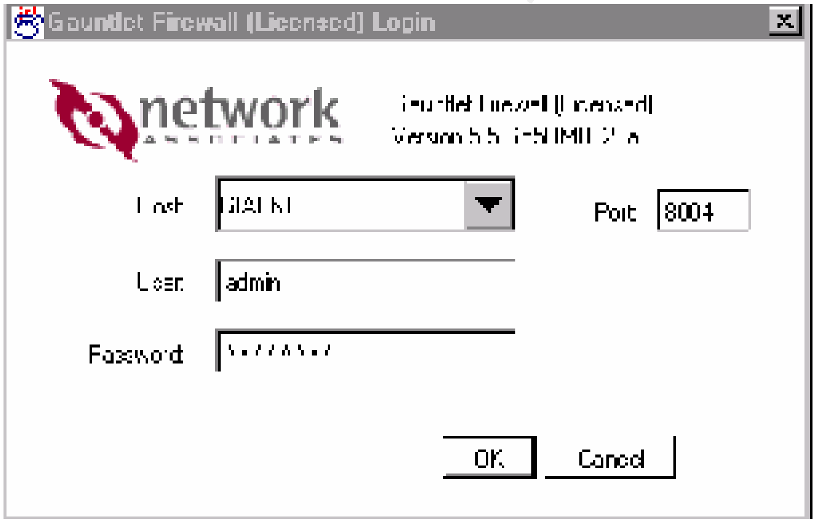
In order to develop Gauntlet 5.5 rules, you must understand the following three concepts:

1. Network Groups: These are the network groups that contain the network addresses that are allowed to use certain services.

2. Service Groups: These are the service groups that contain the services the network groups are allowed to use.

3. Destination Rules: These are the rules that define where the service groups are allowed to go.

When you first login to Gauntlet 5.5 you see the following screen:

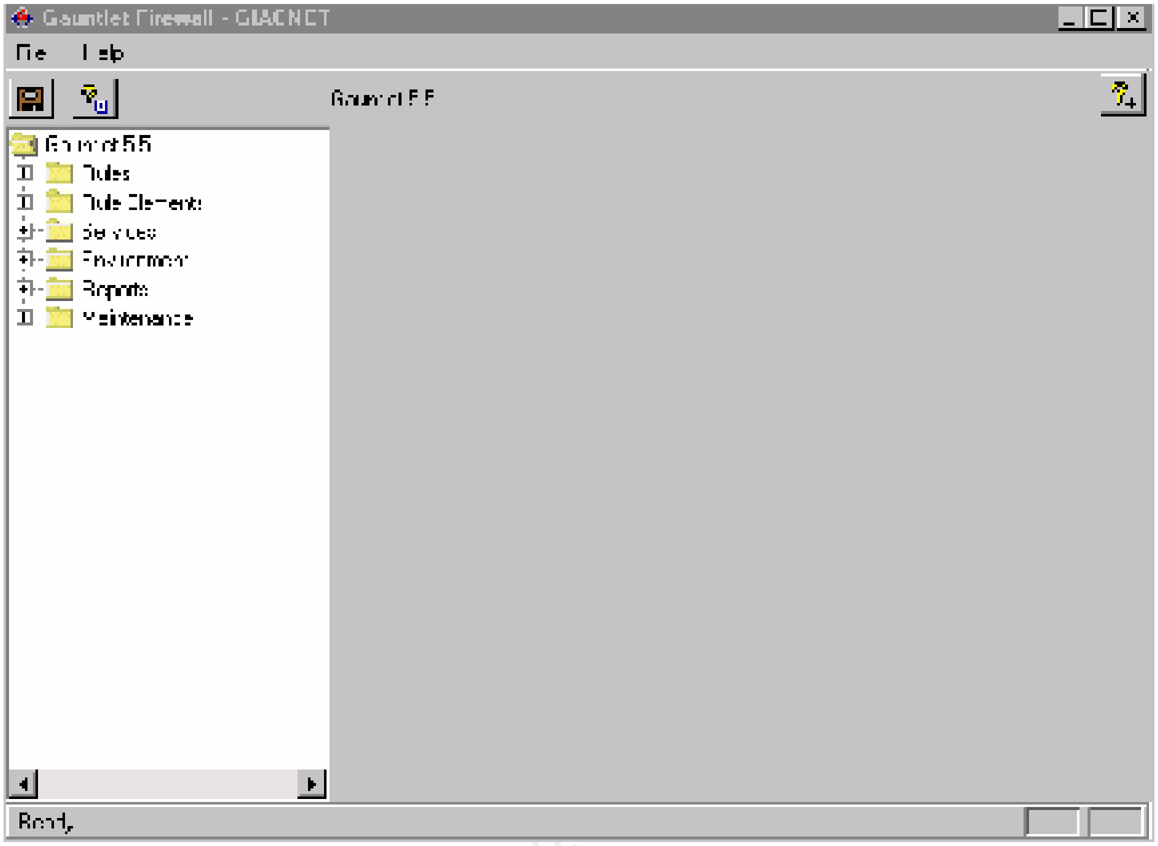


Gauntlet Firewall (Licensed) Login

Gauntlet Firewall (Licensed)
Version 5.5 [illegible]

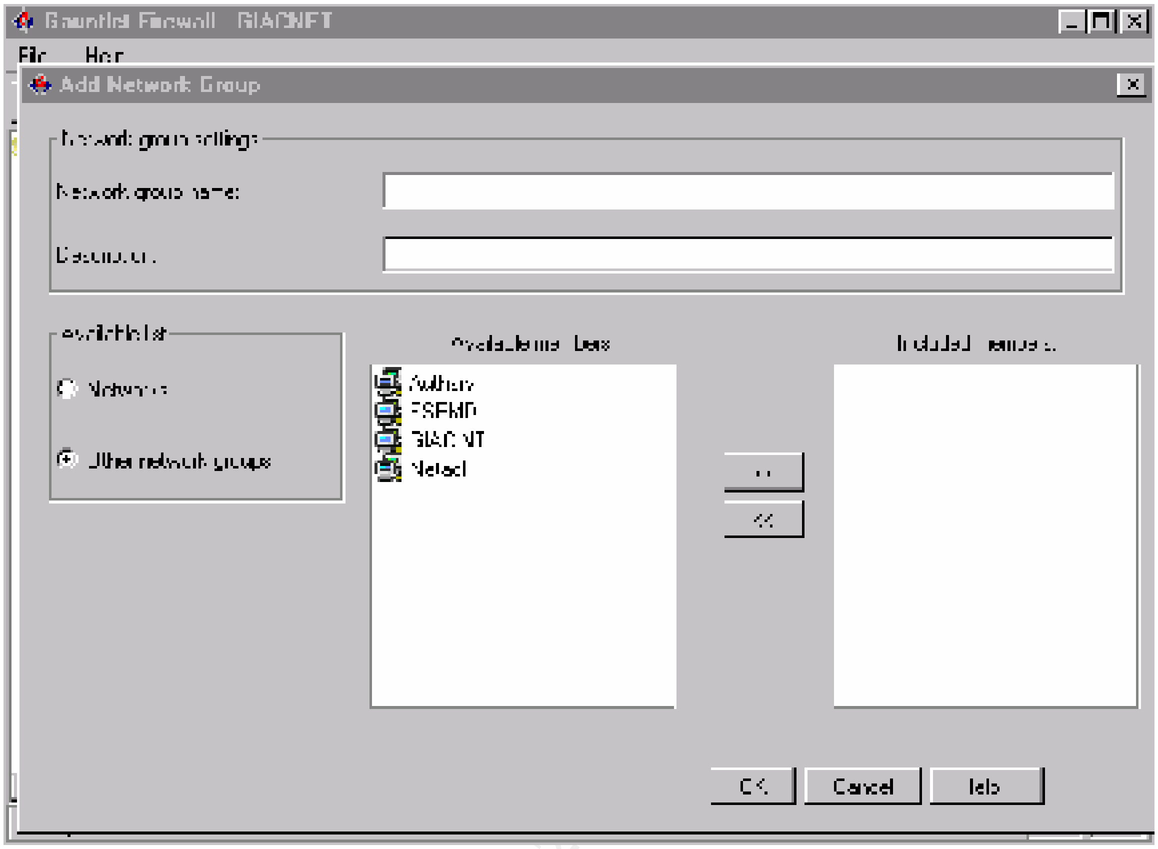
Host: [illegible]
Port: 8004
User: admin
Password: ********

After providing the firewall name (or IP address), user name, and password, click on OK. The main Gauntlet 5.5 screen appears:



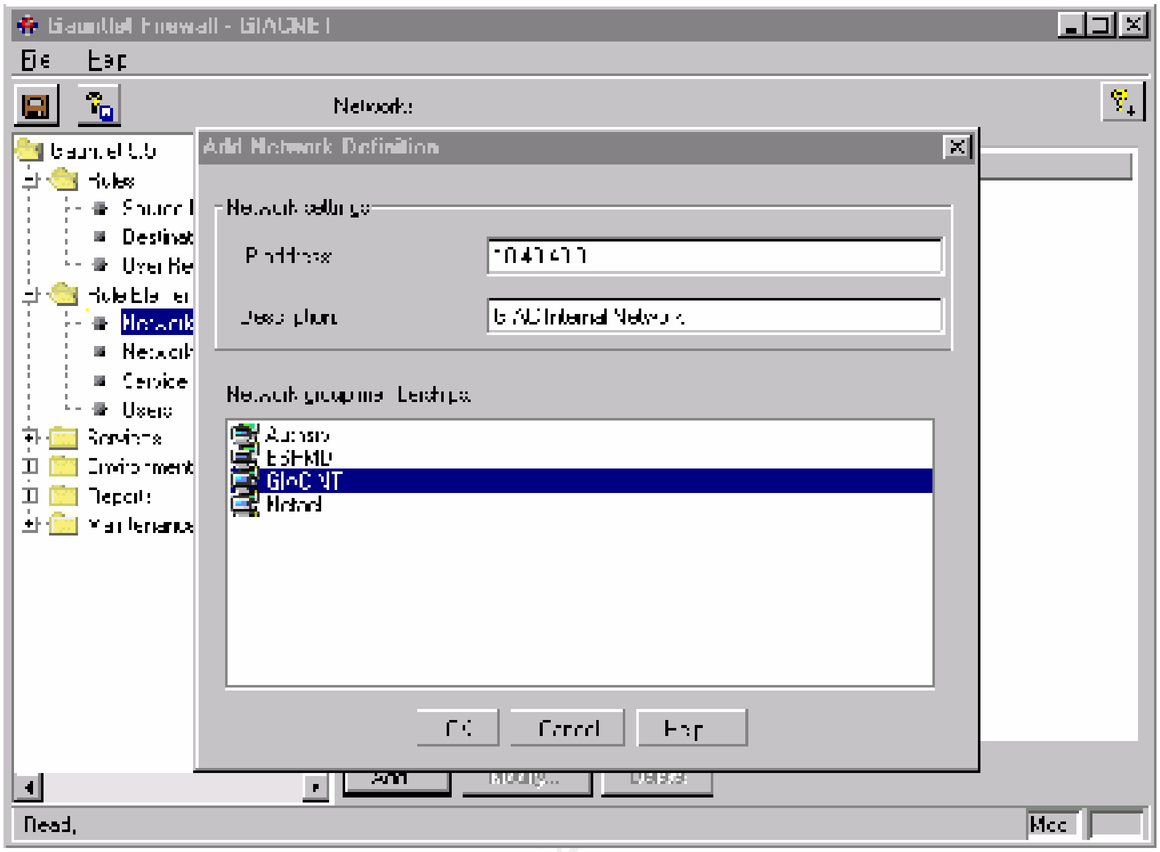
Creating a Network Group:

To create a network group, click on the plus sign next to Rule Elements, then click on Network Groups, and then click on Add. The following screen appears:



To create a network group, a unique network group name must be provided. You may also provide a description of the network group. Click OK when done. A network group must be created before it can be populated.

To create a network object, from the main screen click on the plus sign next to Rule Elements, then click on Networks, and then click on Add. The following screen appears:



Gauntlet Firewall - GIACNET

Add Network Definition

Network settings

IP Address: [illegible]

Description: GIAC Internal Network

Network group memberships

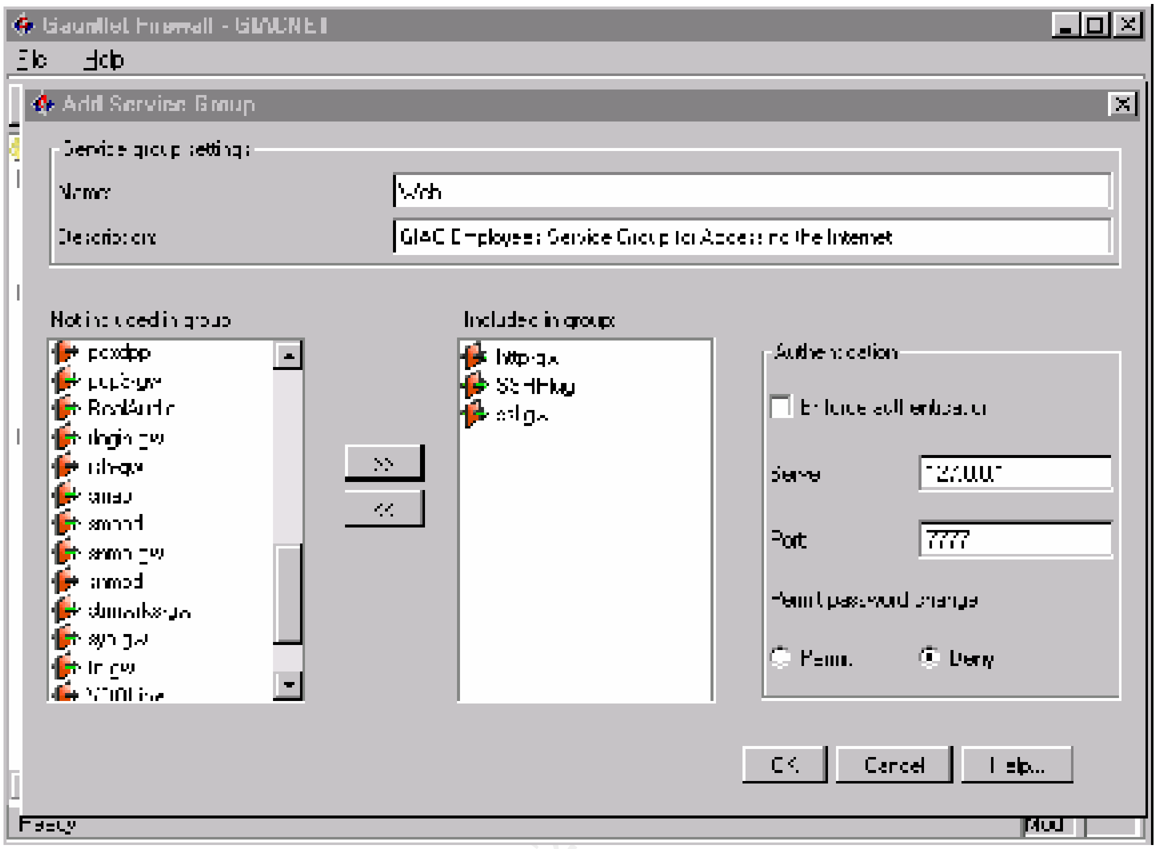
- Admins
- ESHMD
- GIACNET
- Hotord

In the IP address field, enter either a host IP address or a network address as shown in the screen above. You may also enter a description of the network address. It is important to associate the IP address with a network group. Click on the appropriate group name in the network group memberships box and then click on OK.

After you define the needed network groups, it is time to define the needed service groups.

Creating a Service Group:

To create a service group, from the main screen click on the plus sign next to Rule Elements, then click on Service Groups, and then click on Add. The following screen appears:



Service group settings

Name: Web

Description: GIAC Employees Service Group for Access to the Internet

Not included in group:

- pcxdpp
- pop3-gw
- RealAudio
- rlogin-gw
- rsh-gw
- smap
- smapd
- snmp-gw
- snmpd
- strmworks-gw
- syb-gw
- tn-gw
- VDOLive

Included in group:

- http-gw
- SSHPlug
- ssl-gw

Authentication

☐ Enforce authentication

Server: 127.0.0.1

Port: 7777

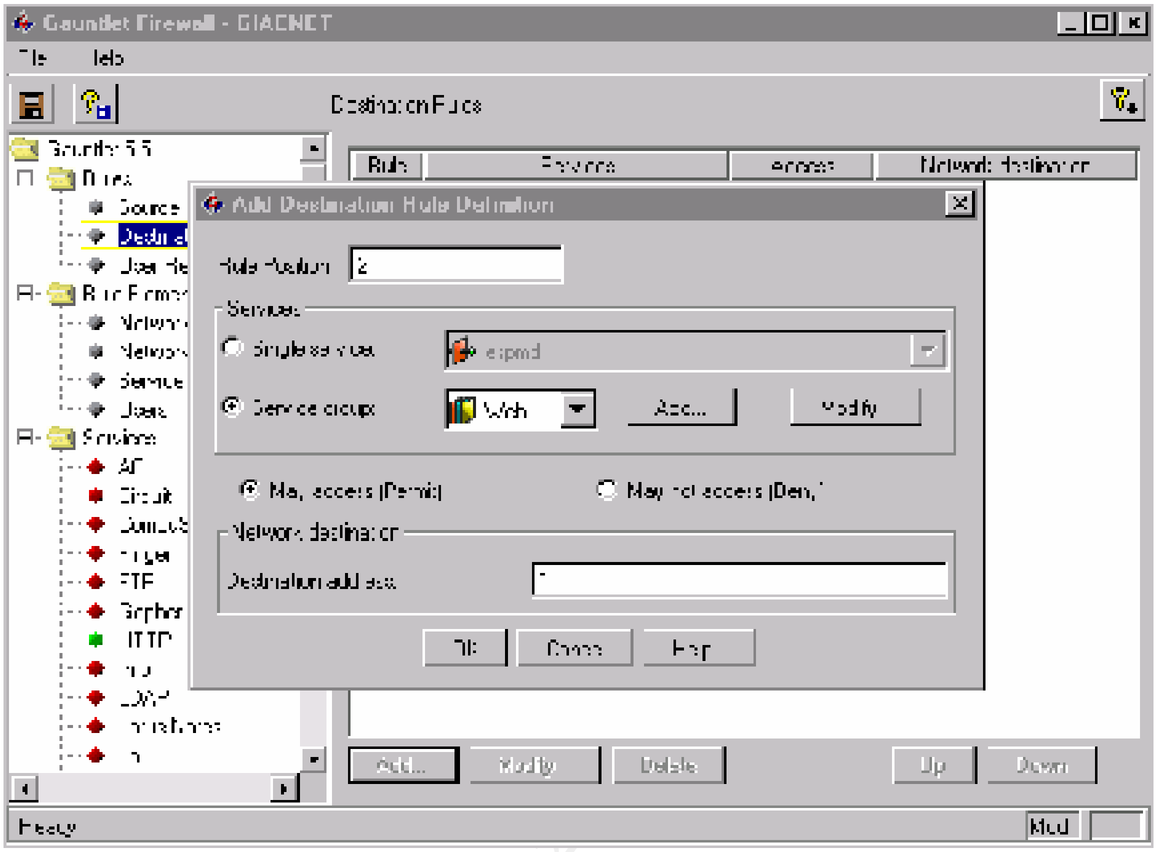
Permit password change: ○ Permit ● Deny

A unique name for the service group must be provided. A description is optional but it is a good practice to use one. All services available are listed in the box titled “Not included in group:” These are the services provided in Gauntlet by NAI or the custom plugs built by the firewall administrator.

To include a service in the service group, click on the service name in the “Not included in group:” box, and then click on the box with the 2 right arrows. The service will then show up in the box titled “Included in group:” To remove a service from the service group, click on the service name in the “Included in group:” box, and then click on the box with the 2 left arrows. The service will then “move” from the “Included in group:” box to the “Not included in group:” box. After adding all the needed services, click on OK.

Creating a Destination Rule:

It is time now to define the destinations a service group is allowed to get to. To create a destination rule, from the main screen click on the plus sign next to Rules, then click on Destination Rules, and then click on Add. The following screen appears:



Rule Position: 2

Services

○ Single service: epmd

● Service group: Web

● May access (Permit)   ○ May not access (Deny)

Network destination

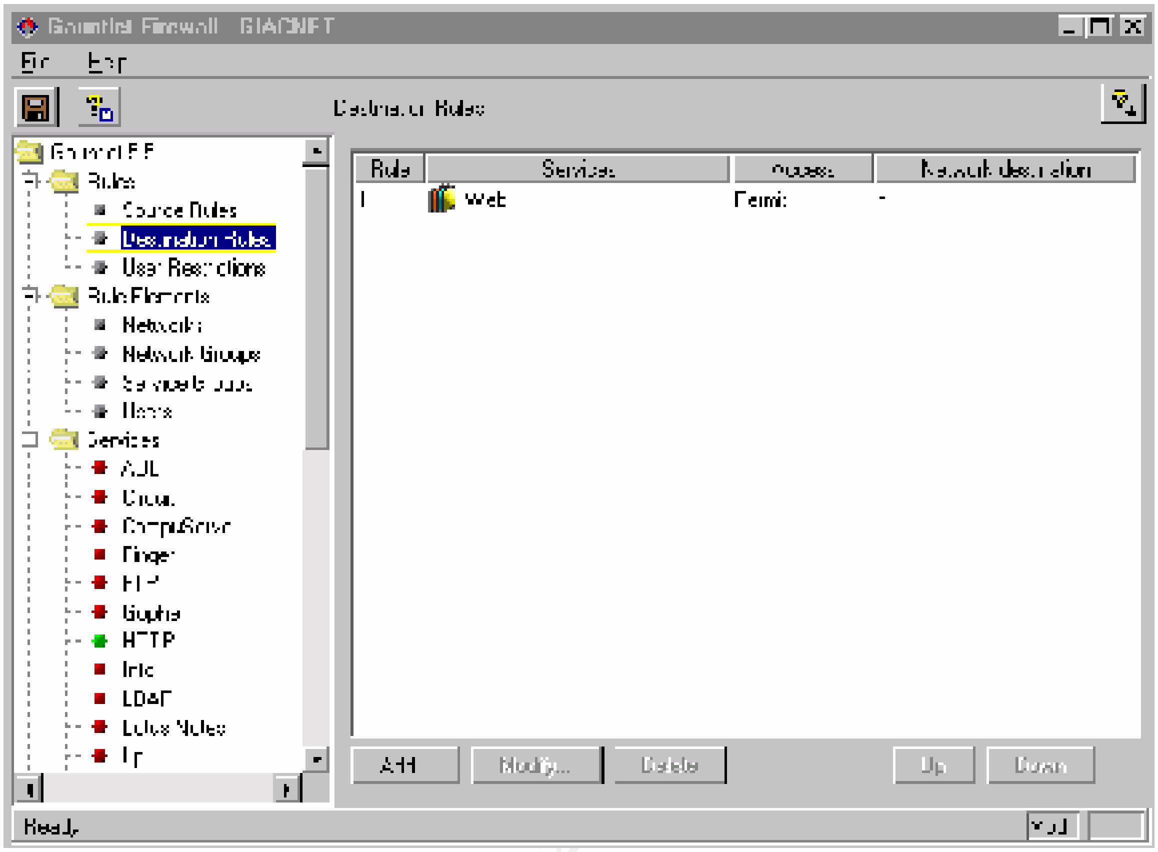
Destination address: *

Click the radio button next to Service group. Opening the pull down box next to the Service group will list all the service groups that have already been defined. Highlight the service group you are adding the destination rule for. In this example it is the service group Web. Click on the “May access” radio button to allow the service group to get to a certain destination address, or click on the “May not access” radio button to deny the service group from getting to a certain destination address.

In the “Destination address:” filed, enter the IP address or the network the service group is allowed to go to. In this example it is “*” which means the service group can go anywhere.

Click OK when done.

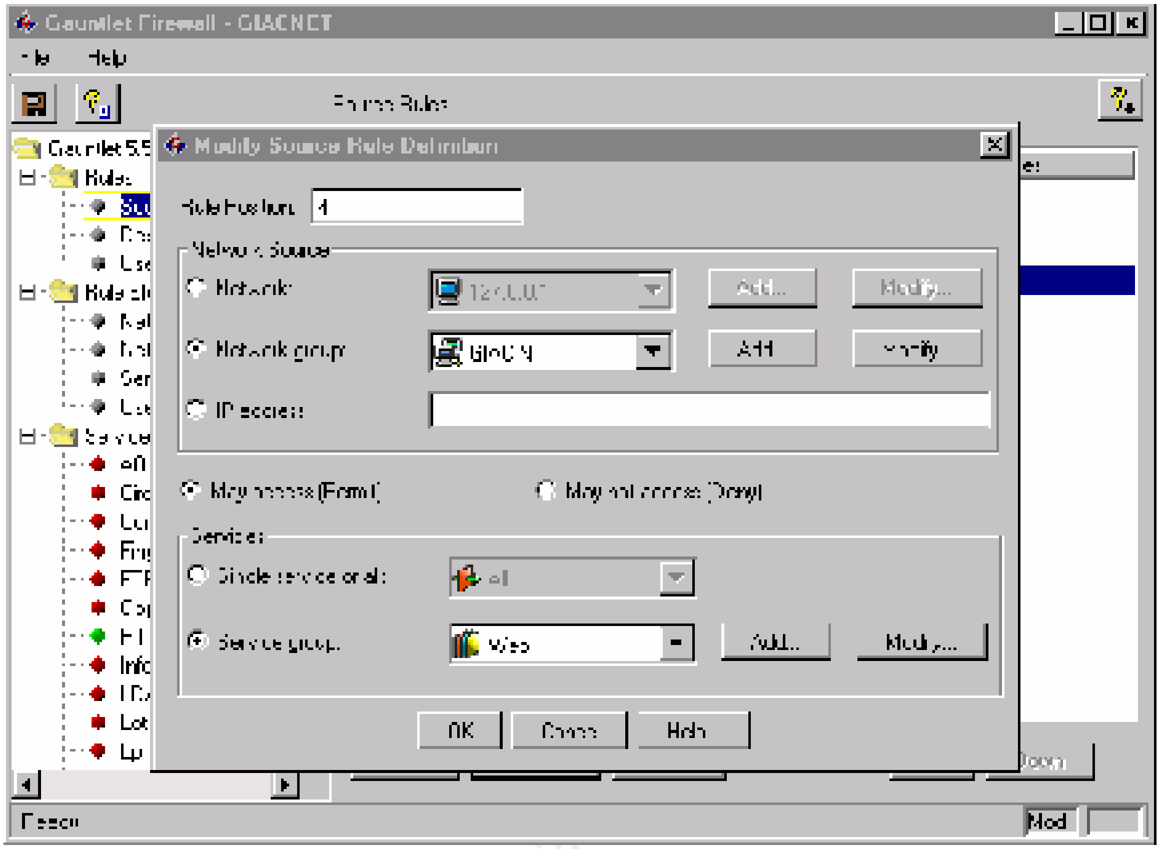
If more destination addresses are needed, this process will need to be repeated for each destination. When done the screen will look like this:



Rule	Services	Access	Network description
1	Web	Permit	-

Creating a Source Rule:

To complete the process, a rule is needed to tie the network group to the service group. To do that, from the main screen click on the plus sign next to Rules, then click on Source Rules, and then click on Add. The following screen appears:



Rule Position: 4

Network Source

○ Network: 127.0.0.1

● Network group: GIACINT

○ IP address:

● May access (Permit) ○ May not access (Deny)

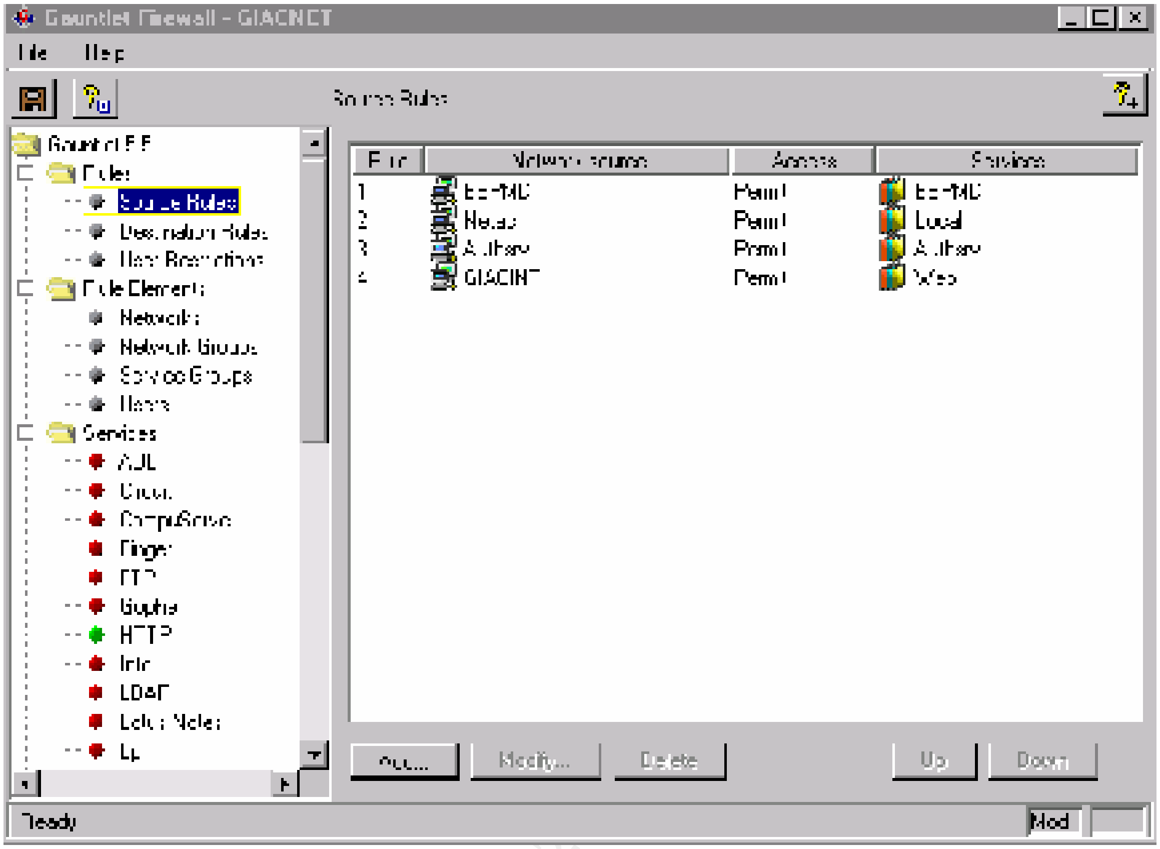
Services

○ Single service or all: all

● Service group: Web

Click the radio button next to Network group. Opening the pull down box next to the Network group will list all the network groups that have already been defined. Highlight the network group you are adding the source rule for. In this example it is the network group GIACINT. Click on the "May access" radio button to allow the network group to access a certain service group, or click on the "May not access" radio button to deny the network group from accessing a certain service group.

Click the radio button next to Service group. Opening the pull down box next to the Service group will list all the service groups that have already been defined. Highlight the service group you are adding the source rule for. In this example it is the service group Web. Click OK when done. The screen will look like this:



Rule	Network source	Access	Services
1	[illegible]	Permit	[illegible]
2	[illegible]	Permit	Local
3	Authsrv	Permit	Authsrv
4	GIACINT	Permit	Web

The end result is that we have a network group called GIACINT that is allowed to use the service group Web that is allowed to go anywhere.

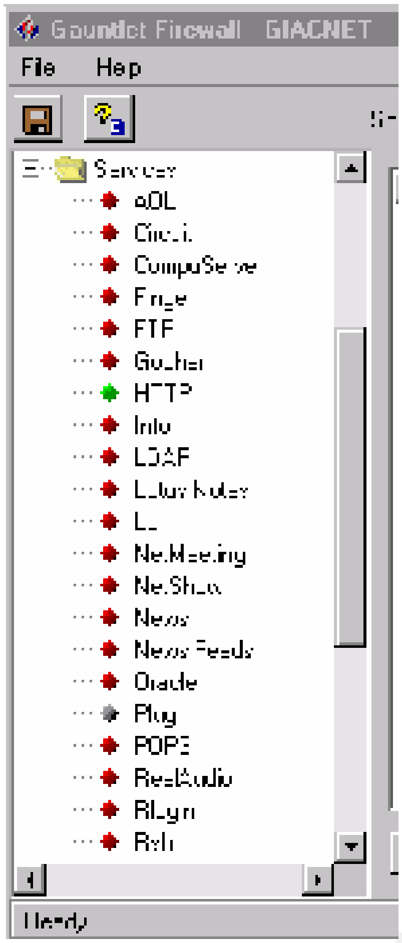
In this example we have one member in the network group. The member is the network 10.40.40.0. The services allowed in the service group are HTTP, SSL, and SSH. So, the firewall is allowing anyone on the internal network of GIAC to be able to use the services HTTP, SSL, and SSH to go anywhere.

If you do a netstat on the firewall, you see the following:

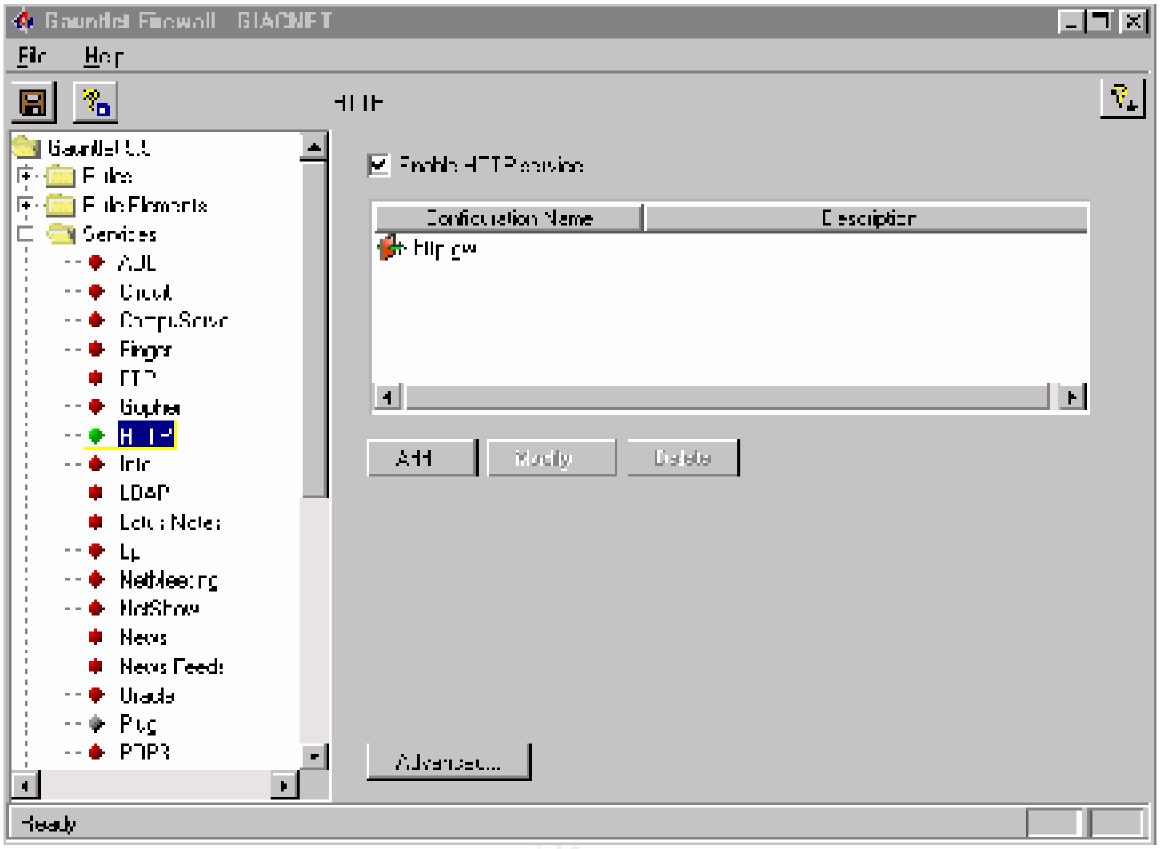
```
# netstat -an | grep LISTEN
tcp     0     0  *.443          *.*             LISTEN
tcp     0     0  *.80           *.*             LISTEN
tcp     0     0  *.22           *.*             LISTEN
```

Which means that HTTP, SSL, and SSH are all listening and accepting traffic on these ports.

Services:

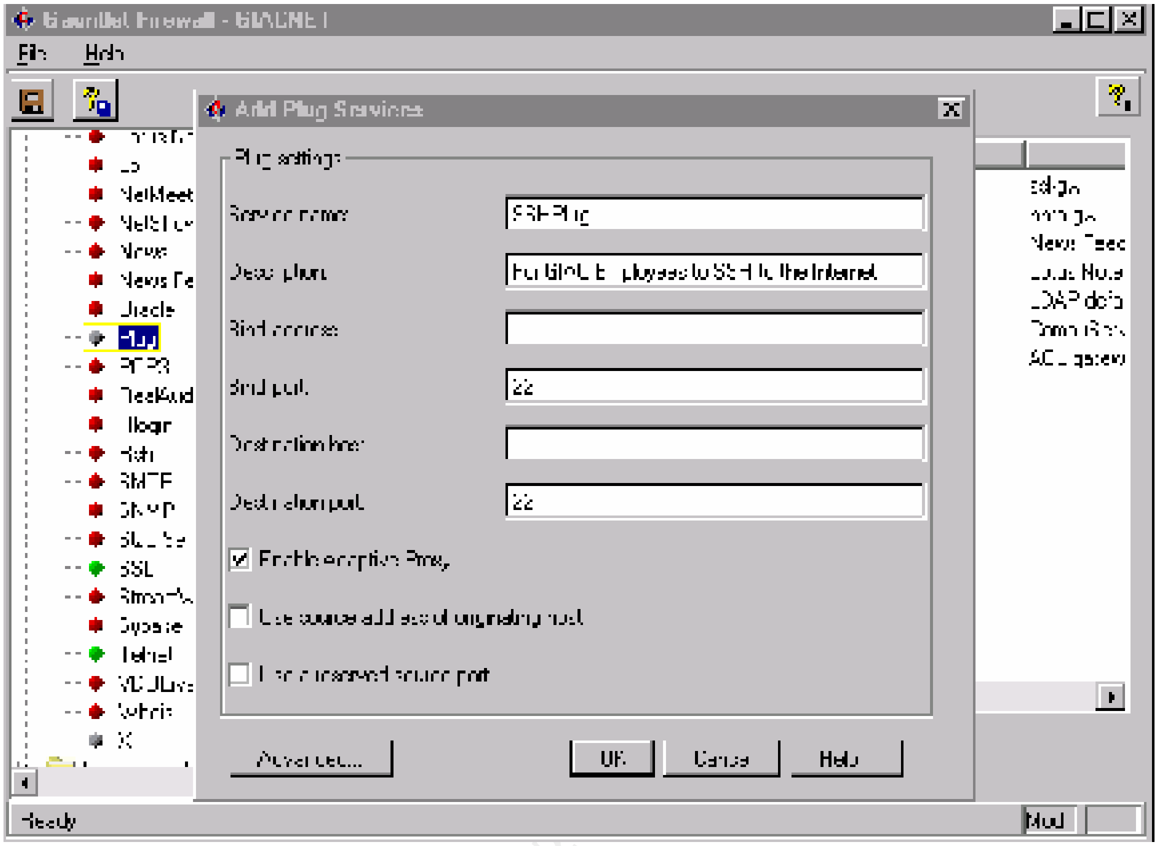


The above screen lists all the services that are available for use on the firewall. These services come pre-configured with Gauntlet. Since everything is denied in Gauntlet unless specifically permitted, you need to enable the needed services. A red button means the service is disabled, and a green button means it is enabled. To get to the above screen, from the main screen click on the plus sign next to Services. To enable a service, click on it, and then click on the enable box as shown in the following screen:



Building a Plug:

As I mentioned before, Gauntlet provides you the ability to build your own services. They call these plugs. To create a plug, from the main screen click on the plus sign next to Services, then click on Plug, and then click on Add. The following screen appears:



Service name: SSH Plug

Description: For GIAC Employees to SSH to the Internet

Bind address:

Bind port: 22

Destination host:

Destination port: 22

☑ Enable encryption Proxy

☐ Use source address of originating host

☐ Use a reserved source port

You need to provide the following fields:

- Service name. A unique name to identify the plug.

- Description: A one-line description of the plug.

- Bind Address: You can provide an IP address to have the plug bound to. You can use the firewall address (internal or external) or an alias bound to one of the firewall interfaces. If an IP address is provided, then the plug listens only on that IP address. Otherwise it will listen on * which means it will listen on all interfaces.

- Bind port: This is the port the plug will listen on.

- Destination host: You can have the plug do a hand-off to a host which means the plug can be used to communicate to that one host only. You can leave that field empty which means the plug is transparent and can talk to any host.

- Destination port: Gauntlet allows you to do port translation. You can have the plug listen on a port and then have it talk to a host on a different port.

- Enable Adaptive Proxy: This is a new feature in Gauntlet 5.5 that is supposed to increase performance. Instead of handling all packets at the plug level, only checking the rules will be handled at the plug level. When it is time to transfer the data, temporary packet filters will be established. A plug runs in the user area where packet filters run in the kernel area and thus the enhanced performance. Packet filters will be explained a little bit later

- Use source address for originating host: If this field is checked then the plug will not NAT the source address

- Use a reserved source port: You would only need to use this option in the rare case when a server requires the source port to be the same as the destination port. This is hardly ever the case.

Packet Filters:

Even though Gauntlet 5.5 is a proxy type firewall, it does provide the ability to create packet filters. Some applications cannot work with proxies or plugs and require packet filters. For example, you have to create packet filters to allow IPSEC traffic through the firewall.

Packet filters are not "Statefull" which means you need to create one packet for the traffic leaving the firewall and another packet for the traffic coming back to the firewall.

Packet filters traverse the OSI model until layer 4. They check each packet against the rules to verify the source and destination IP addresses and also the port number.

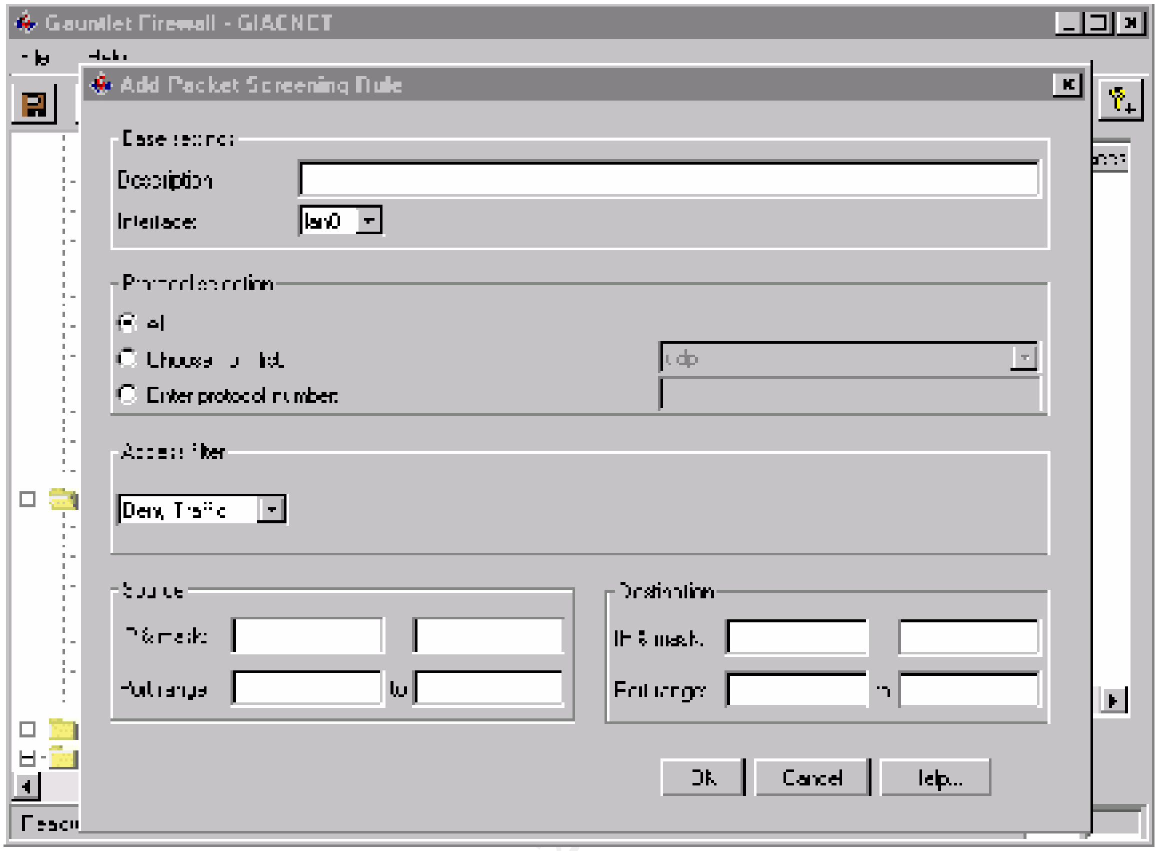
Packet filters run in the kernel module of the firewall. They are much faster than plugs or proxies.

Gauntlet 5.5 provides the capability to create 2 types of packet filters

a. Forward filters – They handle traffic passing through the firewall
b. Local filters – They handle traffic destined to the firewall

Only forward filters are explaining in the next slides.

To create a packet filter, from the main screen click on the plus sign next to Environment, then click on Fwd Filter Rules, and then click on Add. The following screen appears:



Interface: lan0

● All
○ Choose from list
○ Enter protocol number:

udp

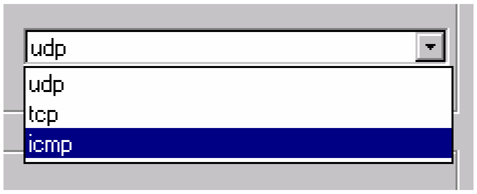
Deny Traffic

You need to provide the following fields:

- Description: Define the purpose of the filter.

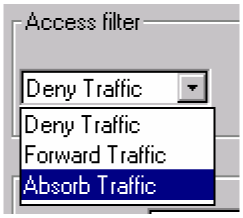
- Interface: Choose the interface the filter is going to apply to. If the filter is for traffic flowing from the trusted to the untrusted network, the filter is applied to the internal interface. If the filter is for traffic flowing from the untrusted to the trusted network, the filter is applied to the external interface.

- Protocol selection: You can specify all protocols, choose from a list, or provide a protocol number. If you click on the Choose from list radio button, then you see the following options in the drop down box located in the right section of the screen:



You may choose UDP, TCP, or ICMP protocol.

- Access Filter: The drop down box provides the following options:

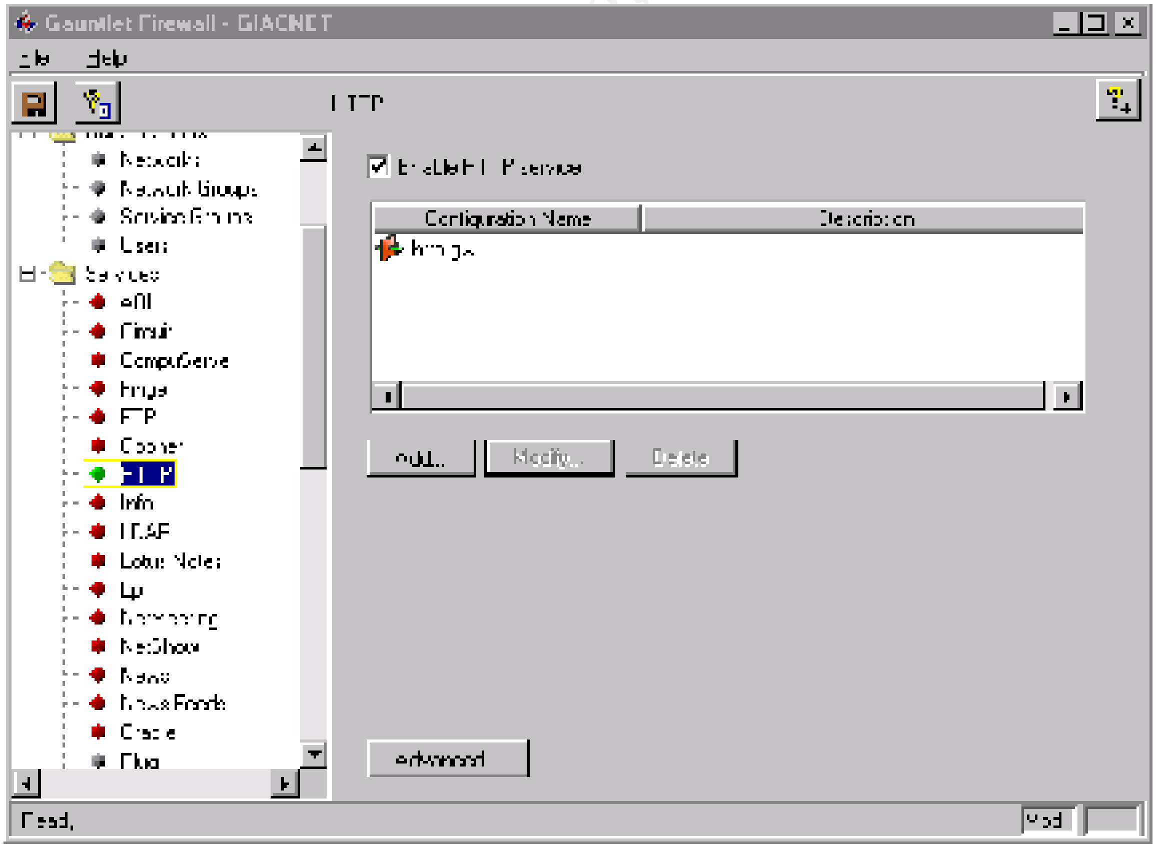


You may either deny the traffic, forward it through the firewall, or absorb it to the firewall so it gets checked by the plug and/or proxy rules.

Sample Proxy (HTTP Proxy):

The following few slides explain how to setup the HTTP proxy in Gauntlet 5.5.

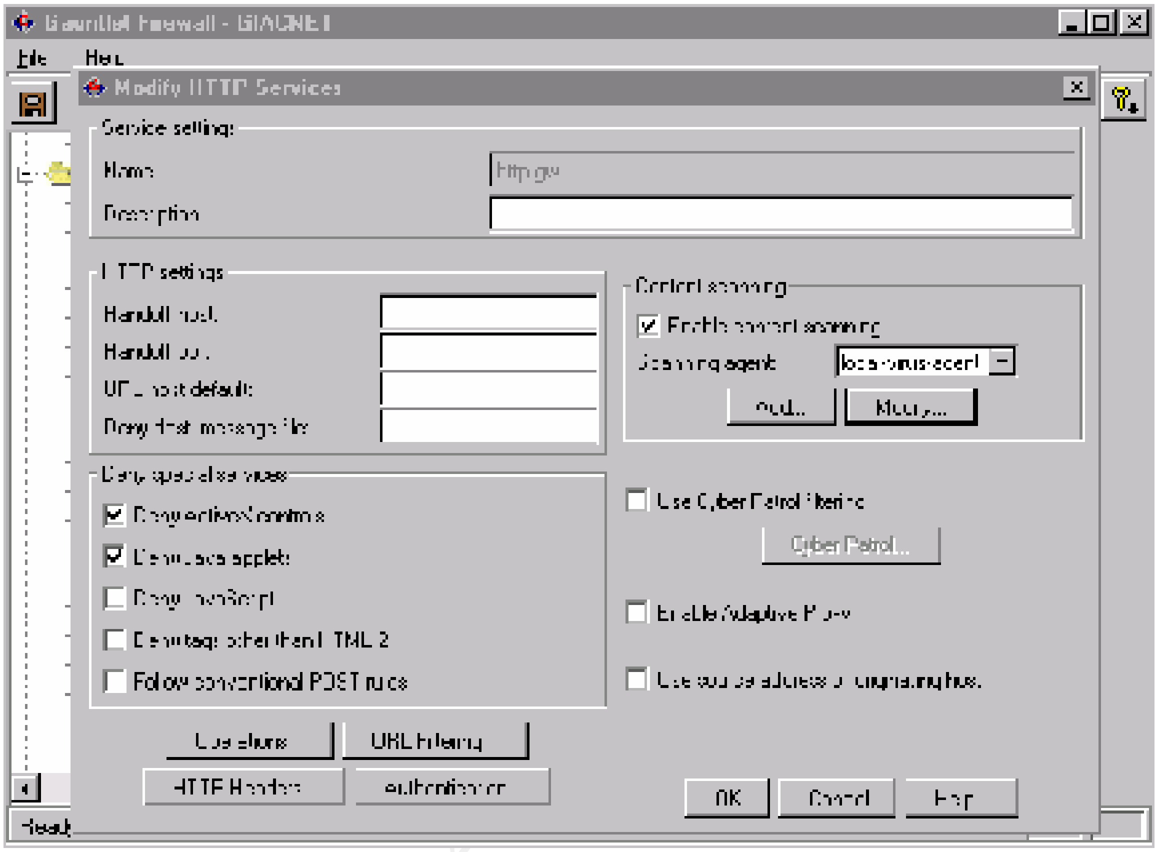
From the main screen, click on the plus sign next to Services, and then click on HTTP. The following screen appears:



☑ Enable HTTP service

Configuration Name	Description
http-gw	

The “Enable HTTP service” is checked to enable the service. Click on the HTTP-gw under the column heading “Configuration Name” and then click on modify. The following screen appears:



Service settings

Name: http-gw

Description:

HTTP settings

Handoff host:

Handoff port:

URL host default:

Deny first message file:

Content scanning

☑ Enable content scanning

Scanning agent: local-virus-agent

Deny special services

☑ Deny ActiveX controls

☑ Deny Java applets

☐ Deny JavaScript

☐ Deny tags other than HTML 2

☐ Follow conventional POST rules

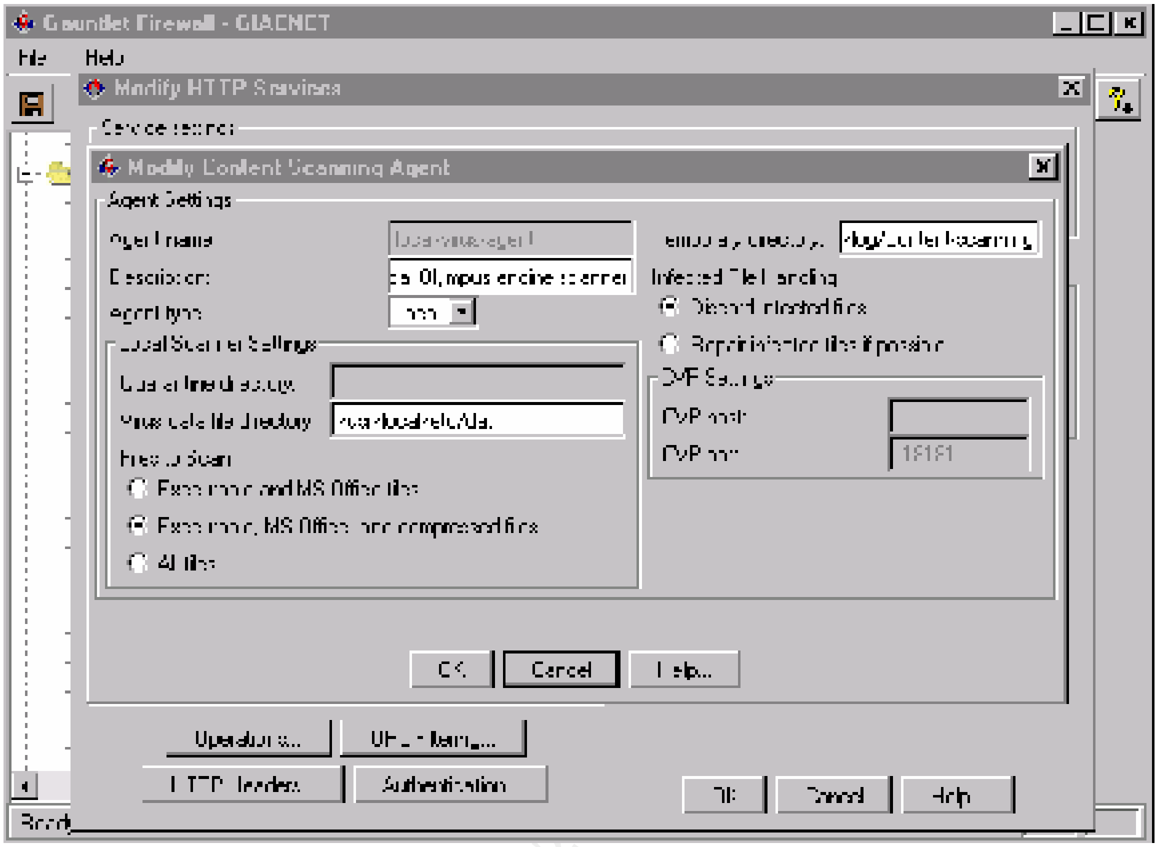
☐ Use Cyber Patrol filtering

☐ Enable Adaptive Proxy

☐ Use source address of originating host

A lot of configurations can be done from the above screen. The HTTP settings section does not get used much because it limits the destination to only one host. The Deny special services section is a very important security feature in the HTTP proxy. This is where you can ask the proxy to block ActiveX, Java, JavaScript, tags other than HTML2, and to follow conventional POST rules. The 2 most popular features are the deny ActiveX controls and Java applets.

Gauntlet 5.5 provides a free built in anti-virus engine based on the Mcafee product. To configure the anti-virus feature, click on the Modify button in the Content scanning box. The following screen appears:



Agent Settings

Agent name: localvirusagent1

Description: ...s Olympus engine scanner

Agent type: local

Temporary directory: /tmp/contentscanning

Infected File Handling

● Discard infected files

○ Repair infected files if possible

CVP Settings

CVP host:

CVP port: 18181

Local Scanner Settings

Quarantine directory:

Virus data file directory: /usr/local/etc/dat

Files to Scan

○ Executable and MS Office files

● Executable, MS Office, and compressed files

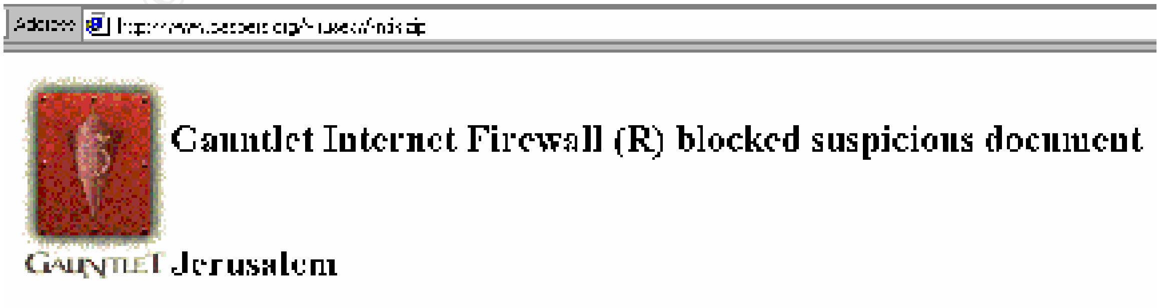
○ All files

On the above screen you can choose which files to scan. Executable & MS Office files, Executable, MS Office, & compressed files, or all files.

You can also decide whether to discard or repair infected files. If you choose to repair infected files, then you have to provide the quarantine directory name.

Gauntlet 5.5 provides a nice script called /usr/local/etc/dat/dat-update.sh to automate pulling the DAT file from ftp.nai.com. You have to be on cluster patch 14 for the script to work in non-interactive mode.

The anti-virus feature is available in other proxies such as the FTP and SMTP proxies. Gauntlet displays the following message when a virus is detected through HTTP:

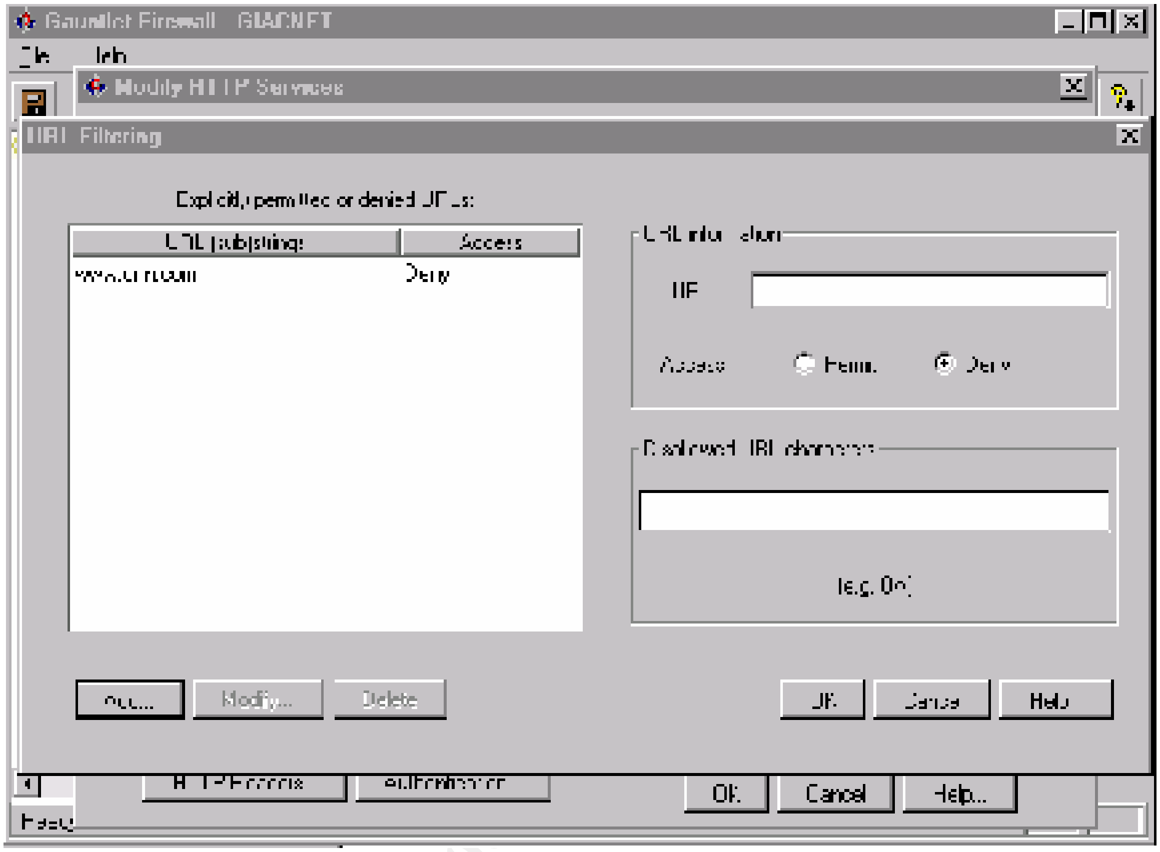


http://www.[illegible]

Gauntlet Internet Firewall (R) blocked suspicious document

Jerusalem

Another feature that can be achieved through the HTTP proxy is URL filtering. From the Modify HTTP Services screen click on URL Filtering. The following screen appears:



Explicitly permitted or denied URLs:

URL (substring)	Access
www.cnn.com	Deny

URL information

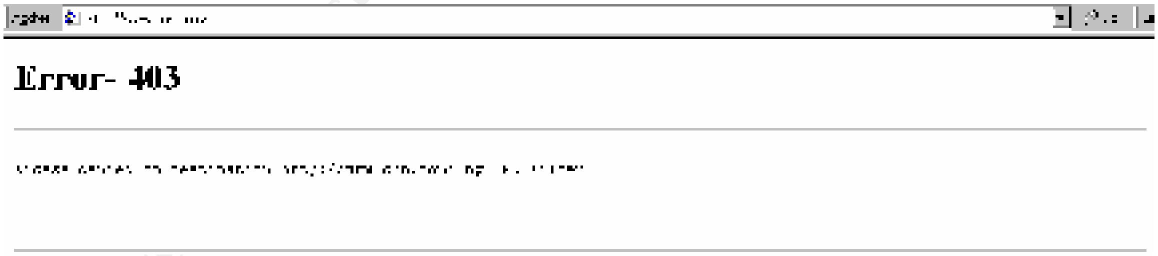
URL

Access ○ Permit ● Deny

Disallowed URL characters

(e.g. 0x)

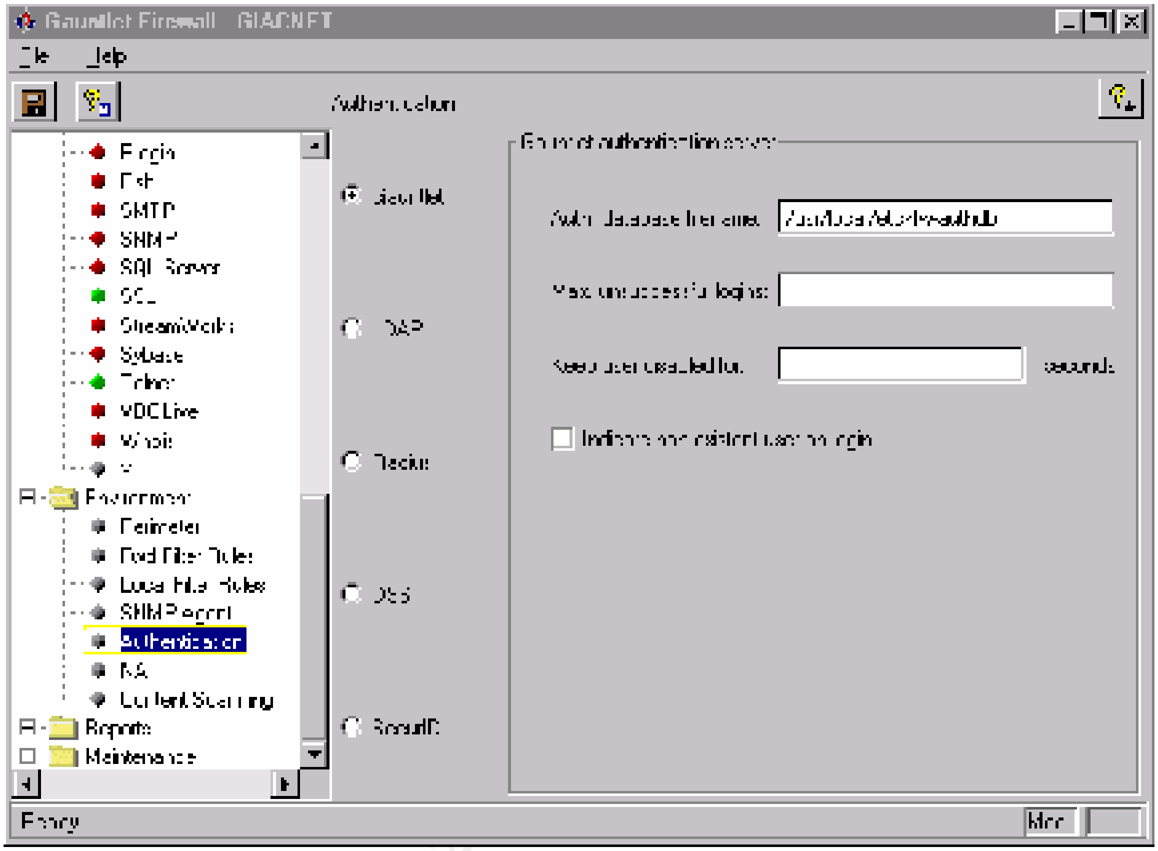
In the above screen you can block a URL by entering the name in the URL box, click on the Deny radio box, and then click on Add. Gauntlet displays the following message when the user tries to go to the blocked site:



Error- 403

[illegible]

Another feature the HTTP proxy provides is Authentication. From the main screen, click on the plus sign next to Environment, and then click on Authentication. A variety of options appear on the right side as shown in the following screen:



● Gauntlet
○ LDAP
○ Radius
○ DSS
○ SecurID

Gauntlet authentication server

Auth database file name: /usr/local/etc/fw-authdb

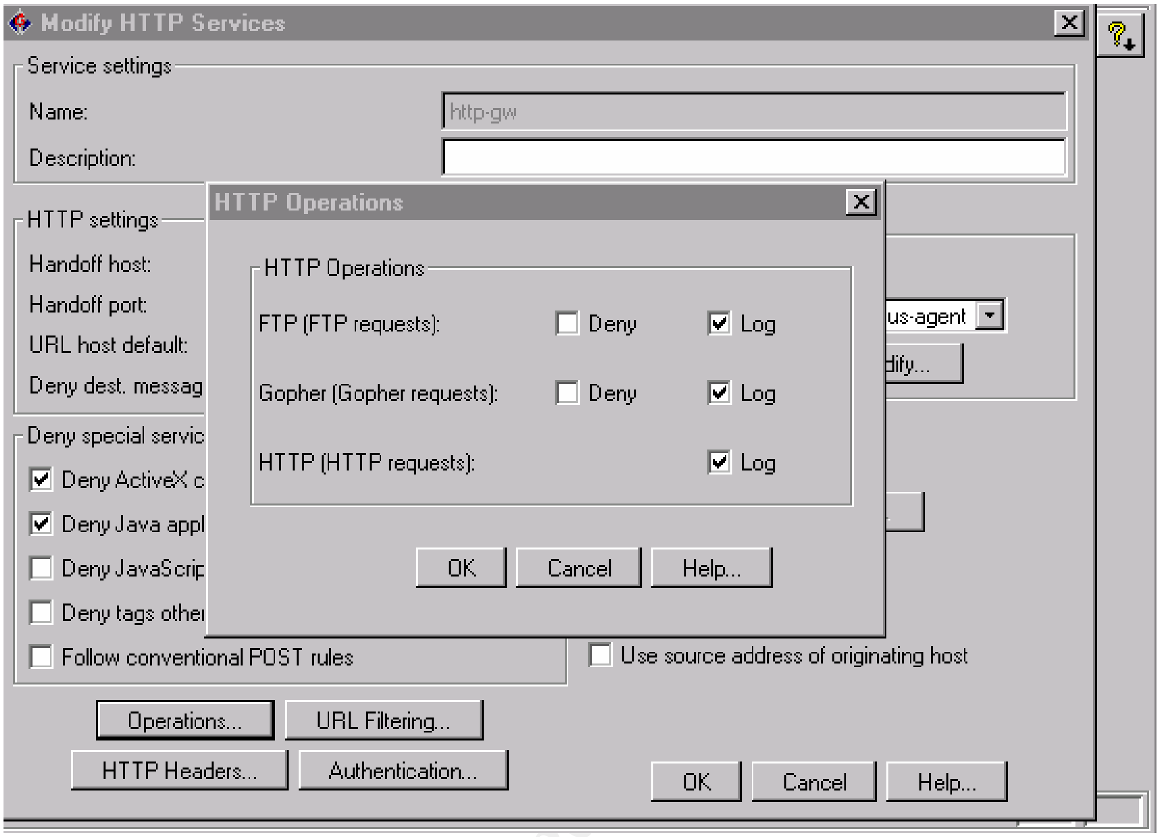
Max unsuccessful logins:

Keep user disabled for: seconds

☐ Indicate non existent user on login

Authentication through Gauntlet, LDAP, Radius, DSS, and SecureID are all supported.

Another nice feature provided by the HTTP proxy is the ability to deny and log FTP and Gopher requests coming through the HTTP proxy as shown in the next screen. To get to this screen, from the "Modify HTTP Services" screen click on Operations:



Modify HTTP Services

Service settings

Name: http-gw

Description:

HTTP settings

Handoff host:

Handoff port:

URL host default:

Deny dest. messag

Deny special servic

☑ Deny ActiveX c

☑ Deny Java appl

□ Deny JavaScrip

□ Deny tags other

□ Follow conventional POST rules

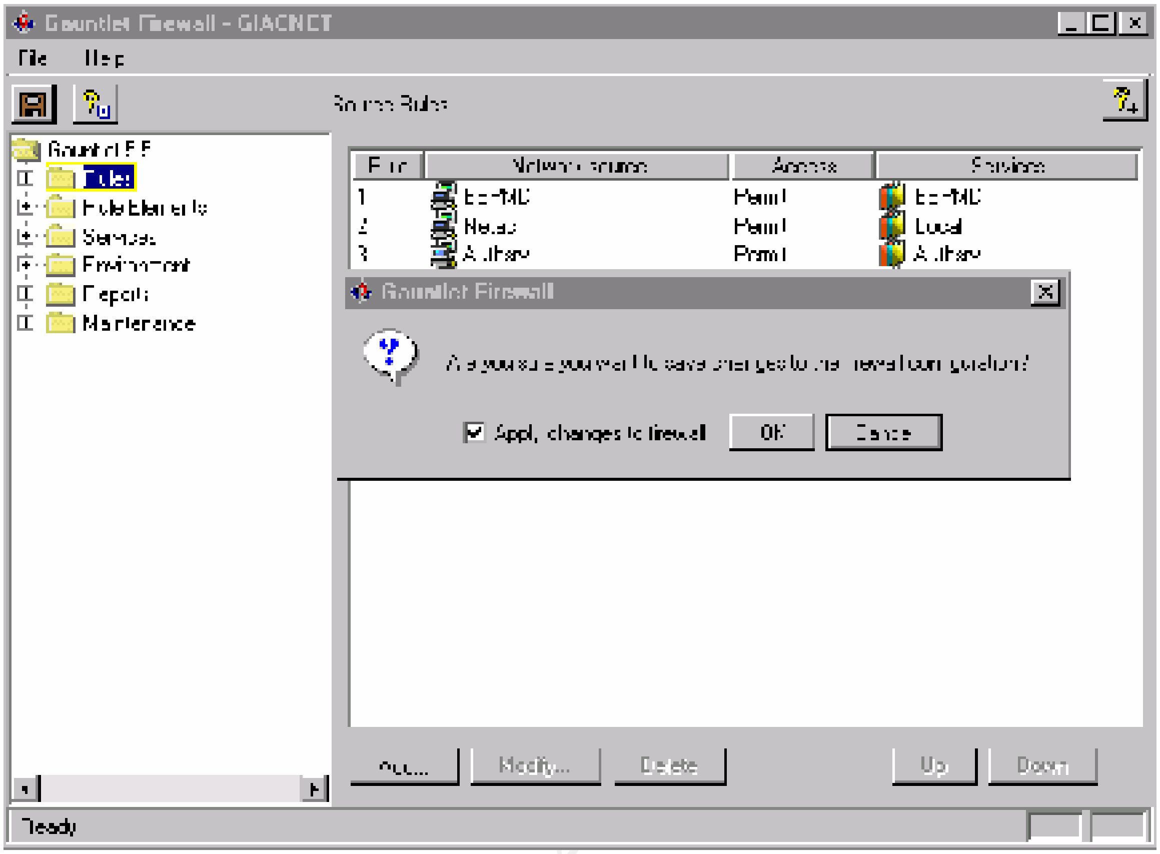
us-agent

□ Use source address of originating host

HTTP Operations

FTP (FTP requests):	□ Deny	☑ Log
Gopher (Gopher requests):	□ Deny	☑ Log
HTTP (HTTP requests):		☑ Log

When you are all done, don't forget to save and apply the changes to the firewall. From the main screen, open the File menu and then click on save. The following screen appears:



Gauntlet Firewall - GIACNET

Source Rules

Prior	Network source	Access	Services
1	Ext-MD	Permit	Ext-MD
2	Netas	Permit	Local
3	Anywhere	Permit	Anywhere

Gauntlet Firewall

Are you sure you want to save changes to the firewall configuration?

☑ Apply changes to firewall

Make sure the Apply changes to firewall is checked and then click on OK.

Appendix B – Fire Proof Tutorial and Setup

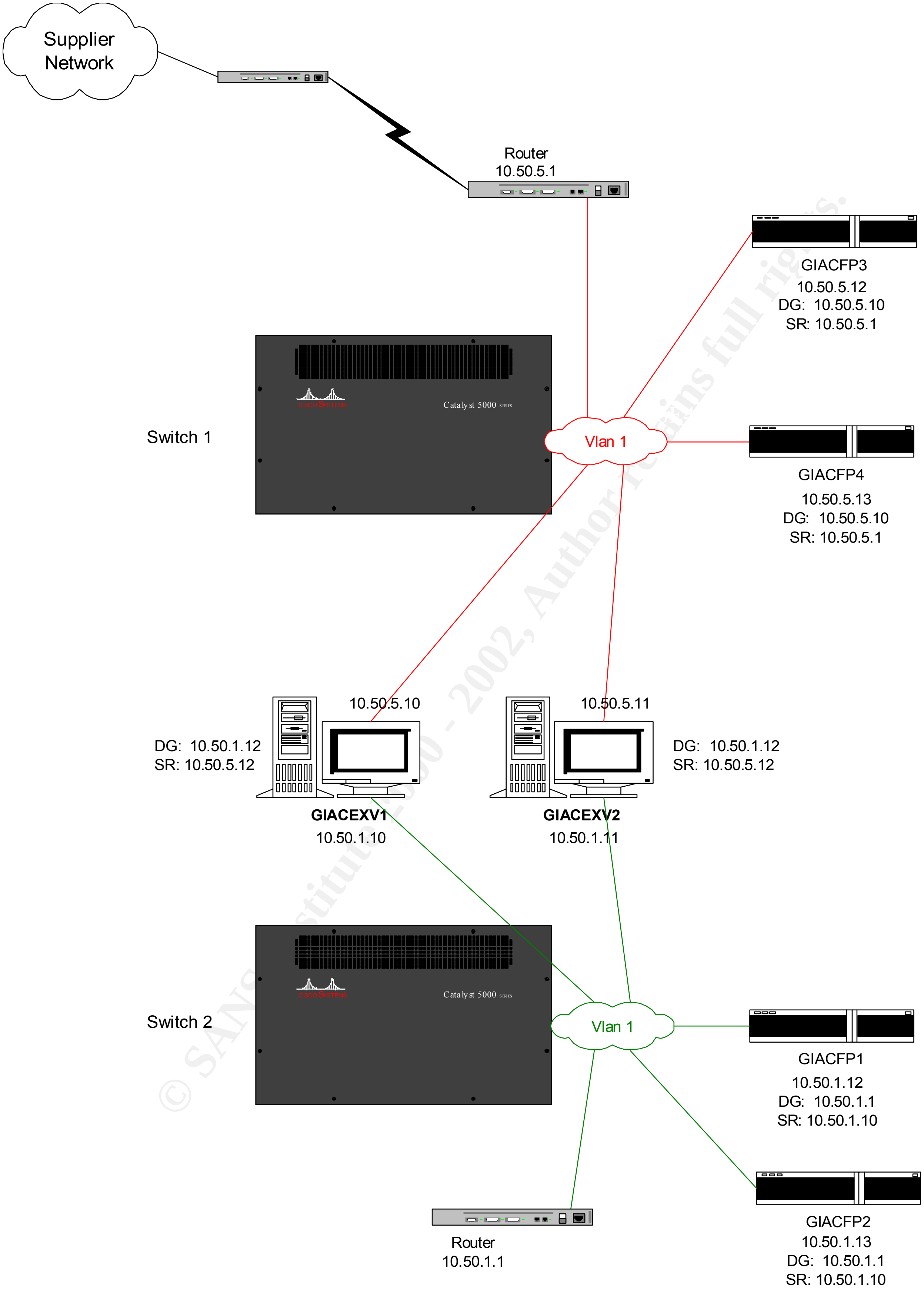
Introduction:

The next few pages explain how to setup Fire Proof boxes from Radware, Inc. to load balance 2 firewalls in a lollipop configuration, as opposed to a 2-port configuration. Lollipop simply means the traffic enters and leaves the Fire Proof on the same port.

A typical installation is one that has 2 firewalls and 2 Fire Proofs on each side of the firewalls. In the diagram on the next page we have the 2 firewalls GIACEXV1 and GIACEXV2. We also have the 2 Fire Proofs GIACFP1 and GIACFP2 on the side of the firewalls facing GIAC network. The other 2 Fire Proofs GIACFP3 and GIACFP4 are on the side of the firewalls facing the supplier network. The Fire Proofs will be setup to load balance the traffic between the 2 firewalls. Both firewalls will take traffic as long as they are up. If one of the firewalls fail, the traffic will be diverted to the other firewall, and the user will notice no interruption.

Note that GIACFP1 and GIACFP2 are configured exactly the same. GIACFP2 is a fail over for GIACFP1. The same applies to GIACFP3 and GIACFP4.

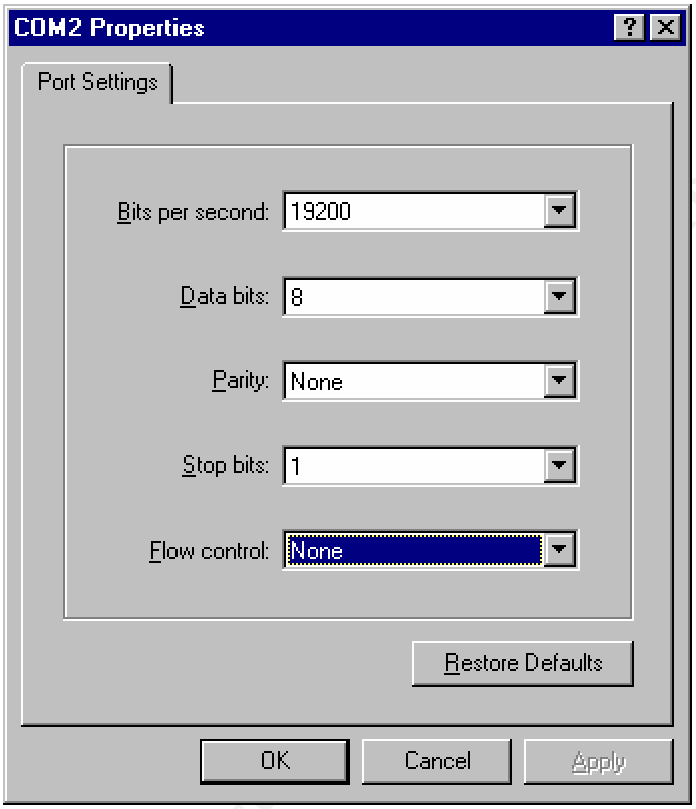
The following pages explain how to setup GIACFP1 and GIACFP2. Applying the same steps to GIACFP3, and GIACFP4 should not be difficult.


Supplier Network
Router
10.50.5.1
GIACFP3
10.50.5.12
DG: 10.50.5.10
SR: 10.50.5.1
Switch 1
Catalyst 5000 series
Vlan 1
GIACFP4
10.50.5.13
DG: 10.50.5.10
SR: 10.50.5.1
10.50.5.10
10.50.5.11
DG: 10.50.1.12
SR: 10.50.5.12
DG: 10.50.1.12
SR: 10.50.5.12
GIACEXV1
10.50.1.10
GIACEXV2
10.50.1.11
Switch 2
Catalyst 5000 series
Vlan 1
GIACFP1
10.50.1.12
DG: 10.50.1.1
SR: 10.50.1.10
Router
10.50.1.1
GIACFP2
10.50.1.13
DG: 10.50.1.1
SR: 10.50.1.10

Setting up the Fire Proof:

1. Connect to the Fire Proof via a serial cable using HyperTerminal. The Fire Proof uses a special cable for the console that attaches to a PC. The cable is straight through with a DB-9 Male connector to attach to the DB-9 female connector.

   The console settings are:



COM2 Properties

Port Settings

Bits per second: 19200

Data bits: 8

Parity: None

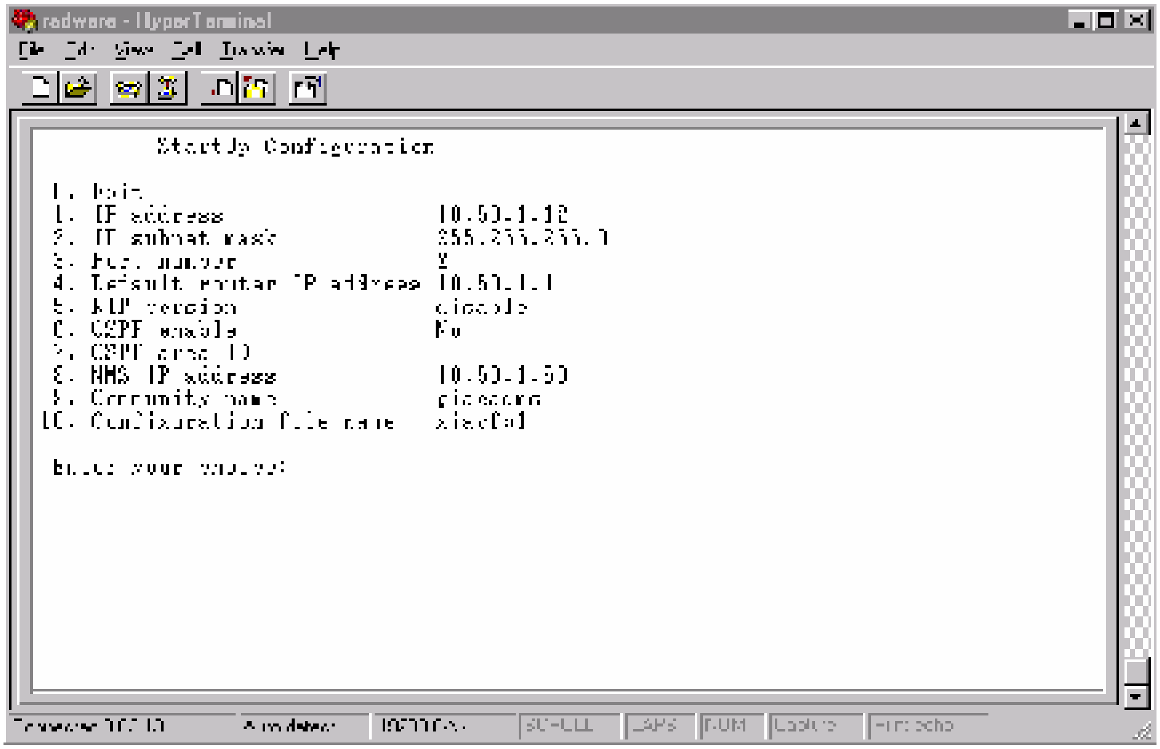
Stop bits: 1

Flow control: None

Click OK to connect. You will see a menu that lets you configure the following parameters:

IP Address	This is the IP address of the Fire Proof device. For example, use 10.50.1.12 for GIACFP1
IP Subnet Mask	255.255.255.0
Port Number	This is the physical port number of the Fire Proof box. Possible values are 1-4 (We are going to use Port 2 for our setup).
Default router IP Address	Should be the firewall for the Fire Proof boxes that are on the untrusted side of the firewall (GIACFP3 and GIACFP4). Should be the internal router for the Fire Proof boxes that are on the trusted side of the firewall (GIACFP1 and GIACFP2).
RIP Version	Disabled
OSPF Enabled	Disabled
OSPF Area ID	Blank
NMS IP Address	Network Management Station's IP Address. This is needed to administer the Fire Proof boxes from one location.
Community Name	Giacadms
Configuration file name	Giacfp1

When done the screen will look like this:



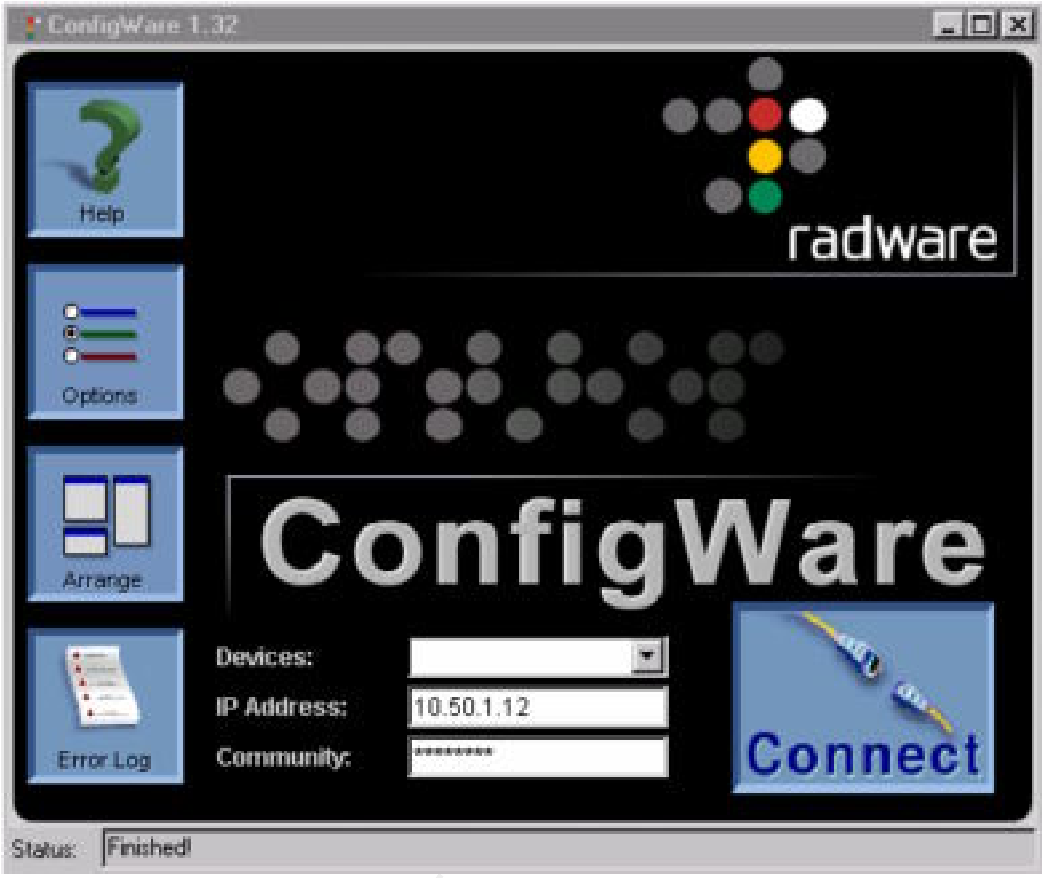
```
          StartUp Configuration

 0. Exit
 1. IP address                  10.50.1.12
 2. IP subnet mask              255.255.255.0
 3. Port number                 2
 4. Default router IP address   10.50.1.1
 5. RIP version                 disable
 6. OSPF enable                 No
 7. OSPF area ID
 8. NMS IP address              10.50.1.50
 9. Community name              password
10. Configuration file name     xiec[0]

 Enter your choice:
```

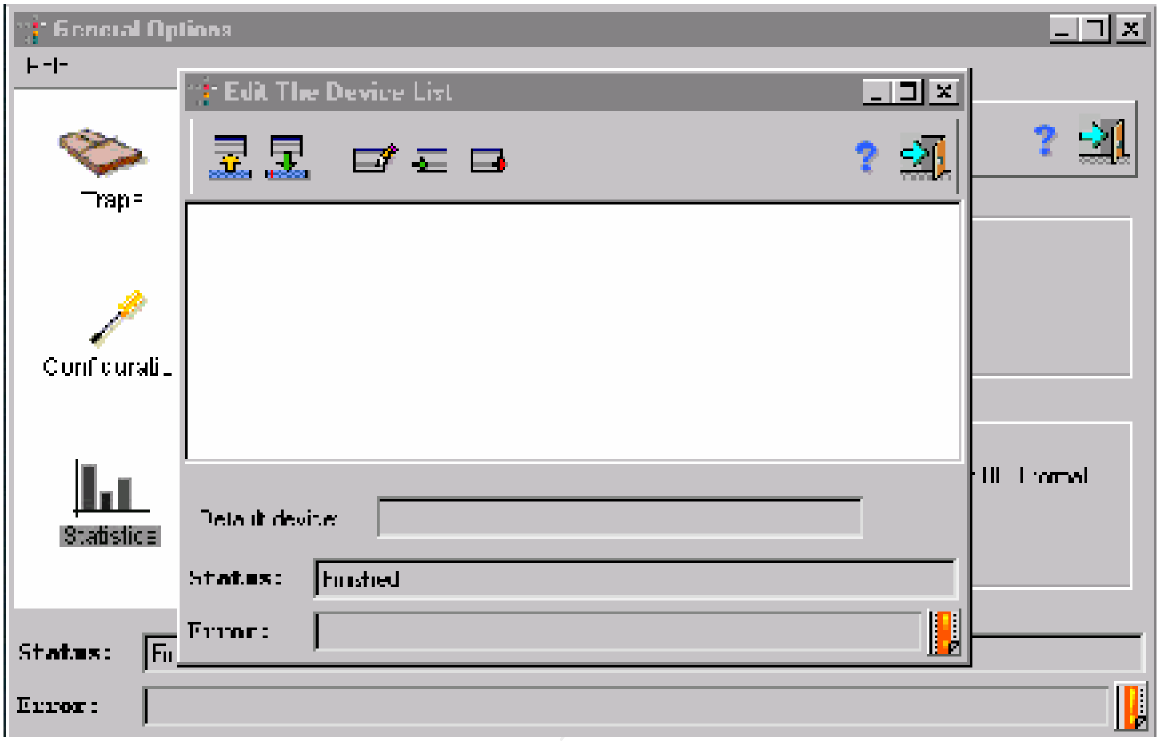
Enter choice 0 and confirm the selections. The Fire Proof will now boot up.

2. At this point it is possible to connect to the device over the network. Invoke ConfigWare (the management software provided by Radware). The screen looks like this:



Devices:
IP Address: 10.50.1.12
Community: *********
Status: Finished!

It is possible to setup multiple Fire Proofs to connect to from this screen. To do so click on Options, then click on Configuration, and then click on "Edit the device list for the initial applet". The following screen appears:



Default device:

Status: Finished

Error:

Status: Fi

Error:

This is a good place to explain some of the symbols used in all Fire Proof screens:

Refresh – Used to read the configurations from the Fire Proof

Set – Used to save the configurations to the Fire Proof

Edit – Used to edit an entry

Insert – Used to create a new entry

Delete – Used to delete an entry

Help – Used to get specific help for a specific screen

Close Screen – Used to exit a screen. It is important to Set before closing a screen if you want to save the information.

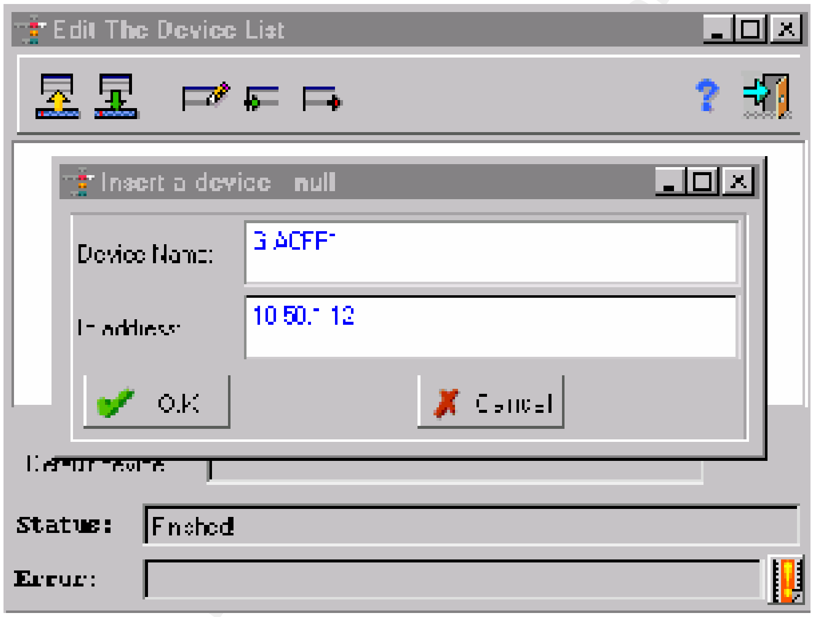
Undo Changes – Used to undo any changes made

Print

Cancel – Used to exit a screen without saving

✓ Update – Used to exist a screen with save

To create an entry for a Fire Proof device, from the above "Edit The Device List" screen click on Insert. The following screen appears:



Device Name: 3 ACFF

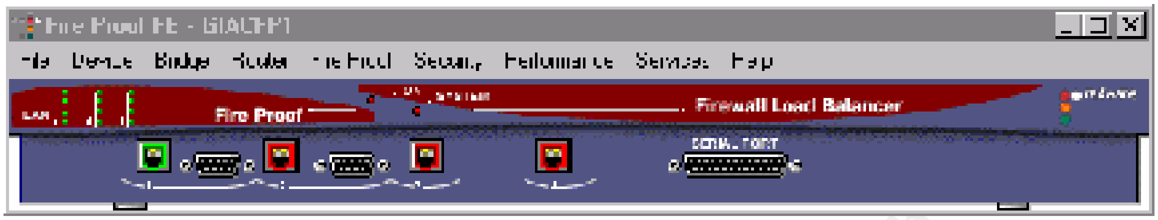
IP address: 10 50. 12

Status: Finished

Error:

Enter a Device Name and Ip address and then click OK. You can repeat this process for each Fire Proof device that needs to be managed. When done click on the Set button to save the new configurations. The devices will now appear in the drop down list on the main ConfigWare screen.

3. Enter the device IP address in the IP Address field on the main ConfigWare screen or choose from the drop down list. Enter giacadms in the Community field and click on Connect. The following screen appears:

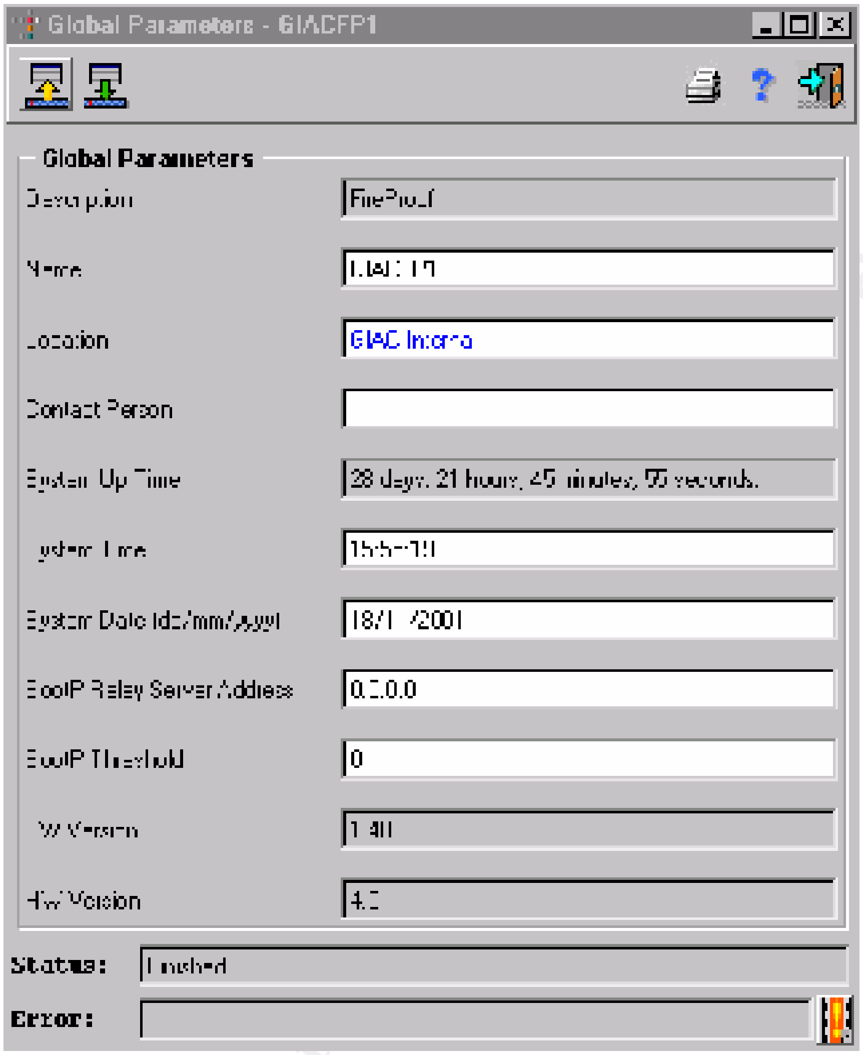


This is the main Fire Proof screen. All configurations can be done from this screen.

4. From the main Fire Proof screen go to Device → Global Parameters and configure the following parameters:

Name	Same as configuration file name.
Location	Specify the appropriate location
Contact Person	Enter the name of the admin
System Time/System Date	Verify that these are correct

When done, the screen will look like this:



Global Parameters - GIACFP1

Global Parameters

Description	FireProof
Name	GIACFP1
Location	GIAC Interna
Contact Person	
System Up Time	28 days, 21 hours, 45 minutes, 55 seconds.
System Time	15:54:19
System Date (dd/mm/yyyy)	18/1 /2001
BootP Relay Server Address	0.0.0.0
BootP Threshold	0
SW Version	1.40
HW Version	4.0

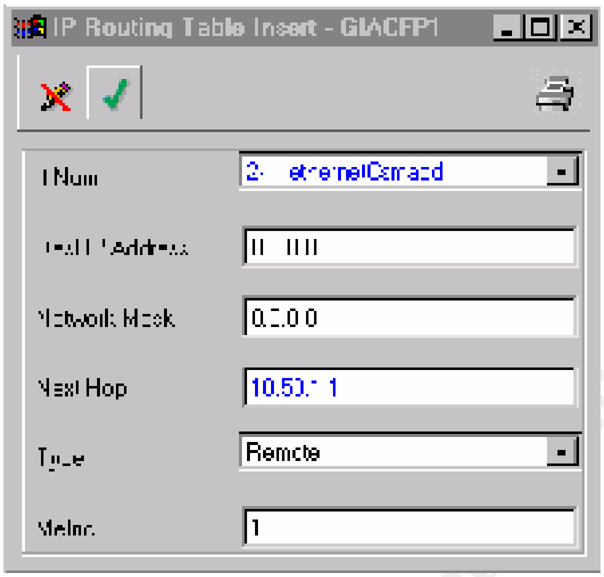
Status: Finished

Error:

Click on the Set button and then click on the Close Screen button

5. From the main Fire Proof screen go to Router → Routing Table and set your default route. Also, any needed static routes are entered here. Assuming that 10.129.101.0 is the Supplier's NATed network, the routes needed for GIACFP1 are as follows:

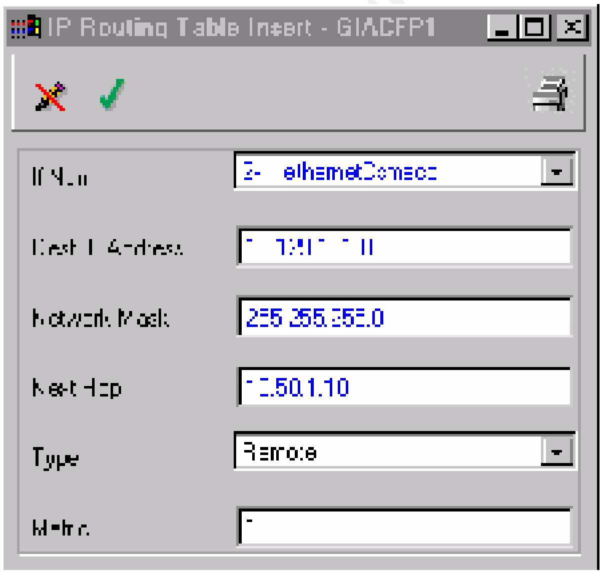
Default Route:



IP Routing Table Insert - GIACFP1

IfNum	2 - ethernetCsmacd
Dest IP Address	0.0.0.0
Network Mask	0.0.0.0
Next Hop	10.50.1.1
Type	Remote
Metric	1

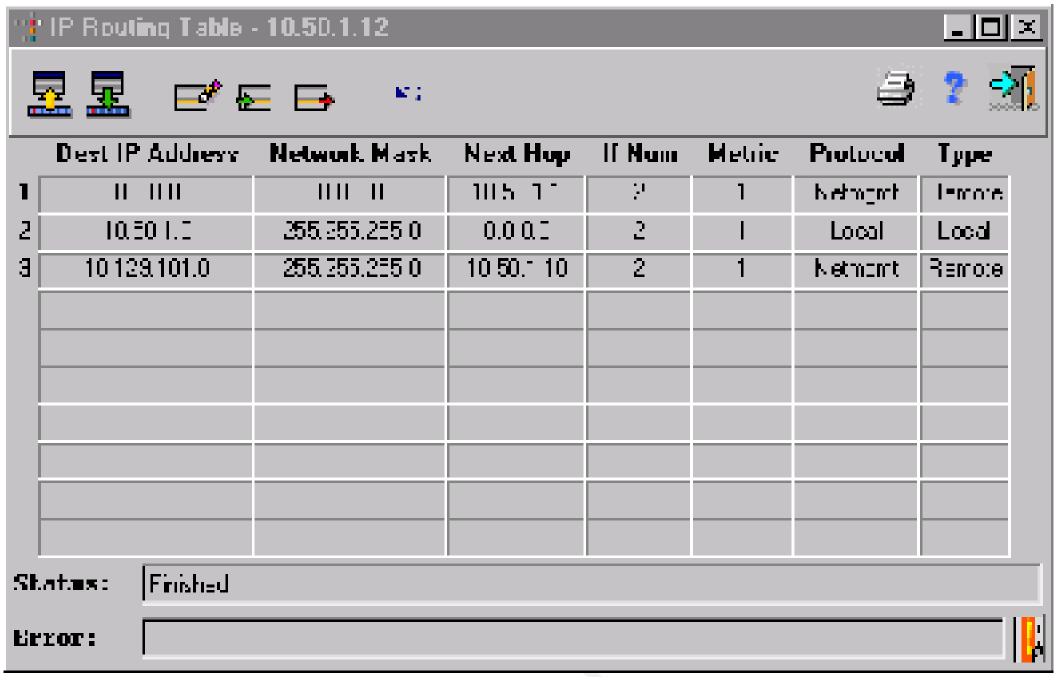
Static Route:



IP Routing Table Insert - GIACFP1

IfNum	2 - ethernetCsmacd
Dest IP Address	[illegible]
Network Mask	255.255.255.0
Next Hop	10.50.1.10
Type	Remote
Metric	1

When done the screen will look like this:



IP Routing Table - 10.50.1.12

	Dest IP Address	Network Mask	Next Hop	IF Num	Metric	Protocol	Type
1	[illegible]	[illegible]	[illegible]	2	1	Netmgmt	Remote
2	10.50.1.2	255.255.255.0	0.0.0.0	2	1	Local	Local
3	10.129.101.0	255.255.255.0	10.50.1.10	2	1	Netmgmt	Remote

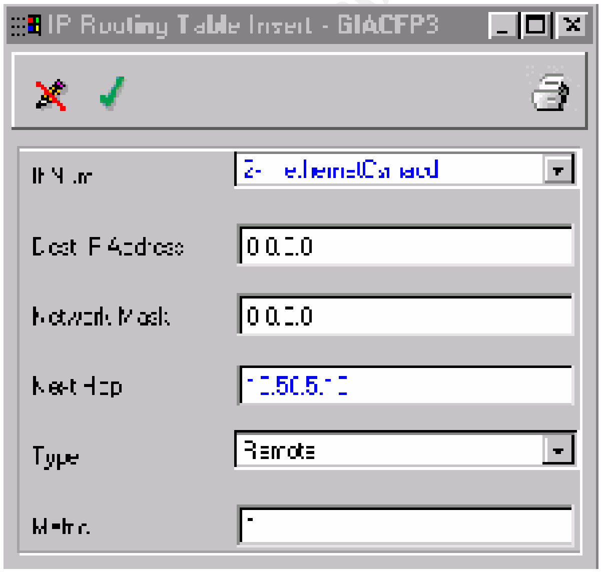
Status: Finished

Error:

Click on the Set button and then click on the Close Screen button

The routes needed for GIACFP3 are as follows:

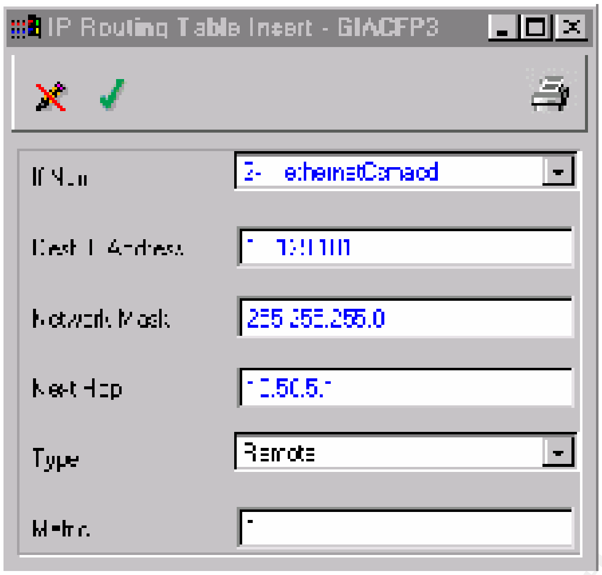
Default Route:



IP Routing Table Insert - GIACFP3

IF Num	2 - ethernetCsmacd
Dest IP Address	0.0.0.0
Network Mask	0.0.0.0
Next Hop	10.50.5.1
Type	Remote
Metric	1

Static Route:



IP Routing Table Insert - GIACFP3

IF Num: 2- ethernetCsmacd

Dest IP Address: 10.129.101.0

Network Mask: 255.255.255.0

Next Hop: 10.50.5.12

Type: Remote

Metric: 1

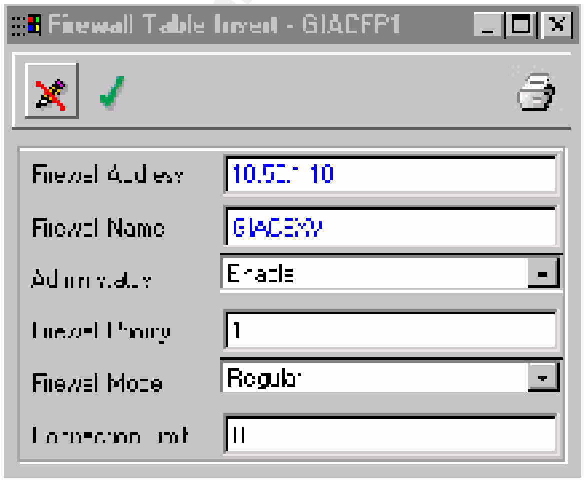
To complete the picture, here are the routes needed on the firewalls:

Destination	Gateway	Flags	Refs	Use	Interface	Pmtu
Default	10.50.1.12	UG	7	41887680	lan1	1500
10.129.101.0	10.50.5.12	UG	0	772798	lan0	1500

6. From the main Fire Proof screen go to Fire Proof → Global Configuration and configure the following parameters:

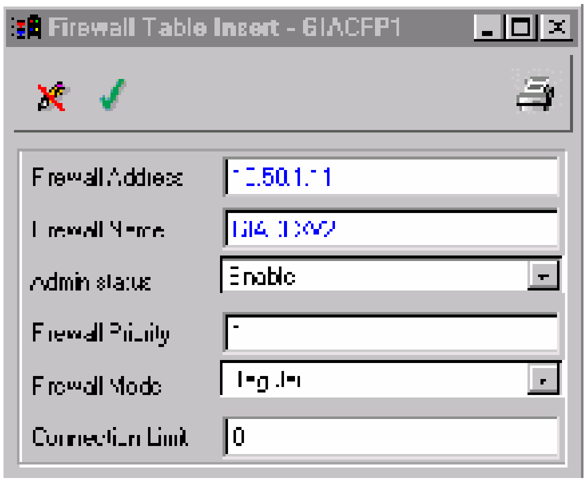
Filed Name	Suggested Value	Possible Values
Admin Status	Enable	Enable/Disable
Dispatch Method	Least amount of traffic	Cyclic Fewest Number of Users NT-1 NT-2 Private-1 Private-2 Fewest Bytes Number
Check Connectivity Status	Enable	Enable/Disable
Check Connectivity Method	Ping	
Polling Interval	10 Seconds	
Number of Retries	5	
Client Aging Time	10800 Seconds	
Session Tracking	Enable	Enable/Disable
Client Mode	Layer 3	Laery3/Layer4
Translate Outbound Traffic to Virtual Address	Disabled (in most configurations)	Enable/Disable
Remove Entry At Session End	Enable	Enable/Disable

7. From the main Fire Proof screen go to Fire Proof → Firewall Table and enter the firewalls that you will be load balancing with this device. The values for GIACFP1 are:



Firewall Table Insert - GIACFP1

Firewall Address	10.5[illegible] 10
Firewall Name	GIACFW
Admin Status	Enable
Firewall Priority	1
Firewall Mode	Regular
Connection Limit	0



Firewall Table Insert - GIACFP1

Firewall Address: 10.50.1.11

Firewall Name: GIACFW2

Admin status: Enable

Firewall Priority: 1

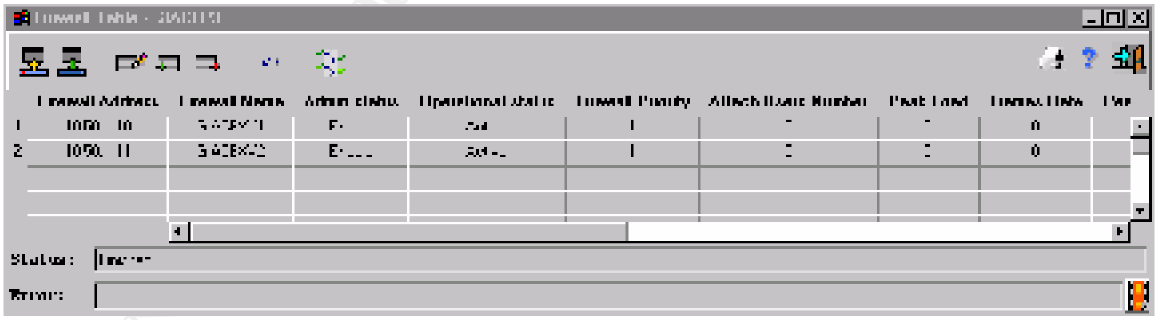
Firewall Mode: Regular

Connection Limit: 0

Possible values for Admin status are enable, disable, or shutdown. Enable means the Fire Proof is to send traffic to this firewall. Disable means the Fire Proof will not send traffic to the firewall. Shutdown means the Fire Proof is to finish the current connections and to stop sending new connections.

Possible values for Firewall Mode are regular or backup. Regular means the traffic is load-balanced between the firewalls. Backup means the firewall will take traffic only when the main firewall is down.

When done the screen will look like this:



Firewall Table - GIACFP1

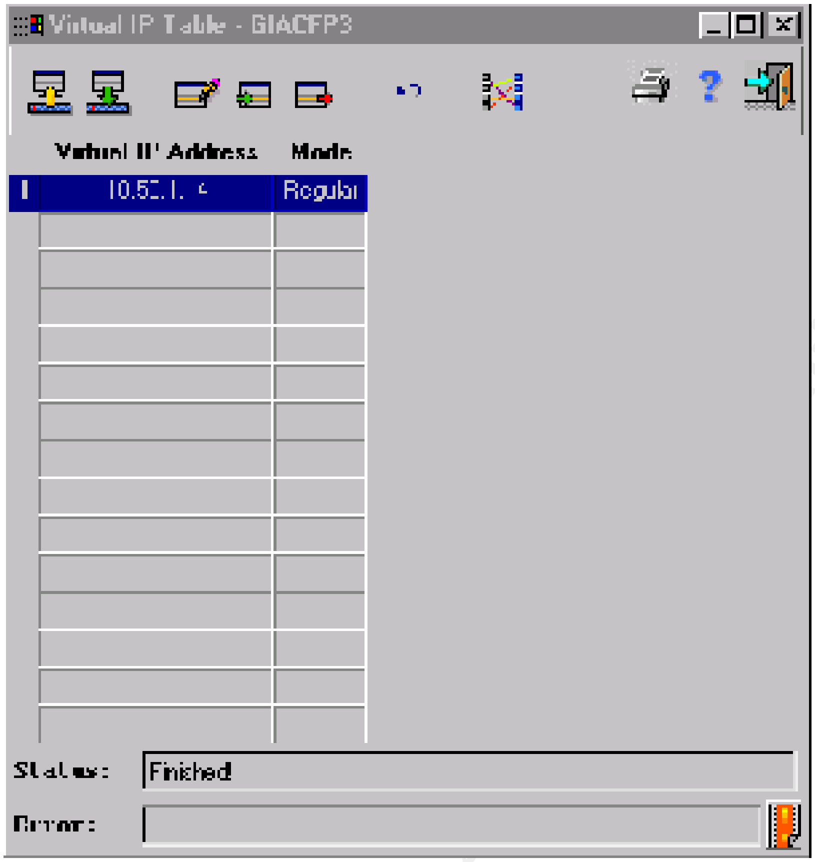
	Firewall Address	Firewall Name	Admin status	Operational Status	Firewall Priority	Attach Users Number	Peak Load	[illegible]	[illegible]
1	10.50.1.10	GIACFW1	Enable	Active	1	[illegible]	[illegible]	0	
2	10.50.1.11	GIACFW2	Enable	Active	1	[illegible]	[illegible]	0	

Status: [illegible]

Errors:

Click on the Set button and then click on the Close Screen button

8. From the main Fire Proof screen go to Fire Proof → Virtual IP and create a VIP (use Regular Mode for a primary Fire Proof box and Backup mode for a backup Fire Proof box). A VIP is used for load balancing "proxied" traffic. Users (through a browser or applications) can point at the VIP to proxy off of it. If all traffic is being routed to the Fire Proofs, then a VIP is not needed. In our case the VIP is 10.50.1.14. Set the changes and when done the screen will look like this:

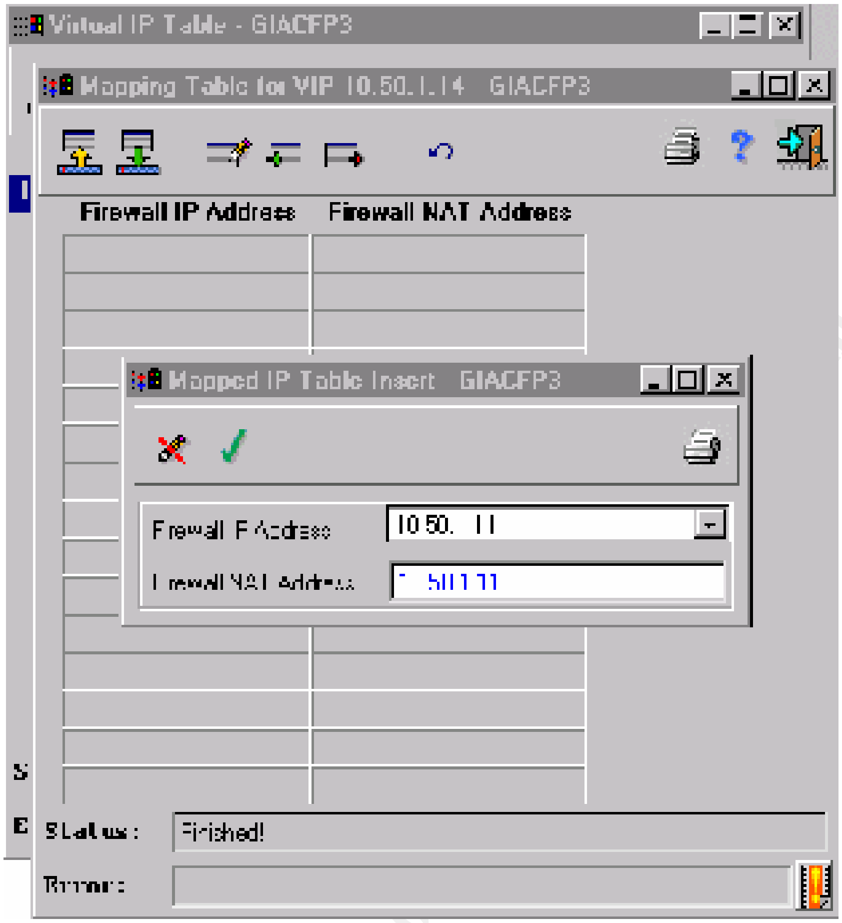


	Virtual IP Address	Mode
1	10.52.1.6	Regular

Status: Finished

Error:

9. Click on the VIP and then click on mapping . When the "Mapping Table…" screen appears, click on the Insert button . The screen will look like this:


Virtual IP Table - GIACFP3

Mapping Table for VIP 10.50.1.14 GIACFP3

Firewall IP Address	Firewall NAT Address

Mapped IP Table Insert GIACFP3

Firewall IP Address: 10.50. 11

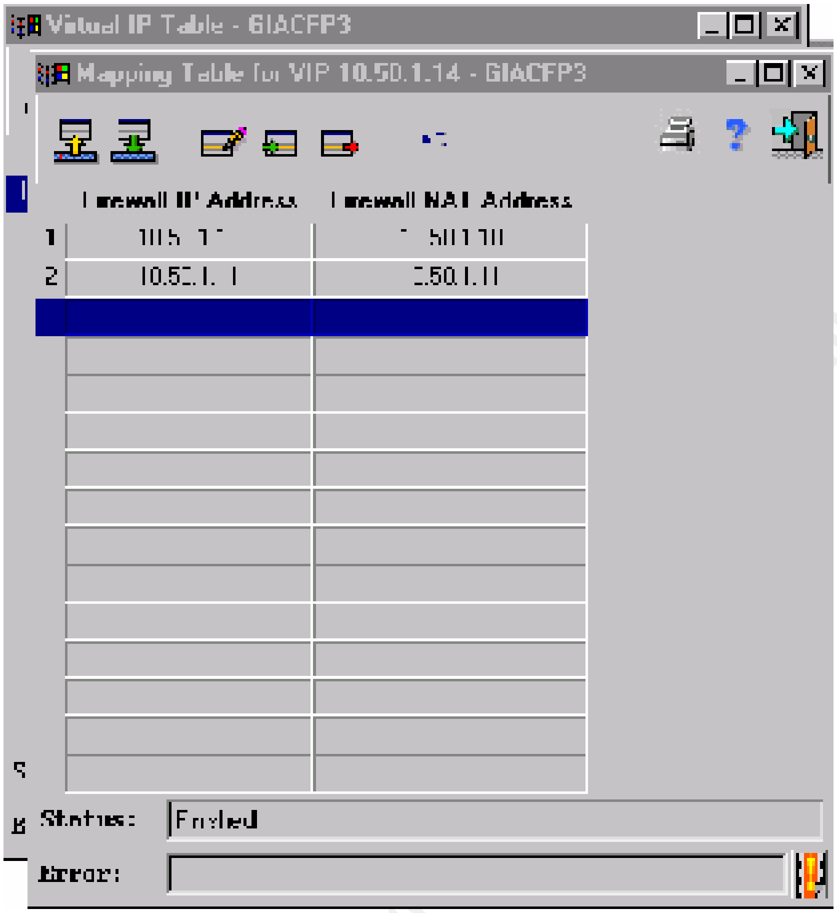
Firewall NAT Address: 50.1.11

Status: Finished!

Error:

Add each firewall’s IP address. The NAT address should be identical to that of the firewall’s IP address. Note that the firewall address will be already listed in the drop down box:

When done the screen will look like this:



Virtual IP Table - GIACFP3

Mapping Table for VIP 10.50.1.14 - GIACFP3

	Firewall IP Address	Firewall NAT Address
1	[illegible]	[illegible]
2	10.5[illegible].1.[illegible]	[illegible].50.1.11

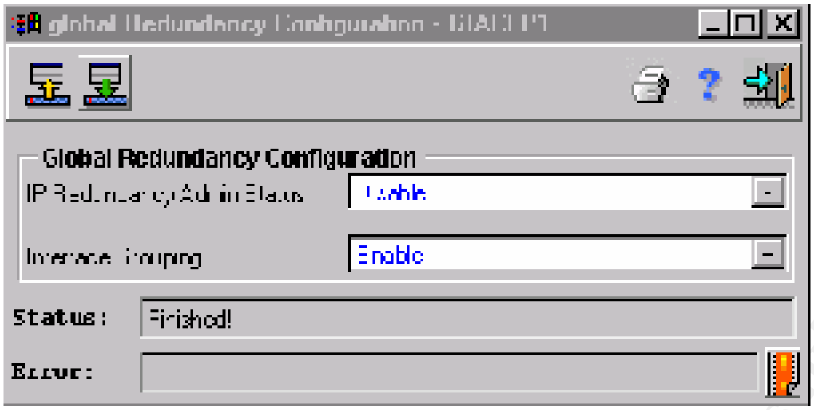
Status: Finished

Error:

Click on the Set button and then click on the Close Screen button

10. It is possible to configure more than one Fire Proof device on a network such that one (the backup device) will back up another (the main device). In this case, a failure of any network interface on the main Fire Proof will fail the whole device, and the backup device, previously idle, will take over all activity. In GIAC's case we have GIACFP2 as the backup for GIACFP1 and GIACFP4 as the backup for GIACFP3.

    To do so, from the main Fire Proof screen go to Fire Proof → Redundancy → Global Configuration. Use the following parameters for a primary Fire Proof box:



global Redundancy Configuration - GIAC FP1

Global Redundancy Configuration

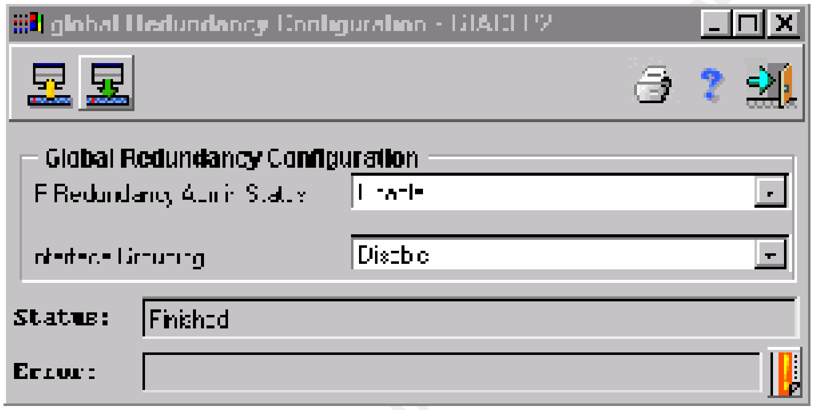
IP Redundancy Admin Status: Disable

Interface Grouping: Enable

Status: Finished!

Error:

Use the EXACT opposite for the backup Fire Proof box:



global Redundancy Configuration - GIAC FP2

Global Redundancy Configuration

IP Redundancy Admin Status: Enable

Interface Grouping: Disable

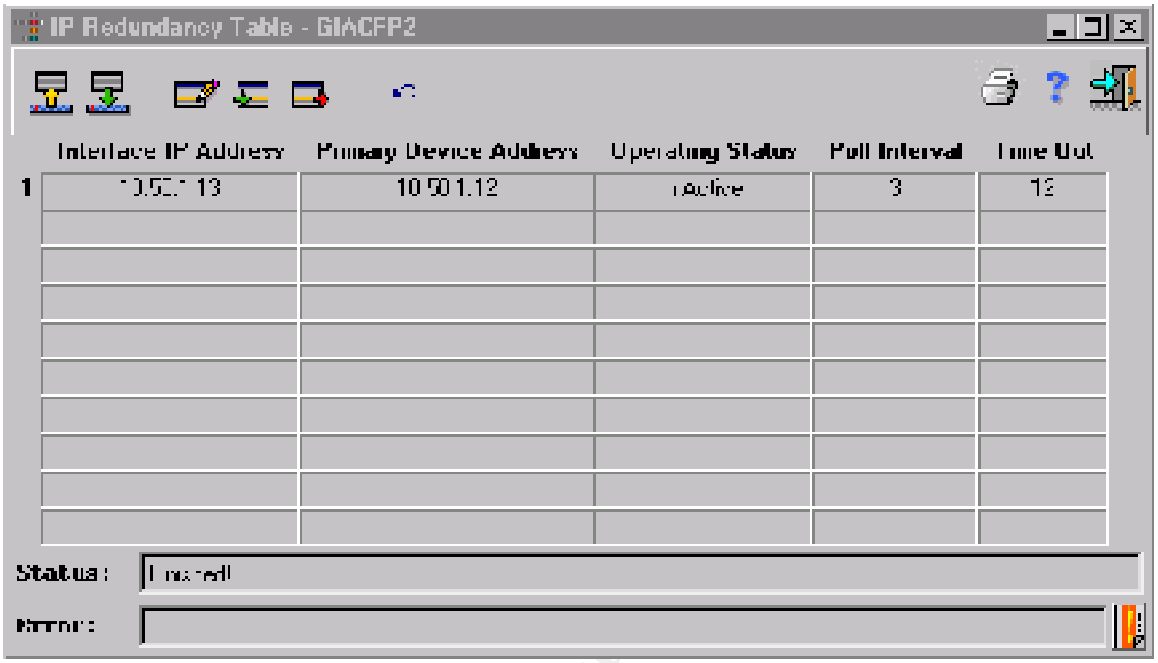
Status: Finished

Error:

IP Redundancy Admin Status – Enabling this feature allows this device to back up another device.

Interface Grouping – Enabling this feature allows this device to be backed up by another device.

11. On the backup Fire Proof box, from the main Fire Proof screen go to Fire Proof → Redundancy → IP Redundancy Table and associate the backup box interface's IP address with that of the primary box. This is done so the backup box will take over whenever the main box fails:



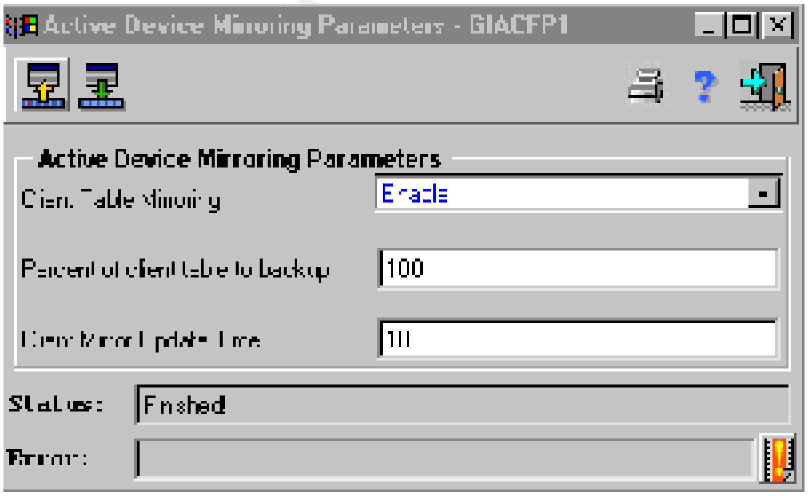
	Interface IP Address	Primary Device Address	Operating Status	Poll Interval	Time Out
1	10.50.1.13	10.50.1.12	nActive	3	12

Status: Finished

Error:

Click on the Set button and then click on the Close Screen button

12. For the primary Fire Proof box, from the main Fire Proof screen go to Fire Proof → Redundancy → Mirroring → Active Device Mirroring Parameters and enable Client Table Mirroring. The other defaults are fine. Mirroring enables a backup redundant device to mirror an active device by sending a snapshot of the information contained on the active device to the backup device. If the active device fails, the backup device can seamlessly resume the sessions.



Active Device Mirroring Parameters

Client Table Mirroring: Enable

Percent of client table to backup: 100

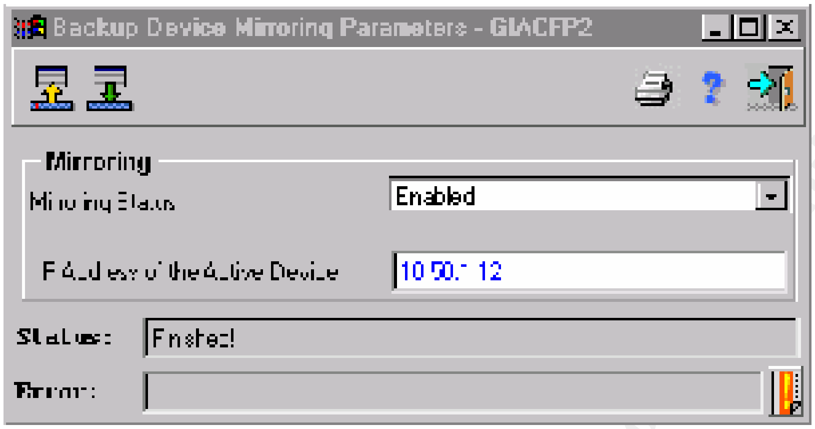
Client Mirror Update Time: 10

Status: Finished

Error:

Click on the Set button and then click on the Close Screen button

13. For the backup Fire Proof box, from the main Fire Proof screen go to Fire Proof → Redundancy → Mirroring → Backup Device Mirroring Parameters and enable Mirroring. Put the ip address of the primary Fire Proof box (not the VIP!!!) as shown below:



Mirroring

Mirroring Status: Enabled

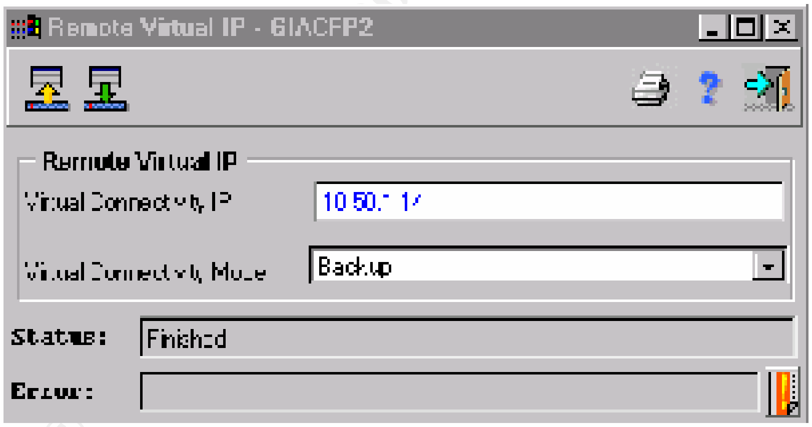
IP Address of the Active Device: 10 50.- 12

Status: Finished

Error:

Click on the Set button and then click on the Close Screen button

14. From the main Fire Proof screen go to Fire Proof → Remote Virtual IP and put the VIP in the Virtual Connectivity IP field. If the Fire Proof box is a primary, use a connectivity mode of Regular. If it's a backup, use the backup mode as shown below:



Remote Virtual IP

Virtual Connectivity IP: 10 50.- 1/

Virtual Connectivity Mode: Backup

Status: Finished

Error:

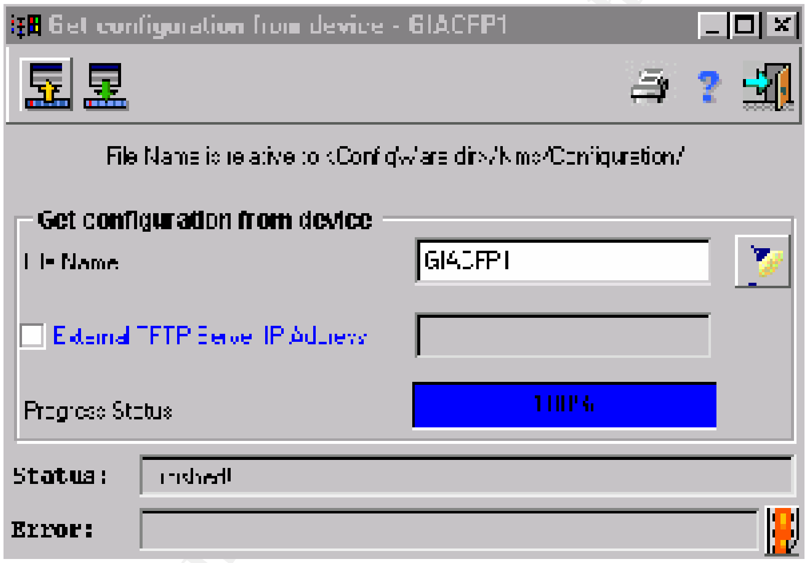
Click on the Set button and then click on the Close Screen button

15. For the Fireproof boxes that reside outside of the firewalls, make sure that 2 packet filters are put in the firewalls for each box:

    - To allow the management station to communicate over port 161 (UDP – SNMP) to the Fire Proof boxes.

    - To allow the Fire Proof boxes to communicate over port 162 (UDP – SNMP) to the management station.

16. To backup the configuration, from the main Fire Proof screen go to File → Configuration File → Receive From Device. Enter the name of the Fire Proof box as the file name. Check the External TFTP Server IP Address and provide a TFTP server IP address or leave it unchecked if you want it backed up to the management station. Click the Set button when ready:



Get configuration from device - GIACFP1

File Name is relative to <Configware dir>/Nms/Configuration/

Get configuration from device

File Name: GIACFP1

☐ External TFTP Server IP Address

Progress Status: 100%

Status: Finished!

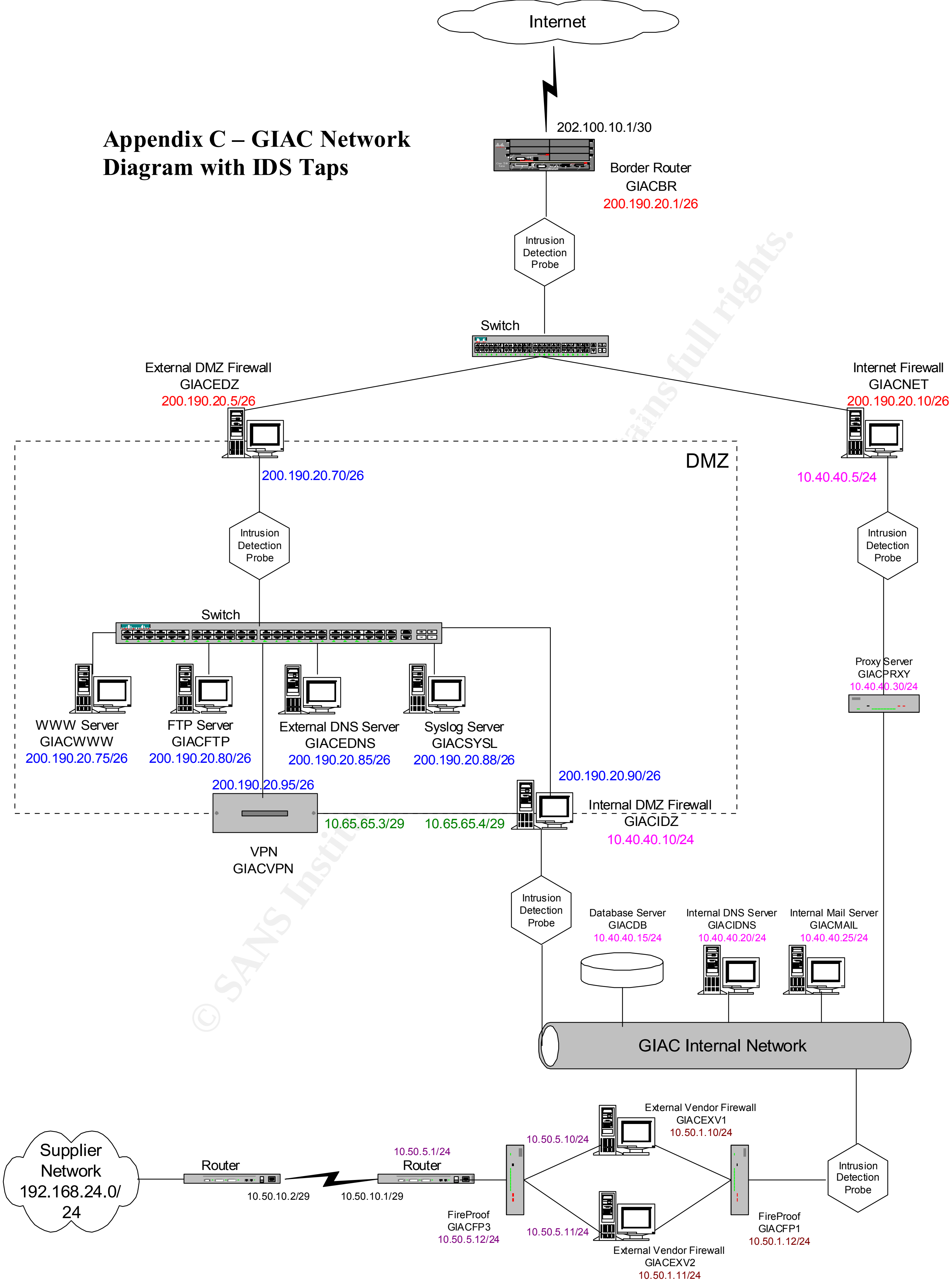
Error:

17. A good trouble-shooting tool is the client table. From the main Fire Proof screen go to Fire Proof → Clients → Client Table. This gives you a snapshot off all the clients' connections and which firewall they are going through.

18. Exit the Fire Proof by File → Exit.

19. Exit the ConfigWare main screen by clicking on the X button at the top right hand corner.

Appendix C – GIAC Network Diagram with IDS Taps



References:

Books and Manuals:

GCFW Course Materials, The SANS Institute.

Network Associates Inc, Gauntlet Firewall for UNIX Administrator's Guide version 5.5. Santa Clara: Network Associates Inc., 1996-1999.

Network Associates Inc, Gauntlet Firewall for UNIX User's Guide version 5.5. Santa Clara: Network Associates Inc., 1996-1999.

Network Associates Inc, Gauntlet Firewall for UNIX Configuration Guide for Fire Proof. Santa Clara: Network Associates Inc., 1999

Nortel Networks, Reference for the Contivity VPN Switch. Billerica, MA: Nortel Networks, 2000.

CheckPoint Inc, Firewall-1 Manuals version 4.1

Scambray, McClure & Kurtz . Hacking Exposed, 2nd edition

Online:

http://www.iss.net/securing_e-business/security_products/intrusion_detection/realsecure_manager/index.php

http://www.finisar-systems.com/htdocssh/products/taps/index.html

Cisco Systems Inc. "Improving Security on Cisco Routers" URL: http://www.cisco.com/warp/public/707/21.html

"Which Ports Does FireWall-1 Use?" URL: http://www.phoneboy.com/faq/0105.html

http://www.insecure.org/nmap

http://www.securiteam.com

Dietrich, Sven. Long, Neil. Dittrich, David. "An analysis of the Shaft distributed denial of service tool" Copyright 2000. March 13, 2000 URL: http://www.adelphi.edu/~spock/shaft_analysis.txt

Kessler, Gary C. "Defenses Against Distributed Denial of Service Attacks". November 29, 2000. URL: http://www.sans.org/infosecFAQ/threats/DDoS.htm

"Results of the Distributed-Systems Intruder Tools Workshop" November 2-4, 1999. Published at the CERT Coordination Center. URL: http://www.cert.org/reports/dsit_workshop.pdf

"CERT Advisory CA-1999-17 Denial-of-Service Tools" December 28, 1999. Last Updated: March 3, 2000. URL: http://www.cert.org/advisories/CA-1999-17.html