



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

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



GIAC Certified Firewall Analyst

GCFW Practical Assignment

Version 2.0

James D. Rauser

SANS Washington, DC

Dec 7-13, 2003

Submitted May 2004

© SANS Institute 2004, Author retains full rights

# Table of Contents

Abstract .....	4
Assignment 1 Security Architecture .....	4
1 Business Operations.....	4
1.1 Business Model.....	4
1.2 Architecture .....	4
1.3 Cost Considerations .....	5
2 Access Requirements .....	7
2.1 Customers.....	7
2.2 Partners and Suppliers.....	8
2.3 Internal Employees .....	8
2.4 Mobile Sales Force and TeleWorkers .....	11
2.5 General Public.....	12
3 Network Design .....	12
3.1 Border Router.....	12
3.2 Firewall.....	13
3.3 VPN.....	14
3.4 IP Addressing Scheme.....	14
3.5 Optional Components .....	15
3.6 Component Descriptions .....	17
Assignment 2 Security Policy and Tutorial .....	18
1 Border Router.....	18
1.1 Component Description and Security Role .....	18
1.2 Center for Internet Security .....	18
1.3 Network Diagram and Border Router Configuration.....	21
1.4 Router Auditing Tool Results.....	31
2 Primary Firewall .....	32
2.1 Component Description and Security Role .....	32
2.2 Firewall Security Settings .....	33
2.3 Access Requirements and Primary Firewall Configuration .....	36
2.3 Secondary Firewall Configuration .....	38
3 VPN's .....	40
3.1 Component Description.....	40
3.2 Border Router VPN .....	40
3.3 Mobile and TeleWorkers VPN.....	42
3.4 Partners and Suppliers VPN .....	44
3.5 Slave Database VPN .....	45
4 CheckPoint FW-1 NG to Cisco 3640 VPN Tutorial.....	46
4.1 Introduction .....	46
4.2 Router Configuration .....	47

4.3 Firewall Configuration .....	50
<b>Assignment 3 Verify the Firewall Policy .....</b>	<b>55</b>
<b>1 Validation Plan.....</b>	<b>55</b>
1.1 Technical Approach .....	55
1.2 Timing Considerations .....	56
1.3 Estimate Costs and Effort.....	56
1.4 Identify Risks.....	57
<b>2 Validation.....</b>	<b>57</b>
2.1 How the Validation was Accomplished.....	57
2.2 Tools and Commands .....	57
<b>3 Results.....</b>	<b>84</b>
3.1 Analysis.....	84
3.2 Recommendations for Improvement .....	85
<b>Assignment 4 Design Under Fire .....</b>	<b>86</b>
<b>1 Select a Network.....</b>	<b>86</b>
<b>2 Firewall Attack.....</b>	<b>87</b>
2.1 Vulnerability.....	87
2.2 Attack .....	88
2.3 Results .....	88
2.4 Countermeasures.....	88
<b>3 Distributed Denial of Service Attack.....</b>	<b>88</b>
3.1 Compromise the Zombies .....	88
3.2 DDoS Design .....	89
3.3 Attack .....	89
3.4 Results .....	90
3.5 Countermeasures.....	90
<b>4 Internal System Compromise.....</b>	<b>91</b>
4.1 Target.....	91
4.2 Attack .....	91
4.3 Results .....	93
4.4 Countermeasures.....	93
<b>Appendix .....</b>	<b>95</b>
IP Addressing.....	95
Firewall Policy .....	97
Component Descriptions .....	99
Wap Configuration .....	102
EAP-TTLS RADIUS Server Configuration .....	106
NMAP Scans .....	108
<b>References .....</b>	<b>116</b>

## Abstract

This paper is part of the GIAC certification process for the GIAC Certified Firewall Analyst. As part of the certification process we are tasked with a practical assignment. The practical assignment consists of four parts. The first three revolve around designing, the security architecture for a fictional company that has on-line sales of fortune cookie sayings. Once the security architecture is designed, the next part is to provide a security policy and tutorial for the major components. The third part is to verify the firewall policy. Finally the fourth piece is to evaluate the security architecture of, a previous student's practical.

## Assignment 1 Security Architecture

### 1 Business Operations

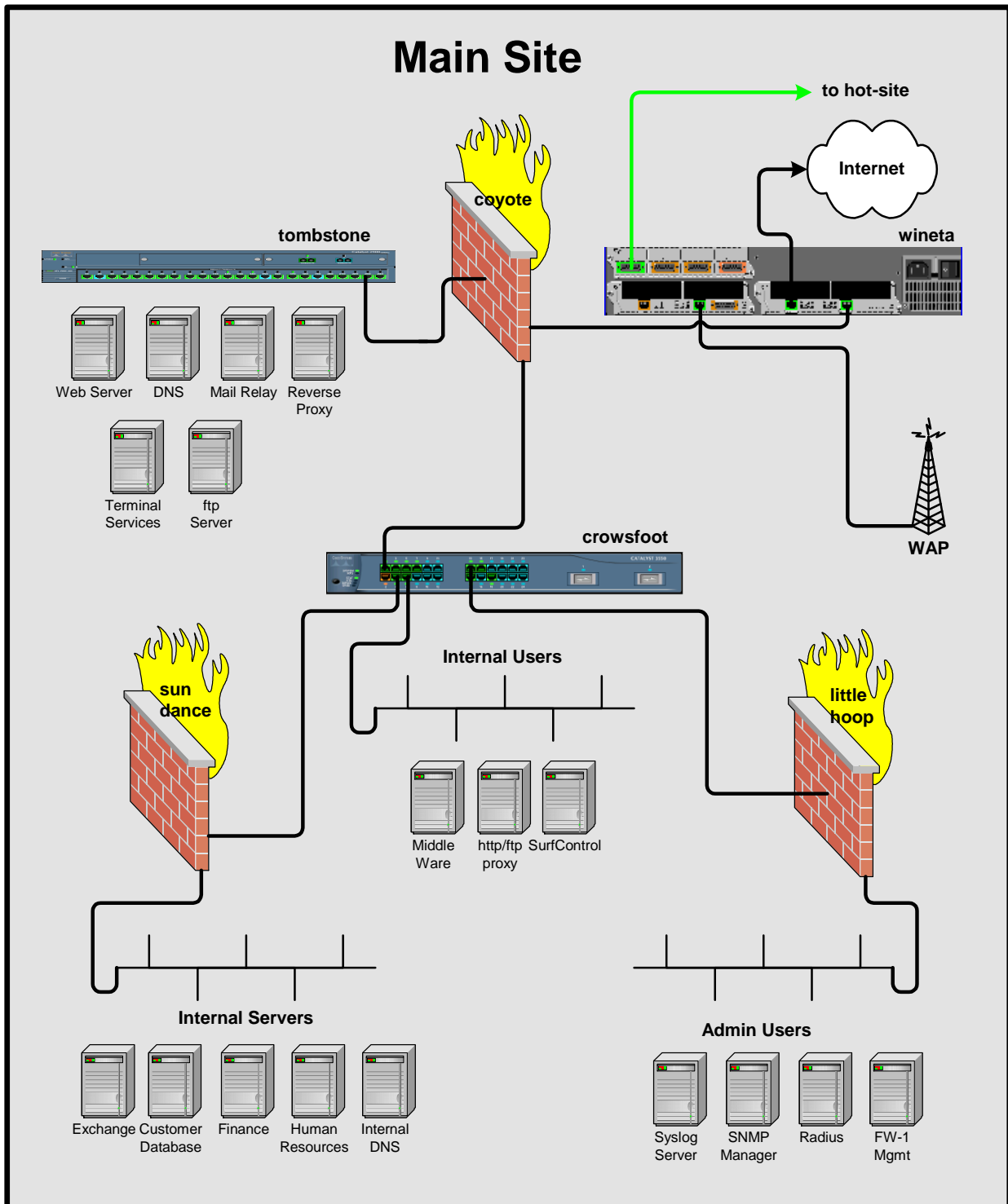
#### 1.1 Business Model

GIAC Enterprises deals in the on-line sale, of fortune cookie sayings. Its business model is that of a virtual merchant in that it operates solely over the web. All revenue is generated from on-line transactions. Its mission is to be the best in class in the quality of its product.

#### 1.2 Architecture

The company consists of several business units: IT, Security, Management, Human Resources, Finance and Sales. Its headquarters is in San Francisco, CA. It also has several partner and additional small offices located in New York, Chicago and Houston. One of the small office locations (Chicago) is also a hot site for disaster recovery and also provides redundant Internet connectivity. All data is mirrored to the hot-site. The amount of data to be transferred will depend on the amount of fortunes and customer data available at any given moment.

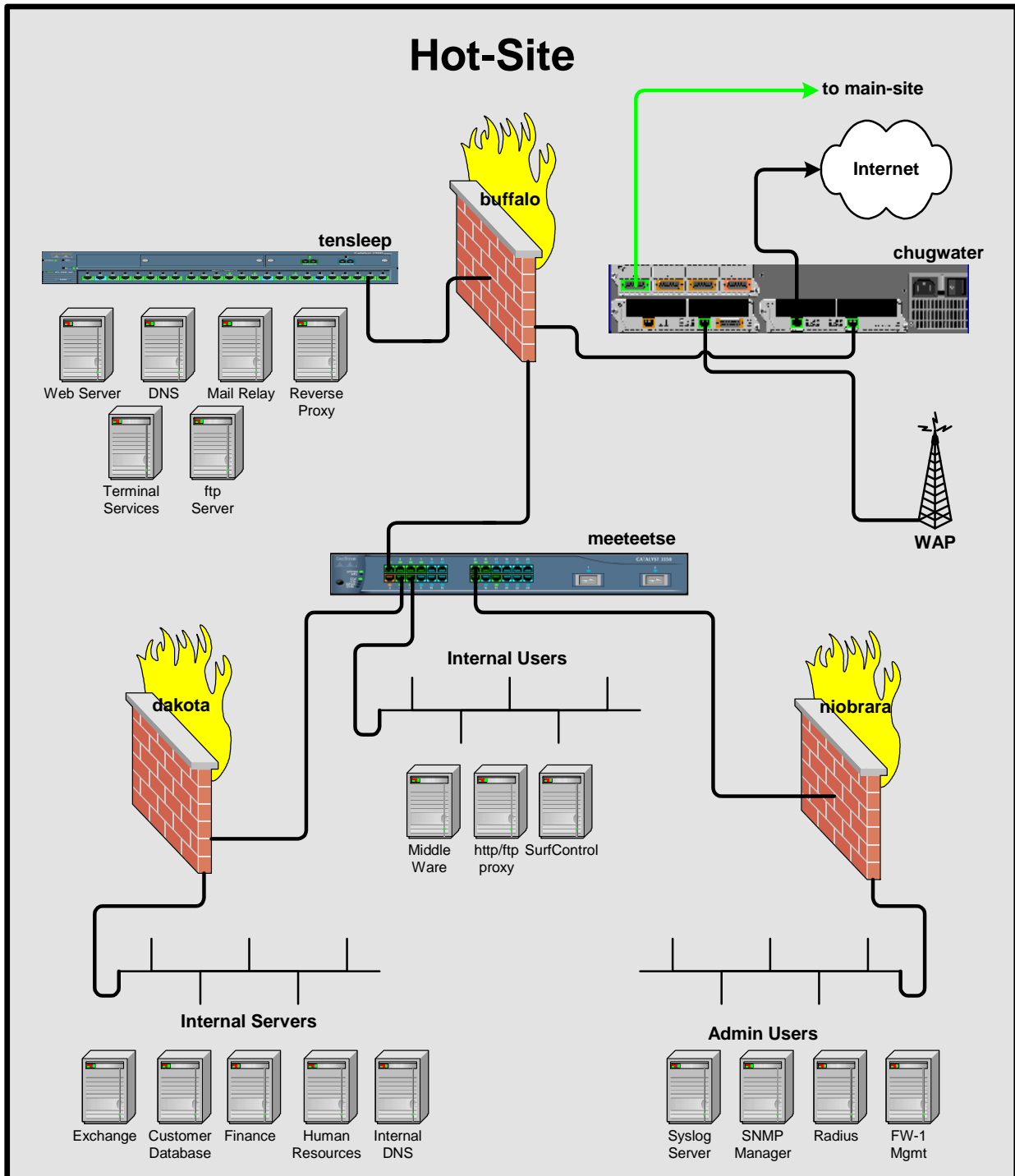
The data available on the web site will be kept to a minimum with the bulk accessible to the web server from the backend database. Once an hour the sites will be synchronized. The databases will be kept in sync with a constant incremental mirroring of the data. When the main sites database is updated the hot-site's slave database will subsequently also receive the updates. The hot-site also enables the main site to go offline for maintenance, without affecting business operations. Similarly the hot-site can also be used for testing any planned upgrades before putting them into production. Both these features will contribute to the total uptime of the site.



### 1.3 Cost Considerations

All but one of the partner and small office locations will connect using VPN's over the Internet, to reduce the cost of leased lines. The hot-site in Chicago is connected via a T-1 line to San Francisco. It is this link that will be used for

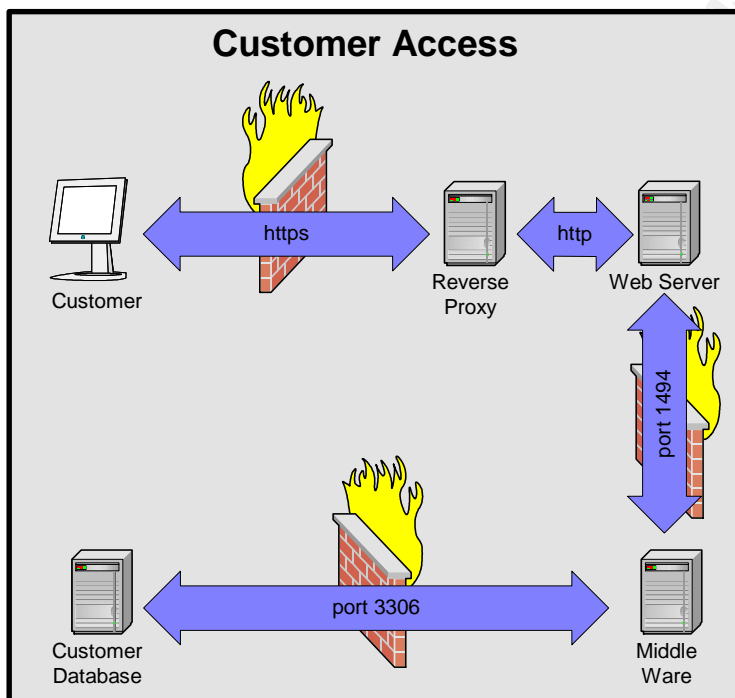
maintaining the synchronization of the headquarters site's web servers and backend databases with the hot-site. The cost for this can be justified, since all revenue is generated on-line, we need five nines availability. The hot site can give us redundancy and high availability in the event of a disaster, or other event at the headquarters site.



## 2 Access Requirements

### 2.1 Customers

The customer's point of contact or sale will be the web servers. The customer will be able to select from among numerous fortunes and add them to their shopping basket via their web browser. The fortunes will be retrieved from the backend database, along with any previous customer information pertinent to the sale. When the customer is ready for checkout, they may pay for the sale using a credit card. All transactions will be over a secure connection using https. The fortunes will then be downloaded to the customer upon approval of the credit card. Once the transactions are complete all data is moved to the backend database. In the event of a compromise only the web server will be affected. No other data will be exposed and any cleanup will involve just a restore of the web server.



The web servers on the screened subnet will not be allowed access to the backend database. In essence the front-end web servers are page servers only. This will be accomplished by some form of middleware on a host on the internal net. Only the middleware server will perform any transaction processing, or interactions with the backend database. So the web servers will have access on one port

to the middleware server, which in turn will have access to the database. This follows the principle of defense-in-depth, since the compromise of the web-server does not expose any data. The data can only be accessed if the web server and middleware server are both compromised.

The access to the web servers will be through a reverse proxy. This will allow for load balancing and/or scrubbing the commands sent to the web-servers. In a sense this will also give us a form of intrusion prevention. Also any https will be to the front-end reverse proxy. The connection to the web-servers will be http in order to allow us to deploy a Snort intrusion detection sensor. Since we can't deploy an IDS on encrypted traffic, we will let the back connections be http so we

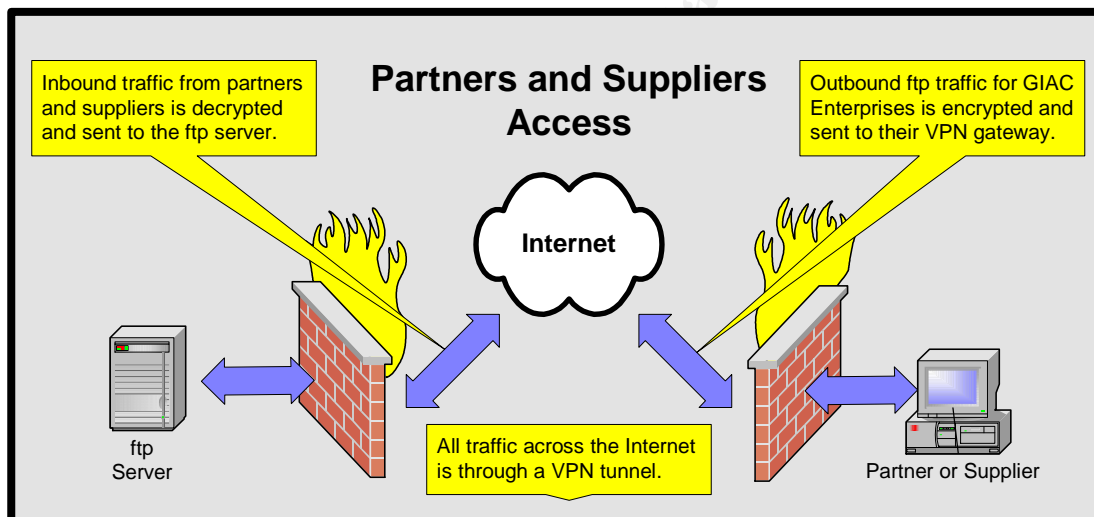


can watch the traffic for any attempt that might make it past the scrubber on the reverse proxy.

The customers will also be able to contact us through the phone of course and via e-mail. The e-mail system will consist of a Trend Micro mail relay on the screened subnet to scrub the incoming mail for worms, Trojans and other malicious content, before delivery to the internal Exchange server. Also a third party e-mail scrubber Postini, will be deployed to do further scrubbing and to perform anti-spam functions. All of the hosts on the screened subnet will be segregated from each other via VLAN

## 2.2 Partners and Suppliers

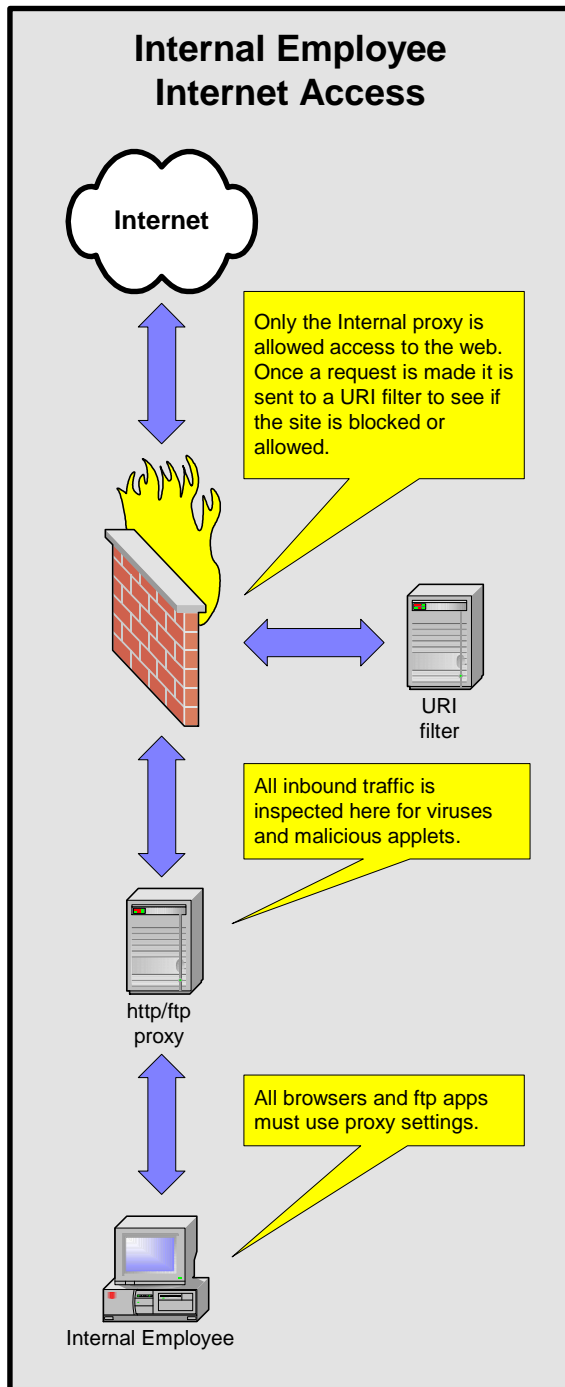
Supplier access will be via site-to-site VPN. Since the suppliers and partners will need to be able to transfer bulk data to us and vice versa. Their connections will be through a VPN to an ftp server on the screened subnet only.



Partners and suppliers will not have access to the internal database. Any uploads or downloads they need to make will have to be staged to an ftp server on the screened subnet. The files retrieved from the ftp server by internal employees will be run through a proxy for scrubbing. This limited access will help insure that infected or trojaned programs do not easily make their way onto the LAN.

## 2.3 Internal Employees

The Internal employees access will be determined by function. General-purpose access to the Internet will be permitted to the web through http and ftp. This access will be through an internal proxy server running Trend Micro. All internal employees will go through the proxy to access the web. Also there will be a URI filter such as SurfControl to allow or deny access to different categories of sites. For example, no access will be allowed to web-mail sites.



Split-DNS will be implemented. The screened subnet will have its own DNS to permit external access to the mail relay and web servers. There will be a redundant host providing the same function at the hot-site. The internal name server will provide lookup functions for internal hosts and Internet lookup. It will also have a backup at the hot-site.

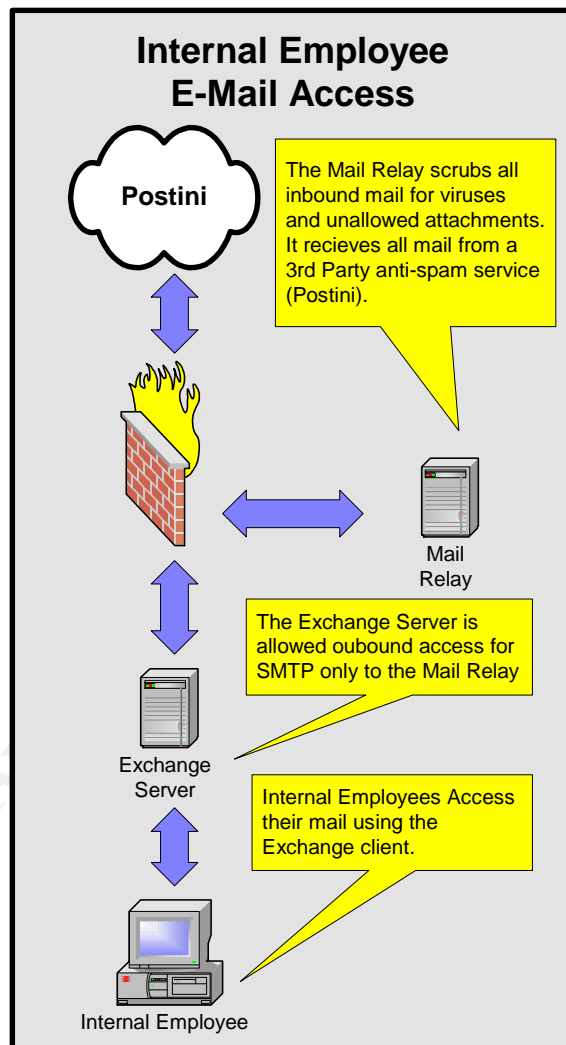
Https will be allowed outbound with no scrubbing. General-purpose e-mail access will be from the Internal Exchange server. All employees will have access to internal file servers for storage of their files. Other access to internal files/applications will be by function, for example Finance employees will have access to the payroll server. HR employees will have access to their server and so forth.

The IT employees will be on their own subnet or admin network. The internal servers (Application Servers, File/Print Servers, DHCP, DNS, the main customer Database, Exchange, Radius, Firewall Mgmt Station, Syslog Servers) also will have their own subnet. These subnets will be protected by an internal firewall from a different vendor than the primary firewall. In addition to the general-purpose access the IT employee's will have SSH access to the internal servers for administrative purposes. They will also have SSH access to

the firewall and network equipment from the internal subnets only. The authentication part of this access will be provided by a RADIUS server. There will be a secondary RADIUS server at the hot-site to provide redundancy

Internal employees will use the Exchange client to retrieve their mail from the Exchange server. The Exchange server forwards and receives its mail to the Mail Relay. The Mail Relay is a Trend Micro InterScan Viruswall, which scans all mail for virus and unauthorized attachments. From here the inbound mail is received from Postini, which performs an anti-spam function. The mail is setup this way to help eliminate the proliferation of worms and viruses.

The internal DNS will have the exchange server for its MX record and the external DNS on the screened subnet will have Postini for its MX record. Postini will be a mail relay for our domain. It will forward mail to the mail relay after performing the anti-spam function. Internal hosts will not be able to send and receive mail. That function will be provided by the combination of Exchange and the Mail Relay. This funnel point allows us to check all mail for unauthorized attachments and viruses. Also if the Mail Relay is compromised or disabled, internal mail is still up. In order for an attacker to get to the Exchange server they must first gain elevated privileges on the Mail Relay. The relay is also easier to restore than the Exchange server.

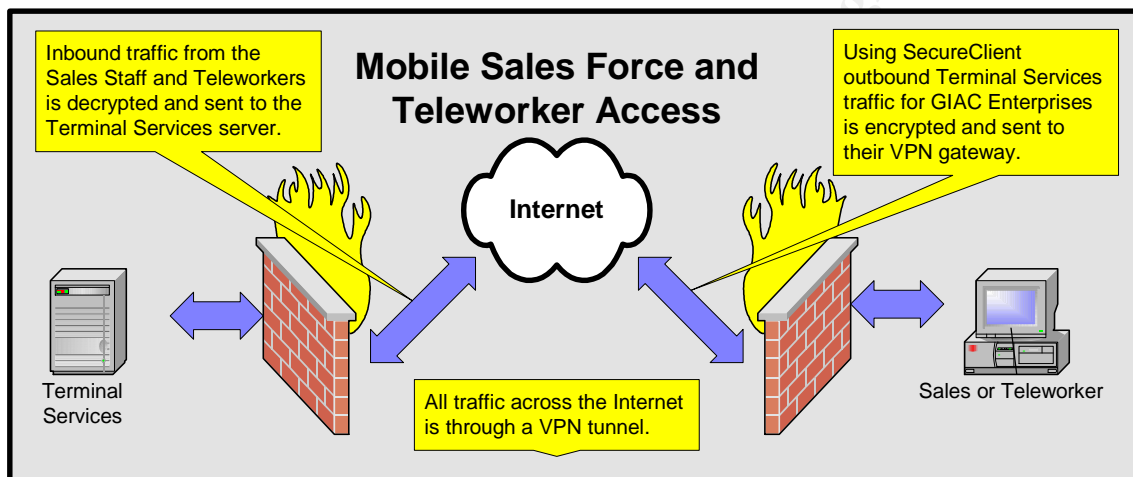


Port based security will also be implemented for all switches on the internal LAN. No unauthorized devices will be allowed to gain Layer 2 access without IT permission.

Admin employees will have access to the border router through a VPN form an Internal Admin network. They will also have access via SSH to hosts on the screened subnet, and the firewall. Their access to the WAP will be SSH but not over a VPN because as of now there is no Cisco IOS IPSEC code for that device.

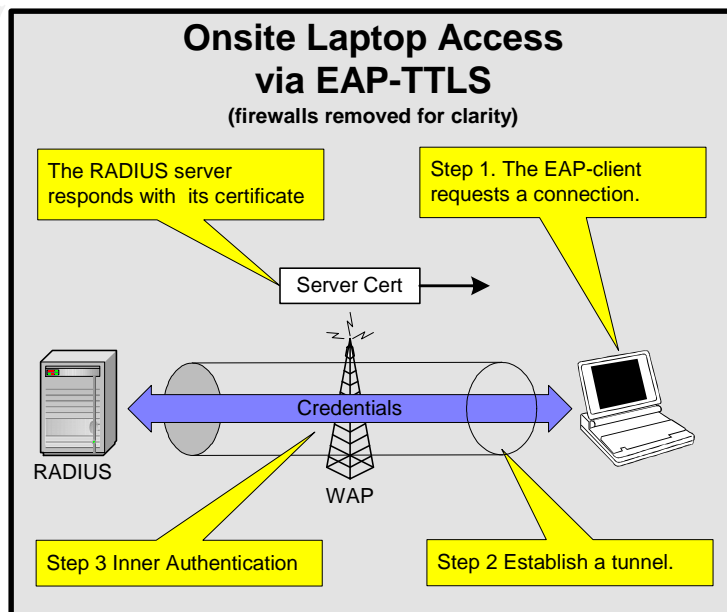
## 2.4 Mobile Sales Force and TeleWorkers

Access for the Sales Force and Teleworkers will be through CheckPoint's SecureClient to a Terminal Services server on the screened subnet. The Terminal server desktop will have access to Exchange, the http proxy and Internal Applications. This will allow the sales people to place orders, retrieve e-mail, and access Internal applications. They will retrieve their policy from the policy server; the policy will not allow other connections to their machine while connected to the GIAC Enterprises VPN gateway.



Access will be permitted after authentication through a RADIUS server; the hot-site will provide a redundant RADIUS server. When offsite the sales people will get their Internet Access from a third party provider.

Laptops onsite will only be allowed to connect through the VPN, never to the LAN. These devices will be given external addresses through which they may connect to the internal networks. These addresses will come from a Wireless Access Point. Yeah, I know WEP is insecure. For this we will use WiFi Protected Access with 802.1x-based authentication. 802.1x authentication or Extensible Authentication Protocol is a new WLAN security method. After authenticating to the WAP using the EAP-TTLS



mechanism, the client will be associated to the WAP and given an IP address. This authentication will use EAP-TTLS.

EAP-TTLS does not require that each user be issued a certificate. User authentication is performed by password but the credentials are passed in an encrypted tunnel based upon the server certificate. When the server determines a user has made a request it sends it certificate to the user. It is not necessary to create an infrastructure for certificates and authentication can be performed against a variety security databases. For example, Windows Domain Controllers, SQL or LDAP databases or token systems can all be used.<sup>1</sup>

So even if an attacker can break the WEP key, there is an inner authentication that they will not be able to sniff the authentication. Once the users have access to the WAP their VPN client will initiate a tunnel to Terminal Services (see the Mobile Sales Force and TeleWorkers Diagram).

Wireless brings its own security problems into the mix but by using WPA and 802.1x along with the VPN these are reduced if not eliminated. This allows us to eliminate the laptops need to connect directly back to the inside network. This enables us to funnel the laptops access through Terminal Services. This will aide in not allowing infected programs to gain access to the Internal LAN. This is a very significant detail in that; our laptops are not a constant source of worms, Trojans or viruses, infecting our LAN devices. I believe this trade-off is acceptable and any additional risk is mitigated through the additional security features.

In addition no magnetic media will be allowed to be removed or to be brought onto the premises without pre-approval from Management and screening by IT staff.

## **2.5 General Public**

The general public will have access via the web-servers, telephone and e-mail. The physical security of the data center will incorporate the traditional locked room with access permitted to IT employees only. In addition on-site security will be trained to watch for unauthorized persons upon the premises.

# **3 Network Design**

## **3.1 Border Router**

The role of the router will be to assist in our defense-in-depth strategy. We will use its packet filtering capabilities to mirror the firewall rules, only allowing in what the firewall will accept. The router with its simple filtering will help reduce the processing on the firewall itself. We will also allow the router to be a router in that we won't use its statefull capabilities but will leave that to the firewall. In addition we will use the Center for Internet Security<sup>2</sup> level-2 benchmark and the Router Auditing Tool to verify the benchmark.

Our Ingress and Egress rules at the network border are modeled from Northcutt<sup>3</sup>, the ingress rules, “which are applied to the serial interface on the WAN link to the Internet.” They will be as follows:

#### Ingress Filter

- Deny and log packets that have invalid source addresses.
- Allow HTTP and HTTPS that are addresses to the web server.
- Allow SMTP to the mail relay.
- Allow IPSEC and IKE packets that are addressed to the external firewall.
- Allow inbound established connections.
- Deny and log any other inbound traffic.

#### Egress Filter

- Allow outbound traffic that has valid source addresses from our networks.
- Deny and log any other outbound traffic.

The T-1 to the hot-site line will be placed outside the firewall to enable failover for the Internet. We could have placed it inside for greater protection, but this way a floating static route on the router can enable failover to the hot-site. It will also have an ACL to avoid misuse. We can reuse the Egress filter for the Internet link, on this link also. The Ingress filters will need to be slightly modified to allow traffic between the two sites.

### 3.2 Firewall

The firewall purpose is to enforce the policies we define.<sup>4</sup> The policies delineate the access control. For example, who is authorized to use services and files on the servers, and what traffic is allowed to cross between the local network and the Internet. Other additional services provided by the firewall include Network Address Translation, URL filtering, authentication, and encryption.

Following the principle of least privilege, our basic approach to the firewall policies or rules will be to deny everything unless explicitly permitted. The firewall is a key piece in the security architecture. Any of the traffic that we allow needs to be scrutinized as it impacts the other aspects of the architecture. Any access that is allowed may need a corresponding defense at another layer of our defense in depth strategy. For example, when we allow inbound access to our e-mail relay, we need to insure that the system has been hardened and is updated regularly with the appropriate security patches. The implementation of an Intrusion Detection Sensor to watch any of the inbound traffic we allow is another good example of mitigating any of the risks we create by allowing access.

Network Address Translation will be performed by the firewall for Internal hosts not on the screened subnet. We will use a many to one translation so all

addresses will appear to come from the firewalls external address. This provides another layer of security (along with disabling IP source routing on the border router) external hosts cannot establish connections to hosts that are NAT'ed. It also hides our internal network-addressing scheme.

Our screened subnet will include a VLAN implementation to separate the hosts located there. Each host will have its own VLAN and will not be able to communicate with other hosts on the same switch. This provides separation of resources. If one of the hosts is compromised on the screened subnet it will not have access to the other hosts due to the VLAN implementation. This helps prevent attackers from leveraging one successful compromise into gaining the whole subnet, since they cannot connect to the other devices.

### 3.3 VPN

In his book *Mastering Network Security*, Chris Brenton<sup>5</sup> states, “ a virtual private network is an authenticated and encrypted communications channel across some form of public network.” We use the Internet to eliminate the need for costly point-to-point connections and dialup modem pools. Since the Internet is inherently secure we will use VPN's to enable secure transfer of information between GIAC enterprisers and our partners and suppliers. Basically this will entail setting up a site-to-site VPN to enable transfer of information between the sites. So the purpose of the VPN is to enable secure communications between our sites to mobile and fixed-site users. Its role is to help eliminate the use of expensive dial-up and dedicated WAN links.

For our mobile users, we will also use CheckPoints SecureClient solution, which enables individual workstations to establish a VPN tunnel with the firewall. It also provides a firewall for the workstation, that we can centrally manage. Each workstation gets its policy from the Security Policy Server when its tunnel is established.

Our border router will also use a site-to-site VPN. This VPN will be established with our firewall to allow internal admin access. In addition it will allow monitoring of the device via Syslog and SNMP. Since there will always be devices outside of the firewall that we will need access to, this is a much “cleaner” configuration. With this scheme we avoid opening SSH to the external address space and we do not have to open inbound ports from these devices on the firewall, the VPN will address this.

### 3.4 IP Addressing Scheme

Our IP addressing scheme will include public and private address space. For our internal addresses we will use private addresses from the 192.168.0.0 range. Our DMZ and border routers will use public address space from addresses that have not been allocated for use. The complete listings of the address space assignment can be found in the Appendix.

### 3.5 Optional Components

Snort Intrusion Detection Sensors will be placed on taps at strategic locations to augment the Security Architecture. The purpose of the IDS is to identify attacks and security incidents. Its role in our security architecture is to find attacks in their early stages and so we may put a stop to them before they progress. An IDS can also be used to identify vulnerabilities and weaknesses. Once identified these weaknesses can be mitigated. The IDS can also be used to help correlate distributed type attacks that might occur over the network.

We will place a sensor outside the firewall, to detect attacks in their early stages, such as reconnaissance. We will also place one inbound to our Internal Net. The screened subnet will also receive one since these hosts have incoming connections from the Internet and are typically the most vulnerable.

The implementation is shown in the diagram on the next page. We will use taps to collect the traffic. They are a passive device that regenerates the traffic into a TX and an RX stream. The VLAN's in the diagram are for keeping the traffic from each zone separated. Since we are using an OS (Solaris x86) that does not support channel bonding, we need another way to aggregate the TX and RX streams from the taps. This can be accomplished by port mirroring. The sensor is multi-homed but is secure, first of all it is all regenerated traffic, and secondly the sniffing interfaces have no IP addresses. As a matter of fact they are not even up and don't even need to be plumbed to sniff the traffic. I came across this fact quite unrepentantly while implementing another IDS system. It is best to leave the duplexing on the ports set to auto detect for this particular setup. On a Catalyst 2924 the configuration to do this is as follows.

```
interface FastEthernet0/1
description to rx tap port
switchport access vlan 2
!
interface FastEthernet0/2
description to tx tap port
switchport access vlan 2
!
interface FastEthernet0/3
description to Snort Sensor
switchport access vlan 2
port monitor FastEthernet0/1
port monitor FastEthernet0/2
!
```

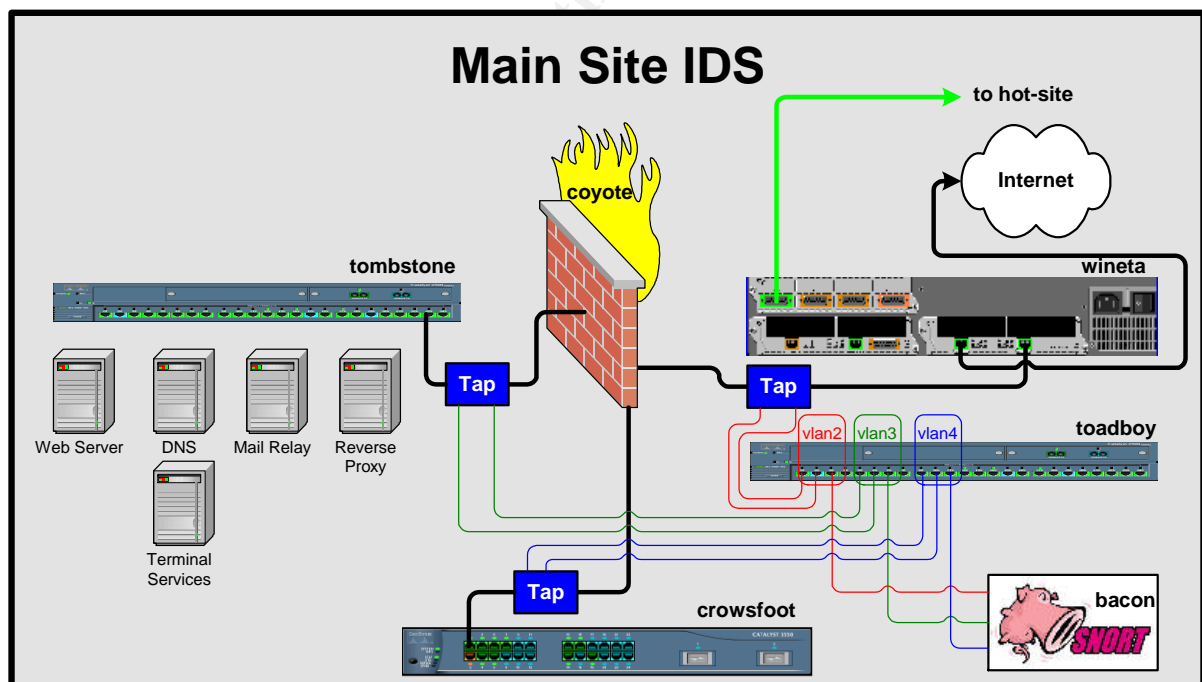
The Snort sensor will run three instances of Snort and send its alerts to a MySQL database. For moderate traffic this setup will work well. If we start dropping packets we can break out to multiple sensors. Also if the traffic is high we can implement barnyard which decouples the packet capture decode phase of Snort from the input output function. We will use the Analyst Console for Intrusion Databases for analyzing our alerts in offline mode. We will implement



Oinkmaster to manage and keep our alert signatures up to date. Additionally we will implement SnortSnarf, which takes the alerts and generates html. SnortSnarf works well for viewing snapshots of alerts.

For the screened subnet a Switch implementation is probably called for to provide real-time alerting capabilities. We will have Swatch look at the logs for that particular instance of Snort. When an alert is detected it can either e-mail us or send an SMB popup. When we detect an alert we can then use the screening router to block any further activity from the offending netblock. Or we could implement SnortSam to auto block it for us. SnortSam is able to send commands to our firewall, telling it to block the source or destination of a particular alert. The SnortSam directive also includes a time component. The IP that is blocked can be unblocked after a suitable time period, like after an hour.

One more thing to do is participate in a distributed intrusion detection system. Since we are already generating alerts, we can now e-mail them on an hourly basis to a DIDS like D-Shield. Some of the benefits in participating in a DIDS include the FightBack response, if we sign up, selected attacks may be forwarded to the ISP from which the attacks came. In addition reports and database summaries are available to anyone about offending addresses.



A Reverse Proxy will be implemented for the web servers on the screened subnet.

### 3.6 Component Descriptions

The component descriptions include the brand, version, and hardware type. The Security Role it plays and how its placement fulfills that role. These descriptions are included in the appendix.

© SANS Institute 2004, Author retains full rights.

## Assignment 2 Security Policy and Tutorial

### 1 Border Router

#### 1.1 Component Description and Security Role

The border router is a Cisco 3640. The 3640 is a modular access platform for medium and large size offices. The software it is running is:

```
Cisco Internetwork Operating System Software
IOS (tm) 3600 Software (C3640-IK9S-M), Version 12.2(11)T, RELEASE
SOFTWARE (fc1)
Copyright (c) 1986-2002 by cisco Systems, Inc.
```

Its main function is to forward traffic between subnets. Its purpose in our Security Architecture is to act as a screening device for our perimeter security. It becomes part of our boundary between the trusted and untrusted networks. That is it does not let in junk packets from the Internet nor does it give away information about our network. The routers main task will be routing and performing some additional security duties. For example filtering out spoofed addresses and only allowing "normal" traffic. So we will try to use the border router to mirror the firewall rules allowing in only what the firewall will accept. As much as this is possible. This will reduce the drops in the firewall logs and also reduce the amount of alerts on the Snort sensor. It will help cleanup some of the "noise" from the Internet so you have fewer events to concentrate on.

#### 1.2 Center for Internet Security

To help automate the process of hardening the router and as a double check of our security configuration we will use the Center for Internet Security Audit Checklist. The checklist covers limiting access or securing the router itself, disabling unnecessary services, network protection and additional security. Once we complete the checklist. We use ncat\_config to setup our local configurations so we may run the Router Auditing Tool<sup>6</sup>. The annotated ncat\_config run which we will use is shown below.

```
C:\CIS\RAT\bin>ncat_config
ncat_config: Select configuration type [cisco-ios] ?
ncat_config: Applying rules from:
ncat_config: C:\CIS\RAT\etc\configs\cisco-ios\common.conf
ncat_config: C:\CIS\RAT\etc\configs\cisco-ios\cis-level-1.conf
ncat_config: C:\CIS\RAT\etc\configs\cisco-ios\cis-level-2.conf
ncat_config: C:\CIS\RAT\etc\configs\cisco-ios\local.conf
ncat_config: Apply some or all of the rules that are selectable [Yes] !
ncat_config: Apply some or all of CIS level 1 rules [yes] ?
ncat_config: Check rules and data related to system management [Yes] !
```

**ncat\_config: Use local authentication [no] ?**

We will use one local user to enable access in the event RADIUS is not available. The login local method will be the last applied so it will only be used if no reply from a RADIUS server.

**ncat\_config: Create new AAA model using local usernames and passwords [yes] !**

**ncat\_config: Create local usernames [yes] !**

**ncat\_config: Username of user for local authentication [joecisco] ?**

**ncat\_config: Apply standard SNMP checks [Yes] !**

**Info: skipping IOS - no snmp-server because it conflicts with IOS - forbid SNMP without ACLs which is already selected**

**ncat\_config: Forbid SNMP read-write [no] ?**

**ncat\_config: Forbid SNMP community string 'public' [yes] !**

**ncat\_config: Forbid SNMP community string 'private' [yes] !**

**ncat\_config: Require an ACL to be applied for all SNMP access [yes] ?**

**ncat\_config: Specify ACL number to be used for filtering SNMP requests [99] ?**

**ncat\_config: Define SNMP ACL [yes] ?**

**ncat\_config: Address block and mask for SNMP access [192.168.22.0 0.0.0.255] ?**

SNMP will be enabled so we may monitor the router for availability and performance. However access will be restricted to one host (the SNMP Manager) on the Internal Admin network.

**ncat\_config: Apply standard checks to control access to the router [yes] ?**

**Info: skipping Access Allow Telnet because it conflicts with Access Require SSH which is already selected**

**ncat\_config: Specify maximum allowed exec timeout [yes] !**

**ncat\_config: Exec timeout value [10 0] ?**

**ncat\_config: Disable the aux port [yes] ?**

**ncat\_config: Use default AAA login authentication on each line [no] ?**

**ncat\_config: Use explicit named AAA login authentication on each line [yes] ?**

**ncat\_config: Name for login AAA list [radius-login] ?**

Radius-login will be the name of the AAA authentication method.

**ncat\_config: require line passwords [no] ?**

**ncat\_config: Require an enable secret [yes] !**

**ncat\_config: Check line password quality [no] ?**

**ncat\_config: Check user password quality [yes] ?**

**ncat\_config: Require VTY ACL to be applied [yes] !**

**ncat\_config: Specify ACL number to be used for telnet or ssh [182] ?**

Here we limit access to the device. However the tool does not seem to recognize extended named acls. Such as:

```
ip access-list extended egress-filter
permit ip 85.112.229.0 0.0.0.255 any
permit ip 85.112.230.0 0.0.0.255 any
deny ip any any log
```

Looks like we have to use numbers.

**ncat\_config: Define simple (one netblock + one host) VTY ACL [no] ?**

We don't define this because it expects the ACL to contain one host only.

**ncat\_config: Disable unneeded management services [yes] ?**

**ncat\_config: Forbid finger service (on IOS 11) [yes] !**

**ncat\_config: Forbid identd service (on IOS 11) [yes] !**

**ncat\_config: Forbid finger service (on IOS 12) [yes] !**

**ncat\_config: Forbid finger service (on IOS 12) [yes] !**

**ncat\_config: Forbid http service [yes] !**

Unnecessary services are to be disabled.

**ncat\_config: Encrypt passwords in the configuration [yes] !**

ncat\_config: Check rules and data related to system control [Yes] !  
ncat\_config: Synchronize router time via NTP [yes] ?  
ncat\_config: Designate an NTP time server [yes] !  
ncat\_config: Address of first NTP server [63.149.208.50] ?  
ncat\_config: Designate a second NTP time server [yes] ?  
ncat\_config: Address of second NTP server [132.163.4.103] ?  
ncat\_config: Designate a third NTP time server [yes] ?  
ncat\_config: Address of third NTP server [192.43.244.18] ?  
ncat\_config: Apply standard logging rules [yes] ?  
Info: skipping GMT Rules because it conflicts with Localtime Rules which is already selected  
ncat\_config: Timestamp log messages [yes] !  
ncat\_config: Timestamp debug messages [yes] !  
ncat\_config: enable logging [yes] !  
ncat\_config: Designate syslog server [yes] !  
ncat\_config: Address of syslog server [192.168.22.22] ?  
ncat\_config: Designate local logging buffer size [yes] !  
ncat\_config: Local log buffer size [16000] ?

Apply rules for logging.

ncat\_config: Require console logging of critical messages [yes] !  
ncat\_config: Require remote logging of level info or higher [yes] !  
ncat\_config: Disable unneeded control services [yes] ?  
ncat\_config: Forbid small TCP services (on IOS 11) [yes] !  
ncat\_config: Forbid small UDP services (on IOS 11) [yes] !  
ncat\_config: Forbid small TCP services (on IOS 12) [yes] !  
ncat\_config: Forbid small UDP services (on IOS 12) [yes] !  
ncat\_config: Forbid bootp service [yes] !  
ncat\_config: Disable CDP service [yes] ?  
ncat\_config: Forbid config service [yes] ?  
ncat\_config: Use tcp-keepalive-in service to kill stale connections [yes] !  
ncat\_config: Forbid tftp service [yes] ?

Unnecessary services are to be disabled

ncat\_config: Check rules and data related to data flow [Yes] !  
ncat\_config: Apply standard routing protections [yes] ?  
ncat\_config: Forbid directed broadcasts (on IOS 11) [yes] !  
ncat\_config: Forbid directed broadcasts (on IOS 12) [yes] !  
ncat\_config: Forbid IP source routing [yes] !

Disabling these, defeats several types of attacks.

ncat\_config: Apply some or all of CIS Level 2 rules [yes] ?  
ncat\_config: Check rules and data related to system management [yes] ?  
Info: skipping TACACS Plus AAA Rules because it conflicts with Local AAA Rules which is already selected

ncat\_config: Apply level 2 checks to control access to the router [Yes] !  
ncat\_config: Require use of SSH for remote administration? [yes] ?  
ncat\_config: Check for SSH transport only on VTYs [yes] ?  
ncat\_config: Require VTY ACL to be applied [yes] !  
ncat\_config: Define VTY ACL [no] ?

I replied no for the above because I don't care for their acl. We will define our own.

ncat\_config: Check rules and data related to system control [yes] ?  
ncat\_config: Apply non-standard logging rules [yes] ?  
ncat\_config: Use localtime for logging instead of GMT [yes] ?  
ncat\_config: Local timezone name [CST] ?  
ncat\_config: Local timezone offset from GMT [6] ?  
ncat\_config: Check timezone and offset [yes] ?

```
ncat_config: Require summertime clock changes [yes] ?
ncat_config: Apply loopback checks [yes] ?
ncat_config: What is the local loopback interface number [0] ?
ncat_config: Use primary loopback as source address for NTP [yes] ?
ncat_config: Check the existence of the defined loopback interface [yes] ?
ncat_config: What is the local loopback address [85.112.229.45] ?
ncat_config: Check the existence of the defined loopback interface [yes] ?
    Using a loopback is considered best practice. We may then choose that
    traffic generated by the router itself, will use the loopback for its source IP. This
    helps in configuring the security of other devices in the network.
ncat_config: Forbid all non-standard loopbacks [yes] ?
ncat_config: Use loopback for tftp source interface [no] ?
ncat_config: Disable unneeded services [yes] ?
ncat_config: Check rules and data related to data flow [yes] ?
ncat_config: Apply border router filtering rules [yes] ?
ncat_config: What is the primary external interface [fa0/1] ?
ncat_config: Does this border router have a second external interface [yes] ?
ncat_config: What is the secondary external interface [s3/0] ?
ncat_config: Apply ingress filter to second external interface [no] ?
ncat_config: What ACL number (100-199) should be used for ingress filtering [181]
?
```

Network protection is checked for here by the presences of ingress and egress ACLS.

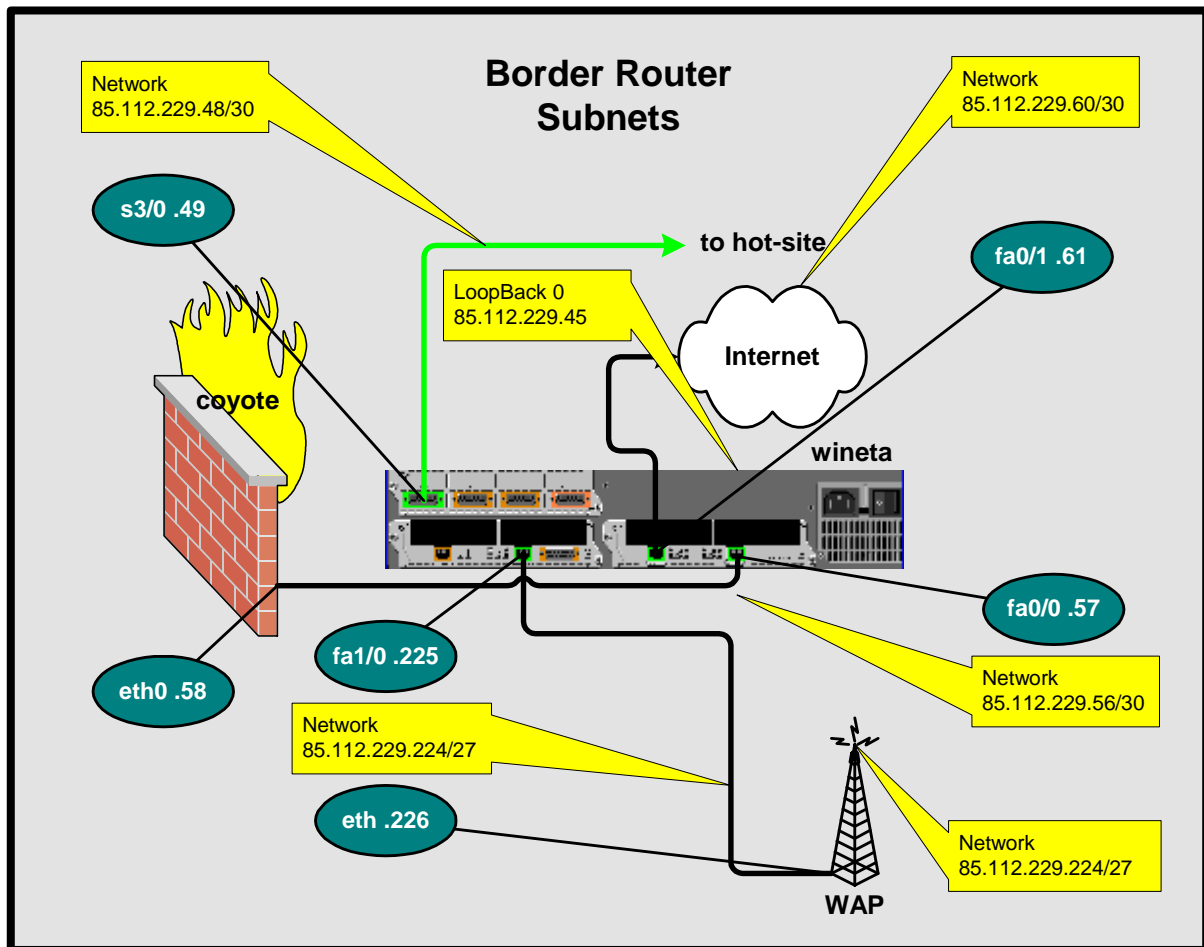
```
ncat_config: Apply egress filter to second external interface [yes] ?
ncat_config: What ACL number (100-199) should be used for egress filtering [180] ?
ncat_config: Test for existence of 2nd external interface [no] ?
ncat_config: Define egress filter [no] ?
ncat_config: What is the the internal netblock and mask [85.112.229.0 0.0.0.255] ?
ncat_config: Apply ingress filter to external interface [yes] ?
ncat_config: Define ingress filter [no] ?
```

We do not have the tool define our ingress and egress filters. Since this will interfere with our custom rules. It will fail our ACL's.

```
ncat_config: Apply egress filter to first external interface [yes] ?
ncat_config: Test for existence of external interface [yes] ?
ncat_config: Apply extra routing protections [yes] ?
ncat_config: Use Unicast RPF for filtering [no] ?
ncat_config: Forbid proxy arp [yes] ?
ncat_config: Forbid tunnel interfaces [yes] ?
Saving selections to C:\CIS\RAT/etc/configs/cisco-ios/local.conf
C:\CIS\RAT\bin>
```

### 1.3 Network Diagram and Border Router Configuration

The diagram below shows the subnetting of the border router.



Our annotated configuration for the border router is as follows:

**version 12.2**

**service tcp-keepalives-in**

We will enable timestamps for our logging and SNMP traps, this enable us to correlate events with other devices on the network.

**service timestamps debug datetime msec show-timezone**

**service timestamps log datetime msec show-timezone**

Password encryption encrypts the passwords that are stored in the configuration file of the router. However it does not encrypt the SNMP community strings, or the pre-shared-secret for VPN connections. It is worth mentioning the encrypted strings, are also easily decrypted by tools readily available.

**service password-encryption**

**service linenumbers**

!

**hostname wineta**

!

**boot system flash:c3640-ik9s-mz.122-11.T.bin**

**logging buffered 16000 debugging**

**logging console critical**

!

We will use a RADIUS login for access to our border router. This gives us an audit trail of who accessed the device. The commands below enable a RADIUS login with a failover to a local user in the case RADIUS is unavailable. These commands also provide information to RADIUS as to when the access is started and when it stopped.

```
aaa new-model
aaa authentication login radius-login group radius local
aaa authorization exec default if-authenticated local
aaa accounting exec default start-stop group radius
aaa accounting network default start-stop group radius
aaa session-id common
enable secret 5 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!
username XXXXXXXX password 7 XXXXXXXXXXXXXXXXXXXX
clock timezone CST -6
clock summer-time cdt recurring
ip subnet-zero
```

We will disable source routing and the bootp server. Source Routing is a feature that enables the packets themselves to specify routes. This can be used in several types of attacks. For example, an attacker could spoof source IP addresses heading for our router or other device behind it and have the reply packets actually routed back to him. Bootp is rarely used and offers an attacker the opportunity to download a copy of the IOS. Finger, TCP and UDP small services are disabled by default in version 12.

```
no ip source-route
!
no ip bootp server
ip ssh time-out 60
ip ssh authentication-retries 2
```

The VPN rules below will be covered in the VPN section

```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
 crypto isakmp key Ab134c$Lmn%2 address 85.112.229.58
!
 crypto ipsec transform-set vpnset esp-3des esp-sha-hmac
!
 crypto map ssh_to_loop local-address FastEthernet0/1
 crypto map ssh_to_loop 1 ipsec-isakmp
 set peer 85.112.229.58
 set transform-set vpnset
 match address 101
!
```

We will use a loopback. It is considered best practice to define one loopback interface and designate it as the source interface for most traffic generated by the router itself.<sup>7</sup> When the router generates traffic it usually uses the outbound interface for the source address. This affects our other devices in the Security Architecture in that we have to define rules that allow traffic from interfaces that should only be used for routing packets. With the loopback we can also use that for our administrative access.



```
interface Loopback0
description Internal interface for Monitoring
ip address 85.112.229.45 255.255.255.252
no ip proxy-arp
!
```

By default Cisco devices perform proxy ARP on all interfaces. We will disable it on all interfaces. IP directed broadcast is disabled on all interfaces in Version 12. We will also turn off ICMP host unreachable and redirects. Attackers can use unreachable, to map our networks. Redirects can be used to send our traffic to a different destination than intended. We will also disable CDP on each interface. CDP or Cisco Discovery Protocol is a layer 2 Protocol that can be used to gather information about Cisco devices.

```
interface FastEthernet0/0
description External connection to Internet
ip address 85.112.229.61 255.255.255.252
ip access-group 181 in
no ip redirects
no ip unreachable
no ip proxy-arp
speed 100
full-duplex
no cdp enable
!
```

```
interface FastEthernet0/1
description Internal connection to Corporate Firewall
ip address 85.112.229.57 255.255.255.252
ip access-group 180 in
no ip proxy-arp
duplex auto
speed auto
no cdp enable
crypto map ssh_to_loop
!
```

```
interface Ethernet1/0
description External Connection to WAP
ip address 85.112.229.225 255.255.255.252
ip access-group 183 in
ip access-group 184 out
no ip redirects
no ip unreachable
no ip proxy-arp
full-duplex
no cdp enable
!
```

Unused interfaces will be shutdown.

```
interface Ethernet1/1
no ip address
no ip proxy-arp
shutdown
half-duplex
no cdp enable
!
```

```
interface Serial3/0
description Connection to Hot-Site chugwater-s3/0
```

```
ip address 85.112.229.49 255.255.255.252
ip access-group 179 in
no ip redirects
no ip unreachable
no ip proxy-arp
serial restart_delay 0
no cdp enable
```

!

```
interface Serial3/1
no ip address
no ip proxy-arp
shutdown
serial restart_delay 0
no cdp enable
```

!

```
interface Serial3/2
no ip address
no ip proxy-arp
shutdown
serial restart_delay 0
no cdp enable
```

!

```
interface Serial3/3
no ip address
no ip proxy-arp
shutdown
serial restart_delay 0
no cdp enable
```

!

```
ip classless
ip route 0.0.0.0 0.0.0.0 85.112.229.62
```

Below is a floating static route for failover to the hot-site.

```
ip route 0.0.0.0 0.0.0.0 85.112.229.50 2
```

The route below is for the screened subnet.

```
ip route 85.112.229.0 255.255.255.224 85.112.229.58
```

The route below is so the router knows where to log to, send its traps and RADIUS requests to. Also the WAP needs this route to enable it to send its RADIUS requests to the Internal-DNS. Otherwise it would use the default route and send them out to the Internet.

```
ip route 192.168.22.0 255.255.255.0 85.112.229.58
```

The route below is to send traffic to the hot-site across the dedicated link. All other outbound traffic will use the default route.

```
ip route 85.112.230.0 255.255.255.0 85.112.229.50
```

The http server provides web-based administration, however it has been the subject of numerous vulnerabilities and attacks. We will disable it.

```
no ip http server
```

!

Use the loopback for RADIUS requests.

```
ip radius source-interface Loopback0
```

We will enable logging to a syslog server at the host below using the loopback as a source. This traffic will be encrypted.

```
logging trap debugging
logging facility local0
```

**logging source-interface Loopback0**  
**logging 192.168.22.22**

The SNMP ACL, allows the SNMP Manager on the Internal Admin Network to perform SNMP queries on the router itself. We will use this for monitoring the border router. This traffic will be encrypted.

```
access-list 99 permit 192.168.22.23 0.0.0.0  
access-list 99 deny any log
```

The VPN ACL, determines which traffic will be encrypted. Its usage will be discussed in the VPN section.

```
access-list 101 permit esp host 85.112.229.58 host 85.112.229.57  
access-list 101 permit ip 192.168.22.0 0.0.0.255 host 85.112.229.45  
access-list 101 permit ip 192.168.23.0 0.0.0.255 host 85.112.229.45  
access-list 101 permit ip host 85.112.229.45 192.168.22.0 0.0.0.255  
access-list 101 deny ip any any log  
access-list 101 deny esp any any log
```

Ingress-filter for secondary external interface to hot-site. This ACL allows traffic in from the hot-site. This ACL is basically the same as the 181 which is discussed in detail below. It will be used if we loose our Internet connection and the traffic is routed to the Serial interface.

```
access-list 179 deny icmp any any log  
access-list 179 deny ip 10.0.0.0 0.255.255.255 any log  
access-list 179 deny ip 127.0.0.0 0.255.255.255 any log  
access-list 179 deny ip 172.16.0.0 0.15.255.255 any log  
access-list 179 deny ip 192.168.0.0 0.0.255.255 any log  
access-list 179 deny ip 85.112.229.0 0.0.0.255 any log  
access-list 179 deny ip any 10.0.0.0 0.255.255.255 log  
access-list 179 deny ip any 127.0.0.0 0.255.255.255 log  
access-list 179 deny ip any 172.16.0.0 0.15.255.255 log  
access-list 179 deny ip any 192.168.0.0 0.0.255.255 log  
access-list 179 permit udp any gt 1023 host 85.112.229.13 eq domain  
access-list 179 permit tcp any eq domain host 85.112.229.13 gt 1023 established  
access-list 179 permit udp any eq domain host 85.112.229.13 gt 1023  
access-list 179 permit tcp any gt 1023 host 85.112.229.17 eq 443  
access-list 179 permit tcp any gt 1023 host 85.112.229.17 eq www  
access-list 179 permit tcp any eq ftp-data host 85.112.229.58 gt 1023 established  
access-list 179 permit tcp any eq ftp host 85.112.229.58 gt 1023  
access-list 179 permit tcp any gt 1023 host 85.112.229.58 gt 1023 established  
access-list 179 permit tcp any eq www host 85.112.229.58 gt 1023 established  
access-list 179 permit tcp any eq 443 host 85.112.229.58 gt 1023 established  
access-list 179 permit tcp any eq smtp host 85.112.229.5 gt 1023 established  
access-list 179 permit tcp any gt 1023 host 85.112.229.5 eq smtp  
access-list 179 permit udp eq isakmp host 85.112.229.58 eq isakmp  
access-list 179 permit esp any host 85.112.229.58  
access-list 179 permit tcp any gt 1023 host 85.112.229.58 eq 256  
access-list 179 permit udp any eq ntp host 85.112.229.61 eq ntp  
access-list 179 deny ip any any log
```

Egress-filter for main-site. This ACL is used for antispoofing. Only source addresses from our network space will be allowed. The http/ftp proxy for internal addresses is NAT'ed to this space. We also need to add the Internal Admin Network here so the SSH traffic can pass. Since it is not NAT'ed, once it is decrypted it has the original source address. The ACL will be applied to this. The same is true of the RADIUS reply.

```
access-list 180 permit ip 85.112.229.0 0.0.0.255 any
```

```
access-list 180 permit tcp 192.168.22.0 0.0.0.255 any
access-list 180 deny ip any any log
```

Ingress-filter for primary external interface to Internet. This ACL controls the inbound traffic to the border router.

```
access-list 181 deny icmp any any log
```

First off we will drop all incoming ICMP, this will help aid in preventing network mapping and certain DOS attacks. The next rules drop traffic with a source or destination of the RFC 1812 address space. This traffic most likely malicious and is not needed. We also drop any traffic with a source of our address space. We should never see valid traffic coming into our network with a source of our address space.

```
access-list 181 deny ip 10.0.0.0 0.255.255.255 any log
access-list 181 deny ip 127.0.0.0 0.255.255.255 any log
access-list 181 deny ip 172.16.0.0 0.15.255.255 any log
access-list 181 deny ip 192.168.0.0 0.0.255.255 any log
access-list 181 deny ip 85.112.229.0 0.0.0.255 any log
access-list 181 deny ip any 10.0.0.0 0.255.255.255 log
access-list 181 deny ip any 127.0.0.0 0.255.255.255 log
access-list 181 deny ip any 172.16.0.0 0.15.255.255 log
access-list 181 deny ip any 192.168.0.0 0.0.255.255 log
```

The next three rules allow incoming DNS requests and replies.

```
access-list 181 permit udp any gt 1023 host 85.112.229.13 eq domain
access-list 181 permit tcp any eq domain host 85.112.229.13 gt 1023 established
access-list 181 permit udp any eq domain host 85.112.229.13 gt 1023
```

Allow incoming traffic to the web site.

```
access-list 181 permit tcp any gt 1023 host 85.112.229.17 eq 443
access-list 181 permit tcp any gt 1023 host 85.112.229.17 eq www
```

Allow incoming established connections for internal ftp-proxy.

```
access-list 181 permit tcp any eq ftp-data host 85.112.229.58 gt 1023 established
access-list 181 permit tcp any eq ftp host 85.112.229.58 gt 1023
```

The next rule is needed to allow passive ftp transfers.

```
access-list 181 permit tcp any gt 1023 host 85.112.229.58 gt 1023 established
```

Allow incoming established connections for internal http-proxy

```
access-list 181 permit tcp any eq www host 85.112.229.58 gt 1023 established
access-list 181 permit tcp any eq 443 host 85.112.229.58 gt 1023 established
```

Allow incoming and established SMTP requests.

```
access-list 181 permit tcp any eq smtp host 85.112.229.5 gt 1023 established
access-list 181 permit tcp any gt 1023 host 85.112.229.5 eq smtp
```

The next two rules allow incoming new and established IKE and ESP connections.

```
access-list 181 permit udp any eq isakmp host 85.112.229.58 eq isakmp
access-list 181 permit esp any host 85.112.229.58
```

Allow SecureClient Topology downloads.

```
access-list 181 permit tcp any gt 1023 host 85.112.229.58 eq 256
```

The next rule allows incoming NTP replies to the border router. An internal router will get its time from the border router and all other internal hosts will sync to the internal device.

```
access-list 181 permit udp any eq ntp host 85.112.229.61 eq ntp
```

Drop and log the rest.

```
access-list 181 deny ip any any log
```

The VTY ACL only allows access to the border router from the Internal Admin Network. This traffic is encrypted, when it gets to the border router it is decrypted and the original source address is used.

```
access-list 182 permit tcp 192.168.22.0 0.0.0.255 any
access-list 182 deny ip any any log
```

The WAP ACL below allows the WAP access to the Internal DNS. And the protocols necessary to establish a VPN connection.

```
access-list 183 permit udp any gt 1023 host 192.168.22.12 eq 1812
access-list 183 permit udp any gt 1023 host 192.168.22.12 eq 1813
```

The next two rules allow outgoing new and established IKE and ESP connections.

```
access-list 183 permit udp any eq isakmp host 85.112.229.58 eq isakmp
access-list 183 permit esp any host 85.112.229.58
```

Allow SecureClient Topology downloads.

```
access-list 183 permit tcp any gt 1023 host 85.112.229.58 eq 256
```

Allow SecureClient Policy downloads.

```
access-list 183 permit tcp any gt 1023 host 85.112.229.58 eq 18213
```

Allow SSH access from Internal Admin Network.

```
access-list 183 permit tcp host 85.112.229.226 eq 22 192.168.22.0 0.0.0.255 gt 1023
access-list 183 deny ip any any log
```

The WAP ACL below allows access only from the VPN Gateway and the Internal RADIUS server.

```
access-list 184 permit udp host 192.168.22.12 eq 1812 85.112.229.224 0.0.0.31 gt 1023
access-list 184 permit udp host 192.168.22.12 eq 1813 85.112.229.224 0.0.0.31 gt 1023
access-list 184 permit udp host 85.112.229.58 eq isakmp 85.112.229.224 0.0.0.31 eq isakmp
access-list 184 permit esp host 85.112.229.58 85.112.229.224 0.0.0.31
access-list 184 permit tcp host 85.112.229.58 eq 18231 85.112.229.224 0.0.0.31 gt 1023
access-list 184 permit tcp host 85.112.229.58 eq 256 85.112.229.224 0.0.0.31 gt 1023
access-list 184 permit tcp 192.168.22.0 0.0.0.255 gt 1023 host 85.112.229.226 eq 23
access-list 184 deny ip any any log
```

```
no cdp run
```

```
!
```

Use the loopback as a source for sending snmp traps. This traffic will be encrypted. We choose not to disable the snmp-server but instead rely on the VPN tunnel and ACL as a secure means of providing this service.

```
snmp-server engineID local 00000009020000107BA7FAC1
snmp-server community XXXXXXXXXXXX RO 99
snmp-server trap-source Loopback0
snmp-server chassis-id 364051726
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps tty
radius-server host 192.168.22.12 auth-port 1812 acct-port 1813
radius-server retransmit 1
radius-server deadline 5
radius-server key 7 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!
```

Statement about the router being subject to monitoring, and legal notice prohibiting unauthorized access.



```
auditing 85.112.229.45...
Parsing: /C:\CIS\RAT/etc/configs/cisco-ios/common.conf/
Parsing: /C:\CIS\RAT/etc/configs/cisco-ios/cis-level-1.conf/
Parsing: /C:\CIS\RAT/etc/configs/cisco-ios/cis-level-2.conf/
Parsing: /C:\CIS\RAT/etc/configs/cisco-ios/local.conf/
Checking: 85.112.229.45
done checking 85.112.229.45.
Parsing: /C:\CIS\RAT/etc/configs/cisco-ios/common.conf/
Parsing: /C:\CIS\RAT/etc/configs/cisco-ios/cis-level-1.conf/
Parsing: /C:\CIS\RAT/etc/configs/cisco-ios/cis-level-2.conf/
Parsing: /C:\CIS\RAT/etc/configs/cisco-ios/local.conf/
ncat_report: writing 85.112.229.45.ncat_fix.txt.
ncat_report: writing 85.112.229.45.ncat_report.txt.
ncat_report: writing 85.112.229.45.html.
ncat_report: writing rules.html (cisco-ios-benchmark.html).
ncat_report: writing all.ncat_fix.txt.
ncat_report: writing all.ncat_report.txt.
ncat_report: writing all.html.
```

Upon successful connection the RAT tool will grab the configuration and compare it to the rules files. The one we generated using ncat\_config is C:/CIS/RAT/etc/configs/cisco-ios/local.conf/. It then will generate a fix report. The fix-report contains the commands to fix problems it flags. It also will generate a score based on the report.

The RAT tool requires telnet access, so if you are using SSH you might need to disable it or find another way to get the config files. To disable SSH:

```
Crypto key zeroize rsa
Line vty 0 15
  Transport input telnet
```

Some versions of Cisco 12 code are vulnerable to a Denial of Service attack if the SSH server is enabled. A table listing all the versions affected, and their available fixes can be found here:

<http://www.cisco.com/warp/public/707/ssh-packet-suite-vuln.shtml#Software>

At the time of this writing another set of exploit code for the Cisco IOS platform was released.

<http://www.cisco.com/warp/public/707/cisco-sn-20040326-exploits.shtml>  
<http://www.k-otik.com/exploits/03.28.cge.pl.php>

We are not using telnet, or the http server so those don't apply to our situation. Also we are not using the firewall features of the router so the UDP Flood Denial of Service Vulnerability does not apply. All releases which have the SSH server feature are vulnerable when the SSH server is enabled by issuing the command **crypto key generate rsa** in configuration mode. This does apply

to us. So to keep the SSH server functionality, we will restrict access to specific source IP addresses on the VTY lines through ACL's, as specified in the CIS Audit Checklist.

### 1.4 Router Auditing Tool Results

The results of the RAT tool run aide in obtaining a secure configuration. The tool is not a substitute for knowing the configuration parameters of the router, but rather is a double check on our configuration. It cannot check for a mis-configured ACL, but simply for its existence. So it is not a substitute for an experienced administrator reviewing the ACL's for correctness. The output from the RAT tool is shown below.

Importance	Pass/Fail	Rule
10	pass	IOS - no ip http server
10	pass	IOS - login named list
10	pass	IOS - forbid SNMP without ACLs
10	pass	IOS - forbid SNMP read-write
10	pass	IOS - forbid SNMP community public
10	pass	IOS - forbid SNMP community private
10	pass	IOS - enable secret
10	pass	IOS - apply VTY SSH ACL
10	pass	IOS - apply VTY ACL
10	pass	IOS - Define SNMP ACL
7	pass	IOS 12 - no udp-small-servers
7	pass	IOS 12 - no tcp-small-servers
7	pass	IOS 12 - no directed broadcast
7	pass	IOS - no service config
7	pass	IOS - no ip source-route
7	pass	IOS - no cdp run
7	pass	IOS - exec-timeout
7	pass	IOS - encrypt passwords
7	pass	IOS - Apply ingress filter
7	pass	IOS - Apply egress filter to first external interface
5	pass	IOS 12.1,2,3 - no finger service
5	pass	IOS - user password quality
5	pass	IOS - tcp keepalive service
5	pass	IOS - set syslog server
5	pass	IOS - service timestamps logging
5	pass	IOS - service timestamps debug
5	pass	IOS - require clock summer-time
5	pass	IOS - ntp source
5	pass	IOS - ntp server 3
5	pass	IOS - ntp server 2
5	pass	IOS - ntp server
5	pass	IOS - no ip proxy-arp



5	pass	IOS - no ip bootp server
5	pass	IOS - logging buffered
5	pass	IOS - enable logging
5	pass	IOS - VTY transport SSH
5	pass	IOS - One loopback interface must exist
5	pass	IOS - Defined loopback must be only loopback
3	pass	IOS - logging trap info or higher
3	pass	IOS - logging console critical
3	pass	IOS - disable aux
3	pass	IOS - clock timezone - localtime

Summary for wineta

#Checks	#Passed	#Failed	%Passed	
42	42	0	100	
PerfectWeightedScore		ActualWeighedScore		%WeightedScore
272		272		100
Overall Score (0-10)	10			

Note: PerfectWeightedScore is the sum of the importance value of all rules.  
 ActualWeighedScore is the sum of the importance value of all rules passed,  
 minus the sum of the importance each instance of a rule failed

## 2 Primary Firewall

### 2.1 Component Description and Security Role

The primary firewall is CheckPoint NG with Application Intelligence (R55). It is a SecurePlatform install running on an IBM e-series server. Its main function is to enforce our security policy to maintain perimeter security. Its https screen is shown below.

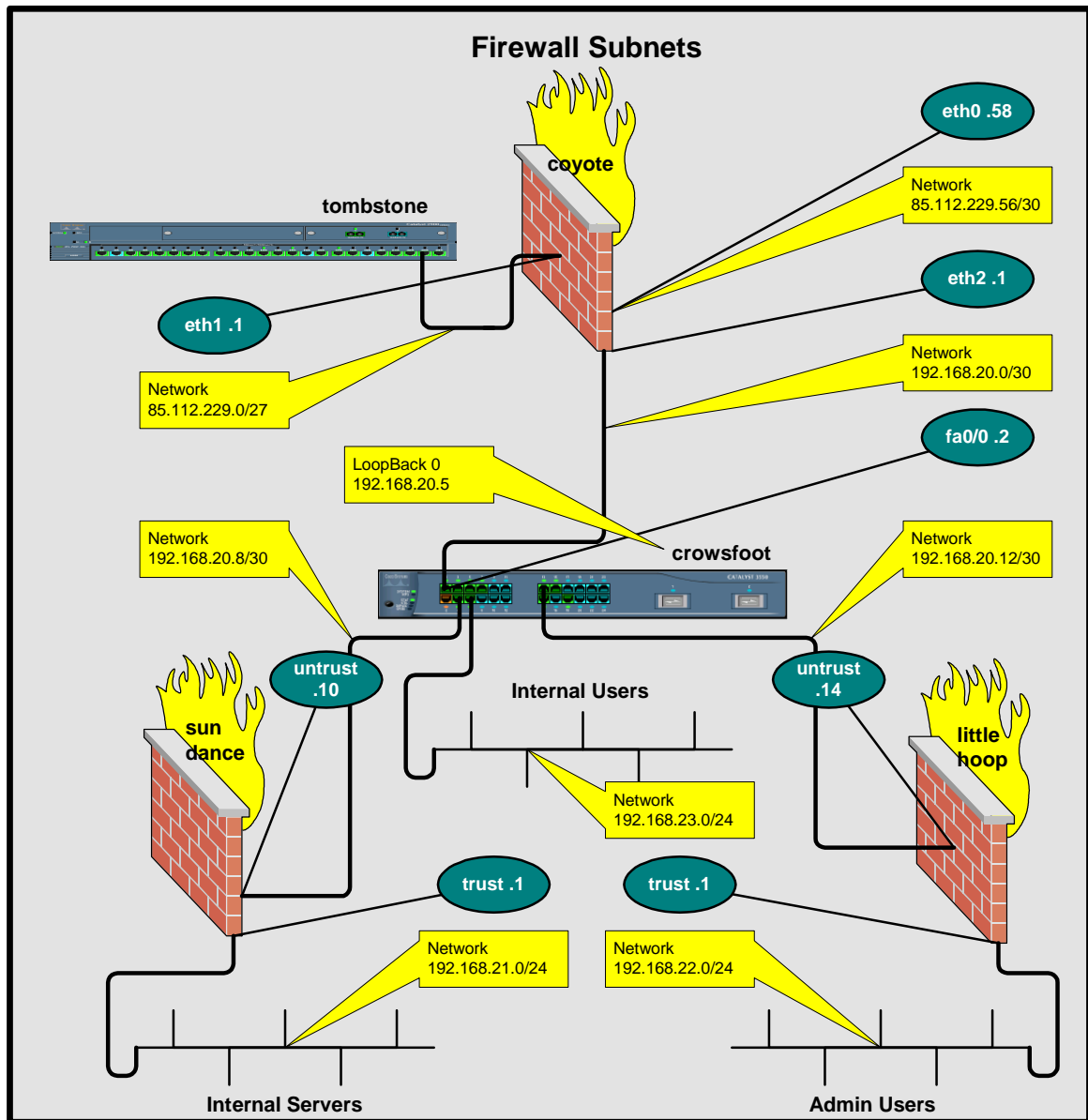
The screenshot shows the Check Point SecurePlatform web interface. On the left is a navigation menu with options: Status, Administration, Networking, Products, Device, and Logout. The main content area is titled 'Device Status' and includes a 'Refresh' button. Below this, there are two sections: 'Device Information' and 'Internet Connections'.

Device Information	
Hostname	coyote
Version and Build	NG with Application Intelligence (R55) 107
Installation Type	VPN-1/FireWall-1 Gateway
Security Policy	coyote
Policy Install Time	Apr 13, 2004 17:27:56
Uptime	12 Days, 3 Hours, 23 Minutes

Internet Connections					
Name	Type	IP Address	Net Mask	Status	Details
eth0	Ethernet	85.112.229.58	255.255.255.252	↑ up	
eth1	Ethernet	85.112.229.1	255.255.255.224	↑ up	
eth2	Ethernet	192.168.20.1	255.255.255.252	↑ up	

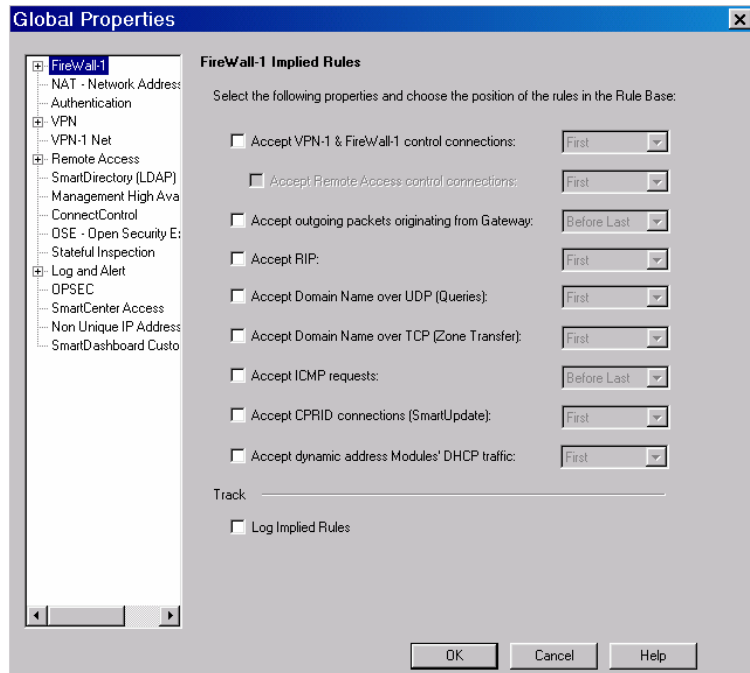
The subnetting of the firewall and internal subnets is shown in the next diagram.



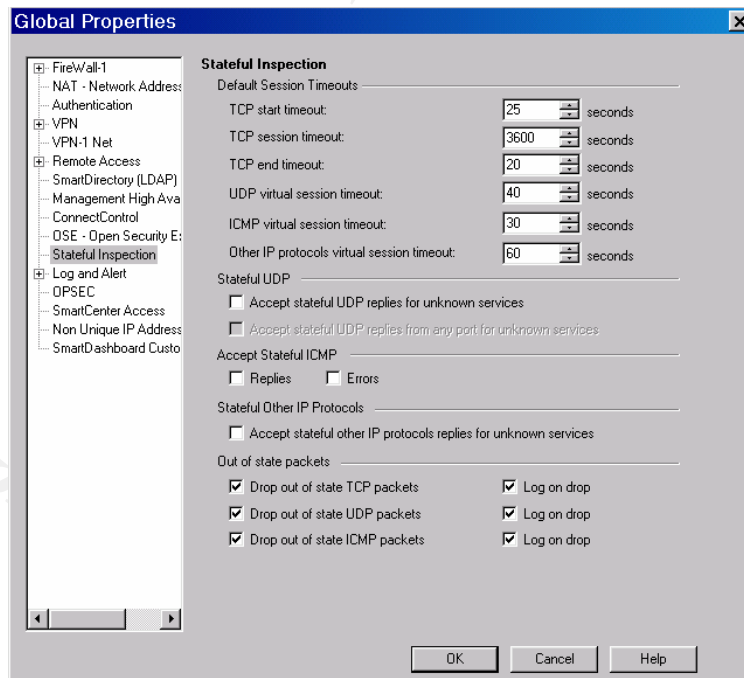
## 2.2 Firewall Security Settings

There are dozens of implied rules that are not shown in the policy viewer, unless you check: view implied rules. We will disable: Accept Remote Access Control Connections. We will control this with the firewall rules and will allow access via SSH from the Internal Admin Network. The figure below illustrates our settings.

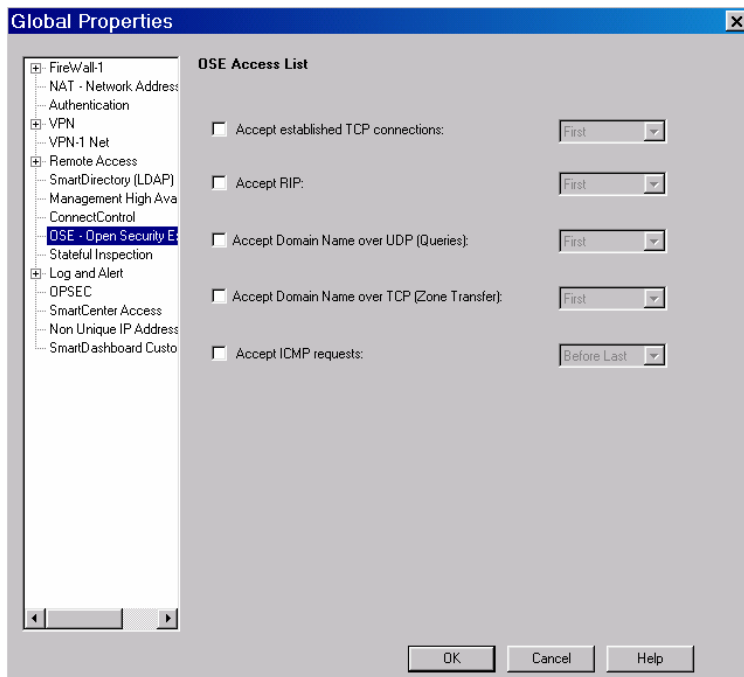
We will rewrite all the rules necessary for managing the firewall and we will uncheck VPN-1 and Firewall-1 Control Connections. This setting disables nearly all of the implied rules. The setting Accept outgoing packets originating from the gateway will also be unchecked. This enables the firewall to send out packets. We will write our own rules for these.



The next settings we will modify are also in the Global Properties, under the Stateful Inspection section. We will leave the default timers alone. We will uncheck Accept Stateful ICMP since we don't care to allow this. Allow Stateful UDP will also be unchecked. The next figure shows the desired settings.



One more place where we will make changes in the Global Properties is the



Open Security Extensions section. The figure to the left shows our settings for this part of the Global Properties. We will uncheck the default of values of: disable Accept Domain Name over UDP (queries) and disable Accept Domain Name over TCP (Zone Transfers) and write our own DNS rules. We will also uncheck the ICMP, RIP and Accept established TCP connections checkboxes. We can do this since we

are not using RIP on the firewall. We will be using static routing. There is no need for ICMP through the firewall so we will not allow any ICMP, if we did we would write our own rules. The same goes for accept established connections. We will let our written rules handle this.

The rules to handle the administration of the firewall and replace the implied rules are shown below. The first rule handles the establishment of VPN's. The FW1\_pslogon\_NG service is required by Secure Client to download its policy. Similarly, Secure Client also uses the FW1\_scv\_keep\_alive, and tunnel\_test services. We will not log these rules just like the implied rules are not logged.

NO.	SOURCE	DESTINATION	SERVICE	ACTION	TRACK	INSTALL ON
<b>Firewall Implied Rules Replacement (Rules 1-2)</b>						
1	* Any	coyote.giac.net	TCP FW1_pslogon_NG UDP FW1_scv_keep_alive UDP IKE UDP tunnel_test TCP FW1_topo	accept	- None	* Policy Targets
2	Mgmt_Station	coyote.giac.net	TCP CPD_amon TCP CPD	accept	- None	* Policy Targets

Rule 2 handles the administration of the firewall. They allow logging to the Management Station and the ability to push policy to the firewall module. The other required rule is the last rule from the end. It allows the firewall module itself to use any service to any destination.

## 2.3 Access Requirements and Primary Firewall Configuration

The access requirements developed in Assignment 1 will become the basis for our security policy. Internal employees need web and ftp access through a proxy. Our Internal Employees will have access to an Exchange Server for their e-mail. The Exchange Server in turn will need access to our Mail-Relay on the screened subnet.

We will put the DNS rules first since they are the most frequently called. Our Internal DNS will also need access to the Internet (via our External DNS) to do recursive lookups of hostnames. The Internal DNS will use UDP and TCP port 53 to communicate with the External DNS. The External DNS will also use these ports. The negated rules insure that the internal nets can't access the external DNS.

Internal and External DNS (Rules 3-5)						
3	Internal_DNS	External_DNS	TCP domain-tcp UDP domain-udp	accept	- None	* Policy Targets
4	External_DNS	<del>GIAC_Internal_Networks</del>	TCP domain-tcp UDP domain-udp	accept	- None	* Policy Targets
5	<del>GIAC_Internal_Networks</del>	External_DNS	TCP domain-tcp UDP domain-udp	accept	- None	* Policy Targets

The rules for the internal employee http access are shown below. The employee's browser is set for the http-ftp proxy. When they attempt to access a web page it is sent to the proxy, which forwards it on. The 1<sup>st</sup> rule the proxy hits is an http resource rule, which sends the request to the SurfControl UFP Server. The UFP server checks the URI resource for acceptable use, URL's that are allowed.

In order for this to work the http\_connection\_method\_proxy setting must be set to true in objects\_5\_0.C. Otherwise, the firewall rejects the connection with a Content Security Access Denied in the logs and the following is displayed in the browser: FW-1 at coyote: Access denied. The http\_connection\_method\_tunnel must also be set to true for the https proxy to work.

Internal Users http and ftp Access (Rules 8-12)						
8	http-ftp-proxy	* Any	HTTP http->SurfControl_Allow_URI	accept	Log	* Policy Targets
9	http-ftp-proxy	* Any	HTTP http->Surf_Control_Block_URI	drop	Log	* Policy Targets
10	http-ftp-proxy	* Any	HTTP http->Coyote_http_URI_Proxy	accept	Log	* Policy Targets
11	http-ftp-proxy	* Any	HTTP https->Coyote_https_URI_Proxy	accept	Log	* Policy Targets
12	http-ftp-proxy	* Any	FTP ftp->Coyote_ftp_Proxy	accept	Log	* Policy Targets

The http-proxy's IP address is translated to the gateways external IP address as shown below.

NO.	ORIGINAL PACKET			TRANSLATED PACKET			INSTALL ON
	SOURCE	DESTINATION	SERVICE	SOURCE	DESTINATION	SERVICE	
6	GIAC_Internal_Networks	* Any	* Any	coyote.giac.net	Original	Original	* Policy Targets

The internal users also need e-mail access. They will use an Exchange client for their access. Rule 18 allows the Exchange server to access the Mail\_Relay on the screened subnet. Rule 19 allows access to the Mail\_Relay from Postini. These rules are shown below.

E-mail Access (Rules 19-20)						
19	Exchange Mail_Relay	Exchange Mail_Relay	TCP smtp	accept	Log	* Policy Targets
20	Mail_Relay Postini_Mail_Services	Postini_Mail_Services Mail_Relay	TCP smtp	accept	Log	* Policy Targets

Customers will have access to the web-server via a reverse proxy. That access is shown below. They also need to be able to resolve the URL to an IP address, that access is covered by the DNS rules above. The web-servers are allowed to connect to the backend database via the Middle\_Ware server as shown in rule 7.

Customer Access (Rules 5-6)						
5	* Any	Reverse_http_Proxy	TCP https TCP http	accept	Log	* Policy Targets
6	GIAC_Web_Servers	Middle_Ware	TCP winframe	accept	Log	* Policy Targets

The rest of the non-VPN rules are shown below. A complete listing of the firewall rules is in the Appendix. Rule 26 allows RADIUS authentication from the firewall itself and the WAP. Secure Client uses this rule when it attempts to establish its VPN. The WAP uses this rule also to authenticate the Mobile Users laptops when they return to the site. This rule is also used by the border router when an Internal Admin user establishes a SSH session over a VPN tunnel to the router.

Misc and Admin Access (Rules 26-30)						
26	coyote_eth2 wAP	Internal_RADIUS	UDP NEW-RADIUS UDP NEW-RADIUS-ACCT	accept	Log	* Policy Targets
27	GIAC_Internal_Admin	GIAC_Screened_Subnet coyote_eth2 wAP	TCP ssh	accept	Log	* Policy Targets
28	SurfControl http-ftp-proxy	www.trendmicro.com www.surfcontrol.com	TCP http	accept	Log	* Policy Targets
29	coyote.giac.net	* Any	* Any	accept	Log	* Policy Targets
30	* Any	* Any	* Any	drop	Log	* Policy Targets

Rule 27 allows admin access to the firewall and to hosts on the screened subnet. There is also a translation rule for this as shown below. Rule 1 is

applied for this access so the source addresses are not translated. We could have chosen to only translate the http-ftp proxy, since it is the only host that uses the translation, but if we do it this way then any outbound traffic will be translated. So if any traffic makes it by the border router it still has to go through a translation process to make it to the internal networks.

NO.	ORIGINAL PACKET			TRANSLATED PACKET			INSTALL ON
	SOURCE	DESTINATION	SERVICE	SOURCE	DESTINATION	SERVICE	
1	GIAC_Internal_Networks	GIAC_Screened_Subnet	* Any	Original	Original	Original	* Policy Targets
2	GIAC_Internal_Admin	WAP	* Any	Original	Original	Original	* Policy Targets
7	GIAC_Internal_Networks	* Any	* Any	coyote.giac.net	Original	Original	* Policy Targets

Rule 28 allows SurfControl and the proxy update their database and signatures. Rule 29 is from the implied rules. We allow the firewall itself to establish any connection to any host. Finally Rule 30 drops and logs all other traffic.

### 2.3 Secondary Firewall Configuration

There will be two internal firewalls one for the Internal Admin network and one for the Internal Server Network. These firewalls will be Netscreen-5GT running ScreenOS 5.0.0r4.1 (Firewall+VPN). The trusted interface will be the internal subnet. The untrust interface will be closest to the firewall coyote.

The policy for the Internal Admin network will allow outbound SNMP and DNS queries, SSH, Web and E-mail access for the Admin Hosts. We will log all the SSH and denied connections. It will also allow the Firewall Mgmt Station access the firewall. The trust to untrust policy is shown below.

From Trust To Untrust, total policy: 7

ID	Source	Destination	Service	Action	Options	Configure
1	Admin_Workstations	Internal_DNS	DNS			<a href="#">Edit</a> <a href="#">Clone</a> <a href="#">Remove</a>
0	Admin_Workstations	http/ftp_Proxy	FTP HTTP HTTPS			<a href="#">Edit</a> <a href="#">Clone</a> <a href="#">Remove</a>
4	Admin_Workstations	Exchange	DCE_RPC_Group			<a href="#">Edit</a> <a href="#">Clone</a> <a href="#">Remove</a>
10	FW1_Mgmt	coyote-eth2	CPD CPD_amon			<a href="#">Edit</a> <a href="#">Clone</a> <a href="#">Remove</a>
2	Admin_Workstations	coyote-eth2 WAP wineta-lo0 GIAC_Internal_Users GIAC_Screened_Subnet GIAC_Server_Subnet	SSH			<a href="#">Edit</a> <a href="#">Clone</a> <a href="#">Remove</a>



3	SNMP_Manager	coyote-eth2 WAP wineta-lo0 GIAC_Internal_Users GIAC_Screened_Subnet GIAC_Server_Subnet	SNMP			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
9	Any	Any	ANY			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>

The policy for the Internal Admin firewall will allow inbound Syslog, SNMP traps and Radius queries to the respective servers located there. The policy will also allow logging from the CheckPoint firewall to the Management Station. The policy will log all of the RADIUS and dropped connections. The untrust to trust policy is shown below.

From Untrust To Trust, total policy: 5

ID	Source	Destination	Service	Action	Options	Configure		
5	http/ftp_Proxy WAP wineta-lo0 GIAC_Screened_Subnet GIAC_Server_Subnet	Syslog_Server	SYSLOG			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
11	coyote-eth2	FW1_Mgmt	FW1_log			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
6	WAP wineta-lo0 GIAC_Screened_Subnet GIAC_Server_Subnet	Admin_Workstations	SNMP_TRAP			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
7	coyote-eth2 WAP wineta-lo0 GIAC_Screened_Subnet GIAC_Server_Subnet	Admin_Workstations	RADIUS_Group			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
8	Any	Any	ANY			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>

The policy on Sundance (the Internal Server subnet's firewall) will allow, incoming DNS queries to the DNS server. It will allow the Middleware Sever to communicate with the backend database. The policy also allows The Terminal Services Server and two internal subnets to access Exchange via Microsoft's DCE\_RPC protocol. The Mail-Relay is allowed access to Exchange for inbound mail. Terminal Services Server and Internal users are allowed access to applications and file and print servers. The untrust to trust policy is shown below.

From Untrust To Trust, total policy: 9

ID	Source	Destination	Service	Action	Options	Configure		
12	Terminal_Services_Server GIAC_Admin_Users GIAC_Internal_Users	Internal_DNS	DNS			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>



10	MiddleWare_Server	mysql_DB	mysql			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
20	Mail_Relay	Exchange	MAIL			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
11	Terminal_Services_Server GIAC_Admin_Users GIAC_Internal_Users	Exchange	DCE_RPC_Group			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
15	Terminal_Services_Server GIAC_Internal_Users	File/Print_Server	NBT			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
13	GIAC_Internal_Users	HR_Server	HR			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
14	GIAC_Internal_Users	Finance_Server	Finance			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
18	GIAC_Admin_Users	GIAC_Server_Subnet	microsoft-ds NBT SSH			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
8	Any	Any	ANY			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>

The trust to untrust policy is shown below. This policy allows inbound DNS, mail and mySQL to the proper server. It is followed by the deny any which we log.

From Trust To Untrust, total policy: 4

ID	Source	Destination	Service	Action	Options	Configure		
17	Internal_DNS	External_DNS	DNS			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
16	Exchange	Mail_Relay	MAIL			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
19	mysql_DB	hotsite_mysql	mysql			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>
9	Any	Any	ANY			<a href="#">Edit</a>	<a href="#">Clone</a>	<a href="#">Remove</a>

### 3 VPN's

#### 3.1 Component Description

Our sites will use CheckPoint FW-1's integrated VPN solution VPN-1 Pro. Its purpose is to provide secure access over an untrusted medium. It accomplishes this by encrypting all data transported over the medium. Another of its purposes is to reduce costs through the reduction of point-to-point serial lines and dial-up access.

#### 3.2 Border Router VPN

One of the VPN's we will setup is to the border router. This VPN is for internal admin use and monitoring of the external device. Since they will always

be devices external to the firewall that we need to administer. We thought this was a good way to maintain control of these devices. It allows a secure channel to the device and we don't have to open a lot of ports on the firewall in order to provide this access. Our VPN rules are shown in traditional mode. The rules to provide this access are shown below.

Border Router VPN (Rules 21-24)						
21	wineta_lo0	GIAC_Internal_Admin	UDP syslog UDP snmp-trap	Encrypt	Log	* Policy Targets
22	wineta_lo0	Internal_RADIUS	UDP NEW-RADIUS UDP NEW-RADIUS-ACCT	Encrypt	Log	* Policy Targets
23	GIAC_Internal_Admin	wineta_lo0	TCP ssh UDP snmp	Encrypt	Log	* Policy Targets
24	crowsfoot-fa0-13	wineta_lo0	UDP ntp-udp	Encrypt	Log	* Policy Targets

Rule 21 allows wineta, (the border router) to send Syslog packets and SNMP-traps to the Internal admin network. Rule 22 allows access to the Internal\_Radius server to authenticate SSH connections. Rule 23 allows SSH access to the device. It also allows for an SNMP Manager to poll the device using SNMP. Rule 24 allows crowsfoot (our internal router) to synchronize its clock using NTP to wineta. All the other internal devices may then synchronize off of crowsfoot.

NO.	ORIGINAL PACKET			TRANSLATED PACKET			INSTALL ON
	SOURCE	DESTINATION	SERVICE	SOURCE	DESTINATION	SERVICE	
3	GIAC_Internal_Admin	wineta_lo0	TCP ssh	Original	Original	Original	* Policy Targets
4	GIAC_Internal_Admin	wineta_lo0	UDP snmp	Original	Original	Original	* Policy Targets
5	crowsfoot-fa0-13	wineta_lo0	UDP ntp-udp	Original	Original	Original	* Policy Targets

Some address translation rules are also necessary, as shown above. These rules were added manually. Rules 3 through 5 do not perform translation on the packets bound for the border router. Our access-list for the VPN on the router is based on the internal address space. We do not wish these packets to be NAT'ed. Also the VTY ACL is based on internal address space, as is the SNMP ACL. The complete configuration of the Border Router VPN is in the Tutorial.

### 3.3 Mobile and TeleWorkers VPN

The SecureClient VPN solution consists of a client component (SecureClient), a Management Server and one or more VPN-enabled Firewall Modules. The SecureClient application obtains information about the site through a topology download. The next step in the process is the initiation of a VPN connection. The user must then authenticate to the site. Upon successful authentication and exchange of keys and encryption parameters a VPN tunnel between the site and client is brought up.

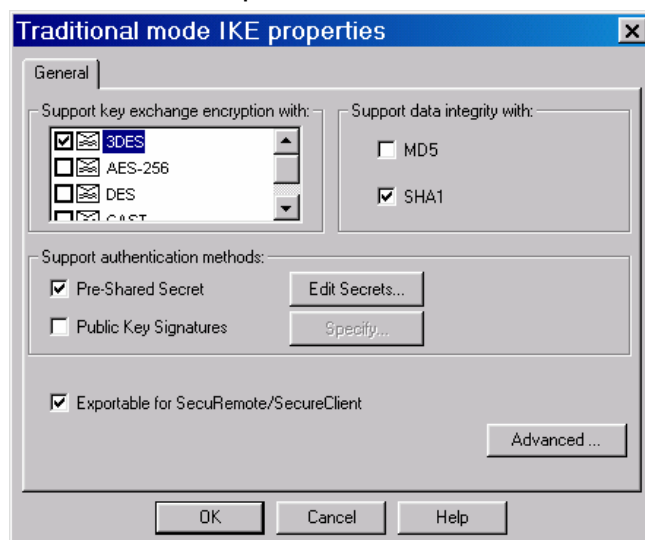
Once the tunnel is up the client logs on to the Policy Server and retrieves his/her policy. It is this policy that is enforced on the client's desktop firewall.



The configuration for the SecureClient VPN is specified in the Firewall Gateway Object. In the VPN section under traditional mode configuration, that dialog box is shown to the right. It shows the Phase 1 parameters.

The check box Exportable must be checked in order to enable the topology download.

Note: The pre-shared Secret is used for our site-to-site VPN's.



Our policy is shown below. The first group is the inbound rules. That is inbound from the point of view of the client. When the client is connected to the Internal network via the VPN, the Rule 2 will drop any

traffic destined for that host. The 1st rule allows the port MS-Terminal-Services to the host from the Terminal Services Server when the host has a VPN connection. The third rule drops any externally initiated traffic when the host is not connected to the VPN.

Inbound Rules					
NO.	SOURCE	DESKTOP	SERVICE	ACTION	TRACK
1	Terminal_Services_Server	MobileUser@Any	TCP MS-Terminal-Services	Accept	Log
2	Any	MobileUser@Any	Any	Block	Log
3	Any	All Users@Any	Any	Block	Log

The determination of what rule applies comes from the Desktop column.

MobileUser@Any indicates the user is logged on and a tunnel is up.

All Users@Any indicates the user is not logged and is not connected to an Internal LAN.

All the rules that contain AllUsers are considered the default Desktop security policy and are loaded as soon as the SecureClient services are started. When you are not logged into the Policy server SecureClient reverts to the default policy.

The outbound rules are next. For each outbound rule that has traffic to an internal host, there is a corresponding incoming rule on the VPN Gateway. The first rule has a source of MobileUser@Any, indicating that the user must be connected to the VPN for this rule to apply. Rule 2 blocks any other outbound traffic when the user is connected. Finally Rule 3 allows outbound connections when the client is not connected.

Outbound Rules					
NO.	DESKTOP	DESTINATION	SERVICE	ACTION	TRACK
4	MobileUser@Any	Terminal_Services_Server	TCP MS-Terminal-Services	Encrypt	Log
5	MobileUser@Any	Any	Any	Block	Log
6	All Users@Any	Any	Any	Accept	Log

The corresponding inbound rule that permits this traffic is rule 12 as shown below. Rules 13 through 16 are shown here as they permit access from their Terminal Services Server to the internal Networks. We allow Terminal Services access to the http-proxy for web browsing. DNS access is also allowed for name

resolution. Access is allowed to the Exchange server via the Exchange Client ports. Finally we allow access to Internal Services for file, print and application access.

Mobile and Teleworker Access (Rules 13-17)						
13	MobileUser@Any	Terminal_Services_Server	TCP MS-Terminal-Services	Client Encrypt	Log	Policy Targets
14	Terminal_Services_Server	http-ftp-proxy	TCP http TCP https TCP ftp	accept	Log	Policy Targets
15	Terminal_Services_Server	Internal_Servers	TCP NBT TCP microsoft-ds	accept	Log	Policy Targets
16	Terminal_Services_Server	Exchange	TCP dce-rpc TCP rpc1189 TCP rpc1314	accept	Log	Policy Targets
17	Terminal_Services_Server	Internal_DNS	TCP domain-tcp UDP domain-udp	accept	Log	Policy Targets

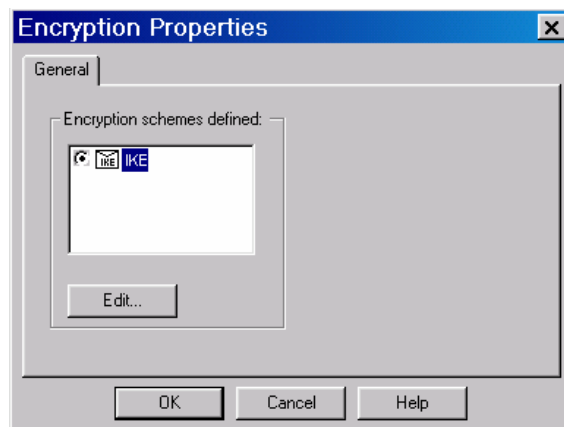
### 3.4 Partners and Suppliers VPN

Access for one representative Partner or Supplier is shown in the rule below. Additional VPN's can be created following this model. The partners or suppliers are allowed to access the ftp-server on the screened subnet for ftp use only. They will not need a translation rule; since they are on the screened subnet there will be no translation performed.

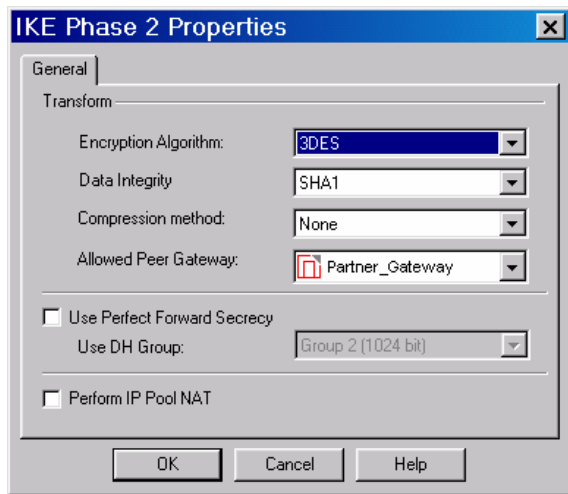
Partners and Suppliers Access (Rule 18)						
18	Partners_and_Suppliers	ftp_Server	TCP ftp	Encrypt	Log	Policy Targets

The VPN setup for Phase 1 will be the same as the SecureClient configuration. Clicking on the Encrypt icon in rule 18 will bring up the Encryption Properties Dialog box shown to the right.

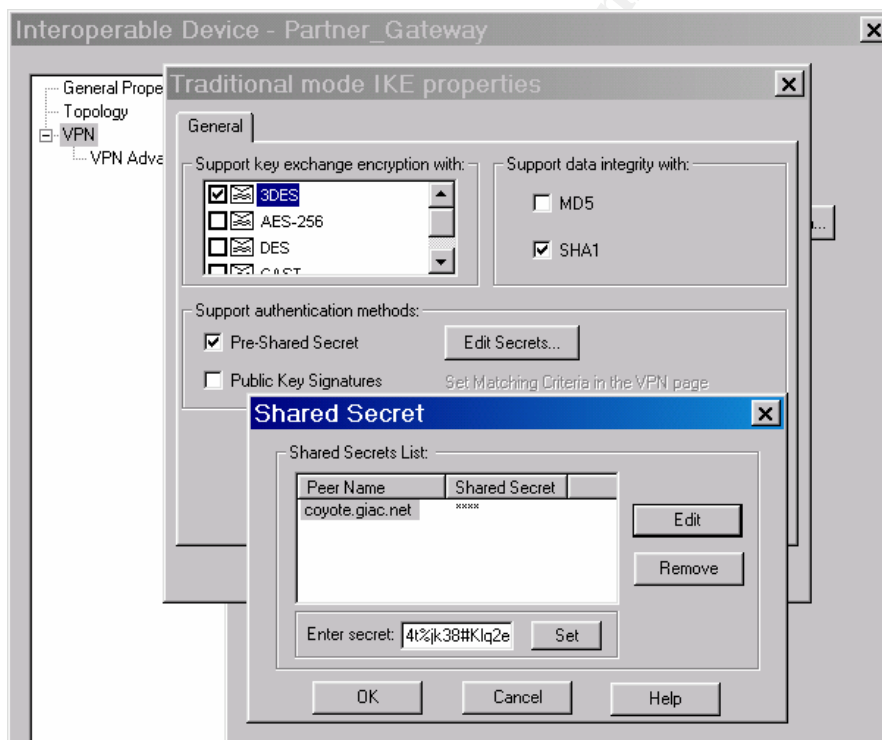
Clicking on the edit button displays the Phase 2 IKE properties. The Phase 2 transform parameters for the connection may be selected from here. We have chosen 3DES for the Encryption Algorithm, SHA1 for Data Integrity and the allowed peer gateway to be coyote.



We will use Group 2 Diffie Hellman and we will not use Perfect Forward Secrecy.



A network object will have to be created for the Partner Gateway. The Partner Gateway is a Netscreen Firewall, so we will create the object as an interoperable device. We need to define its topology and encryption domain. The topology consists of its trust and untrust interfaces. The encryption domain will be it's internal networks. We will also define its Phase 1 properties and the pre-shared secret here. This is shown in the diagram below.



### 3.5 Slave Database VPN

The setup for the slave database VPN will be similar to the partners and suppliers VPN. However the endpoint or allowed peer gateway will be our hotsite, which is a CheckPoint firewall. The setup for this VPN is identical to the partner and supplier VPN. The rule is shown below.



This does need a translation rule since the source is from the Internal Servers subnet.

NO.	ORIGINAL PACKET			TRANSLATED PACKET			INSTALL ON
	SOURCE	DESTINATION	SERVICE	SOURCE	DESTINATION	SERVICE	
6	 Main_Site_DB	 Hot_Site_DB	 mysql	 Original	 Original	 Original	 Policy Targets
7	 GIAC_Internal_Networks	 Any	 Any	 coyote.giac.net	 Original	 Original	 Policy Targets

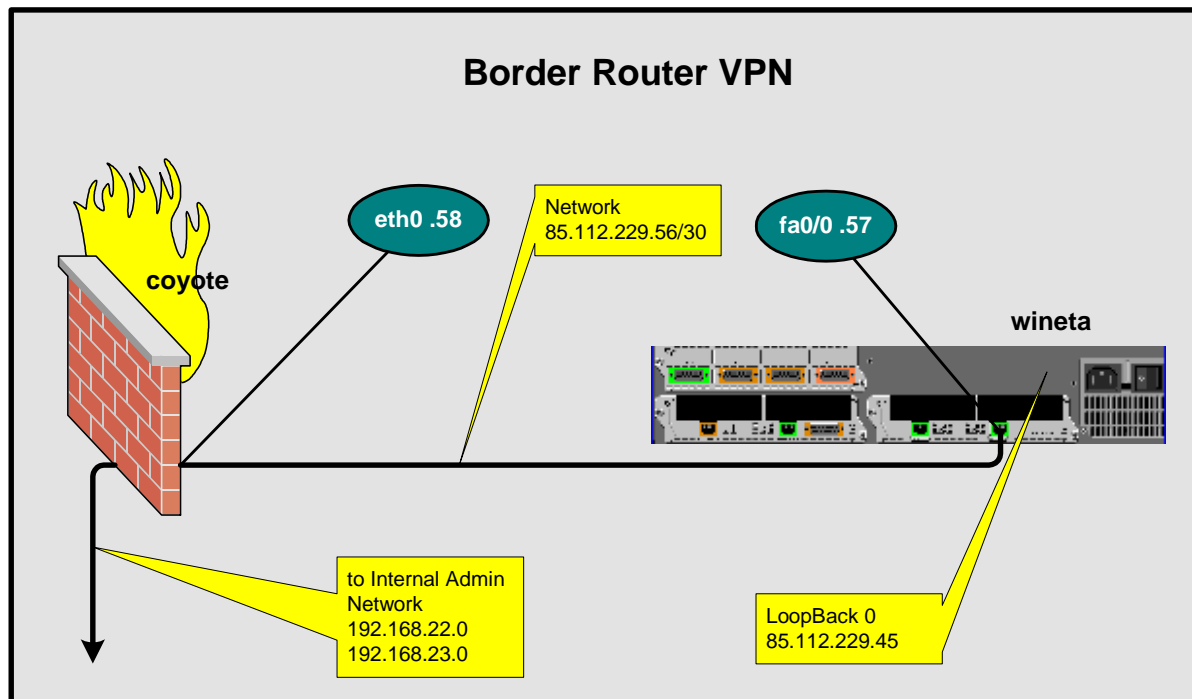
## 4 CheckPoint FW-1 NG to Cisco 3640 VPN Tutorial

### 4.1 Introduction

This section shows how to establish an IPSEC tunnel between a Cisco 3640 router and CheckPoint Firewall-1 NG. The IPSEC tunnel will be able to use multiple internal subnets. The Firewall is Check Point SecurePlatform NG with Application Intelligence (R55) Build 107. The 3640 is running Cisco IOS Software (C3640-IK9S-M), Version 12.2(11)T, RELEASE SOFTWARE (fc1).

In my research for this I came across another paper that deserves mention, [Configuring an IPSEC Tunnel between a Cisco Router and NG<sup>8</sup>](#). This document is very similar in content to this paper. However that paper only discusses one internal subnet for the encryption domain and also includes NAT whereas we do not. This paper will cover the setup of multiple subnets, since I need one subnet for the Internal Admin and one subnet for devices to do NTP. The two devices subnetting is shown below. The encryption domains will be the Internal Networks behind the firewall and the Loopback0 interface of the router.





## 4.2 Router Configuration

The configuration of the IPSEC tunnel has two parts<sup>9</sup>: configuring the IKE policy and configuring the IPSEC policy, including the access list, interface definition, transform set and crypto map. We begin with the definition of the IKE parameters. To define an IKE policy we use the command **crypto isakmp policy 1** in global configuration mode. This puts us into config-isakmp mode where we can define, the encryption algorithm, the hash algorithm, the authentication method, the Diffie-Hellman group identifier, and the security association's lifetime. This is shown below.

```
wineta(config)#crypto isakmp policy 1
wineta(config-isakmp)#encr 3des
wineta(config-isakmp)# hash sha
wineta(config-isakmp)# lifetime 86400
wineta(config-isakmp)# authentication pre-share
wineta(config-isakmp)# group 2
wineta(config-isakmp)#
```

To view all the existing IKE policies we use the command: **sh crypto isakmp policy**.

```
wineta#sh crypto isakmp policy
Protection suite of priority 1
  encryption algorithm: Three key triple DES
  hash algorithm:      Secure Hash Standard
  authentication method: Pre-Shared Key
  Diffie-Hellman group: #2 (1024 bit)
  lifetime:           86400 seconds, no volume limit
```



**Default protection suite**

encryption algorithm: DES - Data Encryption Standard (56 bit keys).  
hash algorithm: Secure Hash Standard  
authentication method: Rivest-Shamir-Adleman Signature  
Diffie-Hellman group: #1 (768 bit)  
lifetime: 86400 seconds, no volume limit

wineta#

Now we define the pre-shared key. It was already defined so we got the following message. The command to view the keys is also shown.

```
wineta(config)#crypto isakmp key Ab134c$Lmn%2 address 85.112.229.58  
A pre-shared key for address mask 85.112.229.58 255.255.255.255 already exists!
```

```
wineta#sh crypto is key  
Hostname/Address Preshared Key  
85.112.229.58 Ab134c$Lmn%2  
wineta#
```

To configure the IPSEC crypto map policy we first have to define an access list for the traffic we want to protect. Basically any ip traffic coming from the two internal subnets bound for the Loopback0 interface.

```
wineta(config)#access-list 101 permit esp host 85.112.229.58 host 85.112.229.57  
wineta(config)#access-list 101 permit ip 192.168.22.0 0.0.0.255 host 85.112.229.45  
wineta(config)#access-list 101 permit ip host 85.112.229.45 192.168.22.0 0.0.0.255  
wineta(config)#access-list 101 permit ip 192.168.23.0 0.0.0.255 host 85.112.229.45  
wineta(config)#access-list 101 permit ip host 85.112.229.45 192.168.23.0 0.0.0.255  
wineta(config)#access-list 101 deny ip any any log  
wineta(config)#access-list 101 deny esp any any log
```

Now we need to define the transform set that will delineate how the traffic is protected. That is which transform set to use to encapsulate the traffic. The command to view the transform sets is also shown.

```
wineta(config)#crypto ipsec transform-set vpnset esp-3des esp-sha-hmac  
wineta(cfg-crypto-trans)#
```

```
wineta#sh crypto ipsec transform-set  
Transform set vpnset: { esp-3des esp-sha-hmac }  
will negotiate = { Tunnel, },
```

We are now ready to define the crypto map. The crypto map joins the IPSEC access list and the transform set. It also specifies that the traffic is sent to the remote IPSEC peer. The command `crypto map ssh_to_loop ipsec-isakmp` issued from global configuration mode puts us into crypto-map command mode. From this mode we can specify the match address, the peer, and the transform set. The match address is an extended access list.

```
wineta(config)#crypto map ssh_to_loop 1 ipsec-isakmp  
wineta(config-crypto-map)# set peer 85.112.229.58  
wineta(config-crypto-map)# set transform-set vpnset
```

```
wineta(config-crypto-map)# match address 101  
wineta(config-crypto-map)#
```

Whew, we are now ready to apply the crypto map to an interface. We get into interface command mode and apply the crypto map. Voila, the results are shown with the command **sh crypto map**.

```
wineta(config)#interface FastEthernet0/1  
wineta(config-if)#description Internal connection to Corporate Firewall  
wineta(config-if)#ip address 85.112.229.57 255.255.255.252  
wineta(config-if)#crypto map ssh_to_loop
```

```
wineta#sh crypto map  
Crypto Map: "ssh_to_loop" idb: FastEthernet0/1 local address: 85.112.229.57
```

```
Crypto Map "ssh_to_loop" 1 ipsec-isakmp  
Peer = 85.112.229.58  
Extended IP access list 101  
  access-list 101 permit esp host 85.112.229.58 host 85.112.229.57  
  access-list 101 permit ip 192.168.22.0 0.0.0.255 host 85.112.229.45  
  access-list 101 permit ip host 85.112.229.45 192.168.22.0 0.0.0.255  
  access-list 101 permit ip 192.168.23.0 0.0.0.255 host 85.112.229.45  
  access-list 101 permit ip host 85.112.229.45 192.168.23.0 0.0.0.255  
  access-list 101 deny ip any any  
  access-list 101 deny esp any any  
Current peer: 85.112.229.58  
Security association lifetime: 4608000 kilobytes/3600 seconds  
PFS (Y/N): N  
Transform sets={ vpnset, }  
Interfaces using crypto map ssh_to_loop:  
  FastEthernet0/1
```

Some of the debugging commands I used for troubleshooting were as follows. This command will show the Phase 1 and Phase 2 negotiations and any problems that occurred.

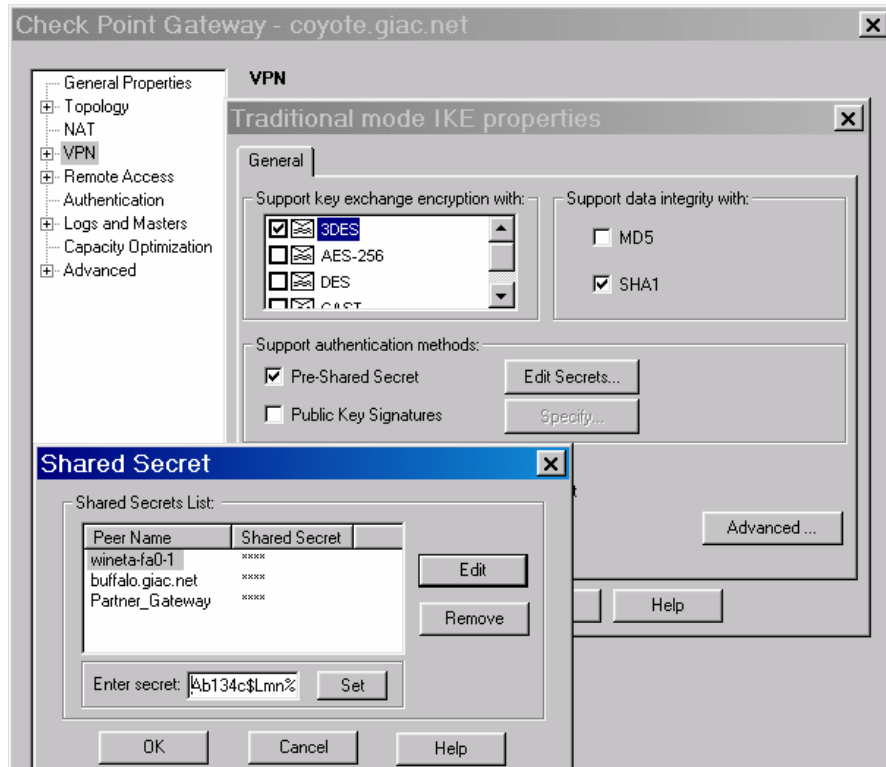
```
wineta#term mon  
wineta#debug crypto isakmp  
Crypto ISAKMP debugging is on  
wineta#
```

### 4.3 Firewall Configuration

The configuration of the VPN on the CheckPoint firewall (coyote) consists of creating objects applying configurations to those objects, and using them in rules. We need to

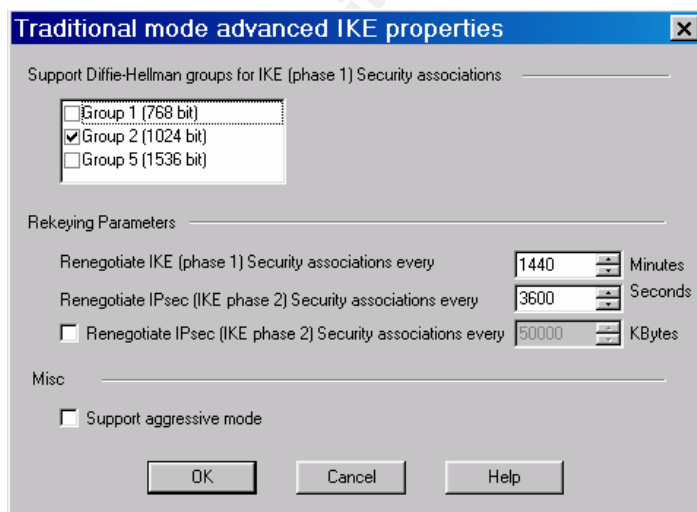
create network objects for the firewall's encryption domain and the router's encryption domain.

These should be created first so we can use them in the next objects we have to create, network objects for the firewall and the router. . Once these objects are created we then define the



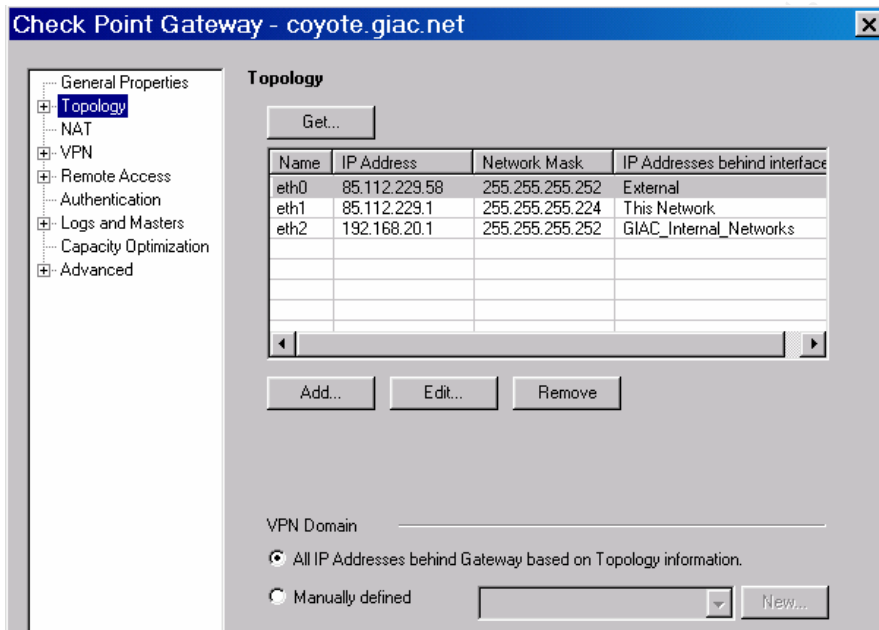
IKE parameters for each. The final step is to create the encryption rule and define its IKE parameters. Also we must add any necessary NAT rules. We

have already created the firewall object. Its relevant parameters for this exercise are shown in the diagram. As we can see its encryption algorithm is 3DES with a hash of SHA1. The shared secret corresponds to the pre-shared key configured on the router. Under the advanced IKE properties tab we select Group 2 Diffie-Hellman. We don't support aggressive mode and we leave the rekeying

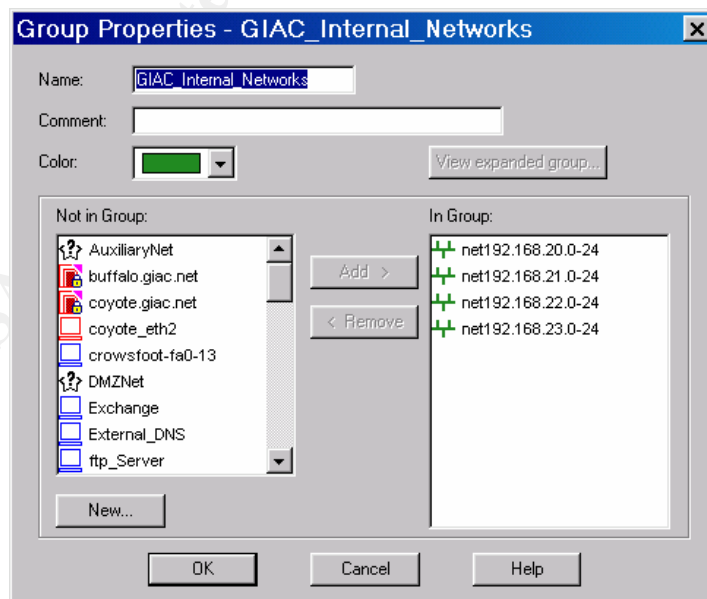


parameters at their defaults as shown above.

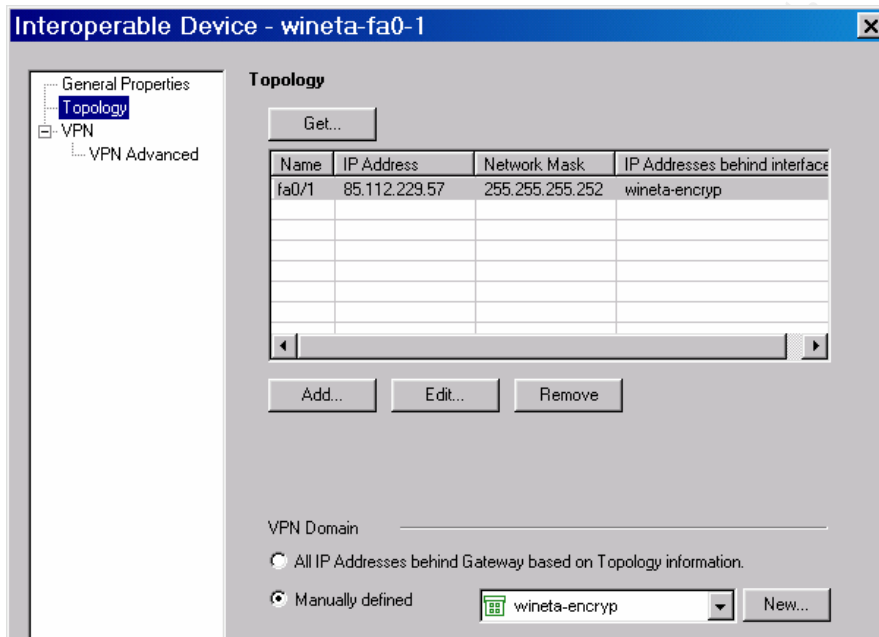
The encryption domain parameters for coyote are found on the topology tab of its object. Its VPN domain is all IP addresses behind the Gateway based on Topology Information. This includes the screened subnet and the group GIAC\_Internal\_Networks as shown in the Diagram below.



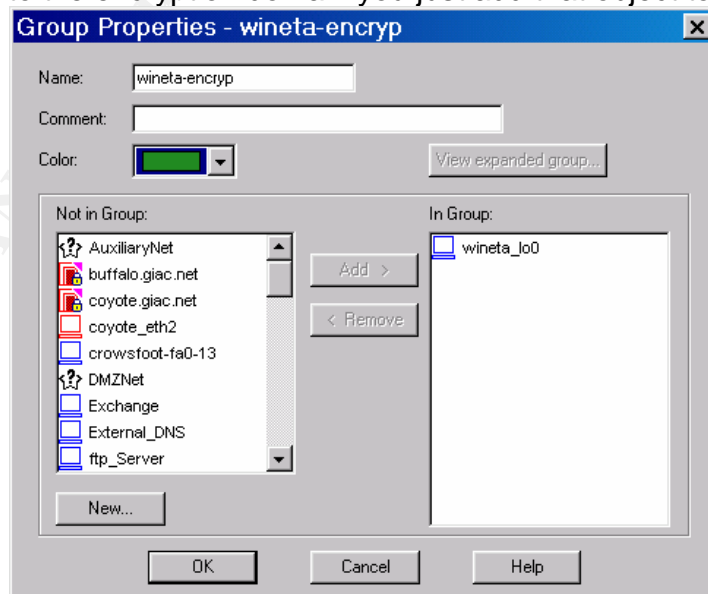
GIAC\_Internal\_Networks consists of four class C networks as shown in the dialog box.



Similarly the encryption domain for the Cisco 3640 (wineta) is also found on the topology tab of its object. This is the next object we define. Wineta-fa0-1 will be defined as an interoperable device. This object must be created so that the firewall knows where to send the encrypted packets. It is analogous to the access-list and match list parameters in the Cisco router configuration. Its topology tab is shown below.



Its VPN domain consists of the group wineta-encryp. Wineta-encryp consists of one object wineta-lo0, which is the loopback0 interface of the router. I find it easier to create groups for encryption domains rather than add single objects. If you need to add to the encryption domain you just add that object to the group.



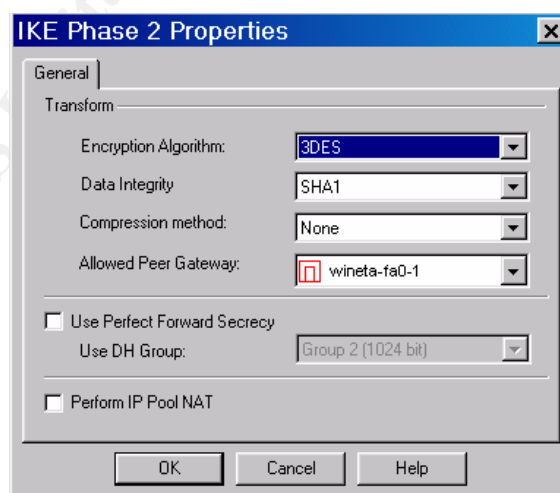
Now that we have our encryption domains and gateways defined we can create the rules that use them. We will also define the IKE properties of these rules by clicking on the encrypt icon of the rule. The rules are shown below.

Border Router VPN (Rules 19-22)						
19	wineta_lo0	GIAC_Internal_Admin	UDP syslog UDP snmp-trap	Encrypt	Log	Policy Targets
20	wineta_lo0	Internal_RADIUS	UDP NEW-RADIUS UDP NEW-RADIUS-ACCT	Encrypt	Log	Policy Targets
21	GIAC_Internal_Admin	wineta_lo0	TCP ssh UDP snmp	Encrypt	Log	Policy Targets
22	crowsfoot-fa0-13	wineta_lo0	UDP ntp-udp	Encrypt	Log	Policy Targets

The translation rules are shown below. If we don't disable the translation we get the error IKE: Quick Mode Received Notification from Peer: no proposal chosen. Followed by encryption failure: Error occurred. And finally: encryption fail reason: Packet is dropped because there is no valid SA - please refer to solution sk19423 in SecureKnowledge Database for more information. This occurs because if it is translated it does not fall into the expected encryption domain of the router, so no Phase 2 proposal can be chosen.

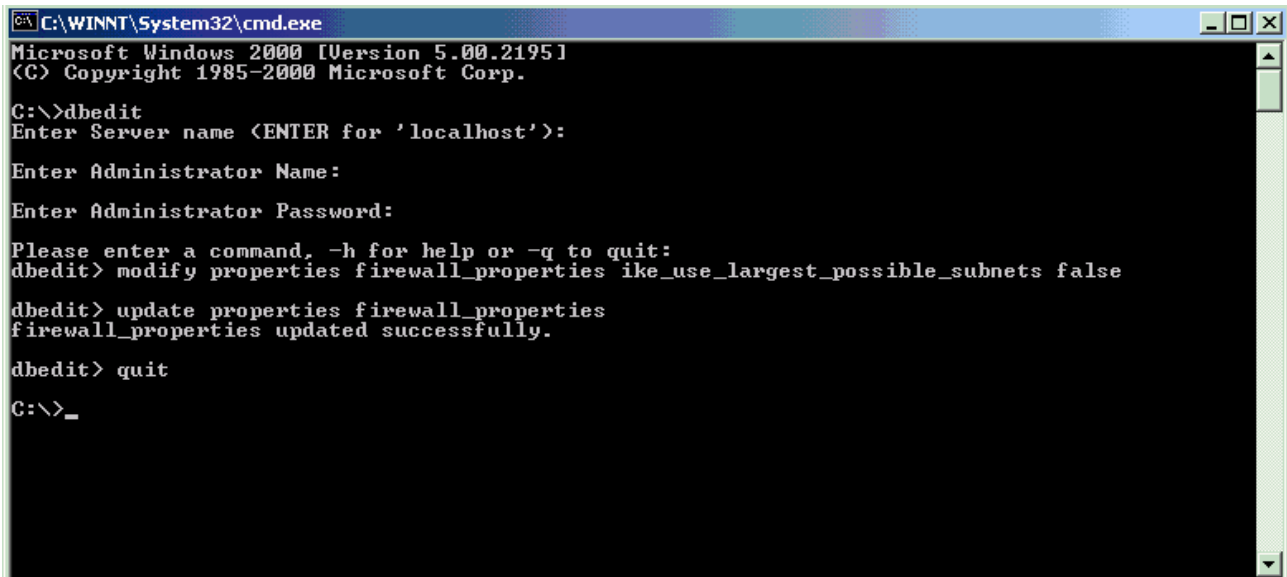
NO.	ORIGINAL PACKET			TRANSLATED PACKET			INSTALL ON
	SOURCE	DESTINATION	SERVICE	SOURCE	DESTINATION	SERVICE	
3	GIAC_Internal_Admin	wineta_lo0	TCP ssh	Original	Original	Original	Policy Targets
4	GIAC_Internal_Admin	wineta_lo0	UDP snmp	Original	Original	Original	Policy Targets
5	crowsfoot-fa0-13	wineta_lo0	UDP ntp-udp	Original	Original	Original	Policy Targets

The IKE properties for all four rules are shown below.



One more thing has to be done in order for the Phase 2 negotiation to succeed. We have to edit a property in objects\_5\_0.c file. This property is

needed in order to support multiple subnets in the encryption domain behind the firewall. We use dbedit to do this on the management station. First close all GUI clients. Then open a DOS window on the management station.



```
C:\WINNT\System32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>dbedit
Enter Server name (ENTER for 'localhost'):
Enter Administrator Name:
Enter Administrator Password:
Please enter a command, -h for help or -q to quit:
dbedit> modify properties firewall_properties ike_use_largest_possible_subnets false
dbedit> update properties firewall_properties
firewall_properties updated successfully.
dbedit> quit
C:\>_
```

Now push the policy, initiate some traffic to bring up a VPN tunnel and voila.

© SANS Institute 2004, Author

# Assignment 3 Verify the Firewall Policy

## 1 Validation Plan

### 1.1 Technical Approach

Our assessment of the firewall is to determine if the perimeter device is operating according to our expectations. The assessment will consist of an internal assessment only. In the assessment we want to test if the policy on the firewall is properly implemented. To perform the assessment we will need to determine if traffic is traversing the firewall correctly.

Since this is an Internal Assessment we have detailed knowledge of the firewall policy and we want to verify if the allowed network traffic complies with the intention of the rules. We will obtain management approval before proceeding. We will also notify all involved parties of the assessment. Our approach will be as follows:

Check that only ports that are allowed by the policy are open.

Check that only allowed stations can traverse these open ports.

Verify that the firewall management station is the only host allowed to make control connections to the firewall.

Verify that only authorized users are allowed to make VPN connections.

We will use the two techniques below to accomplish these objectives.

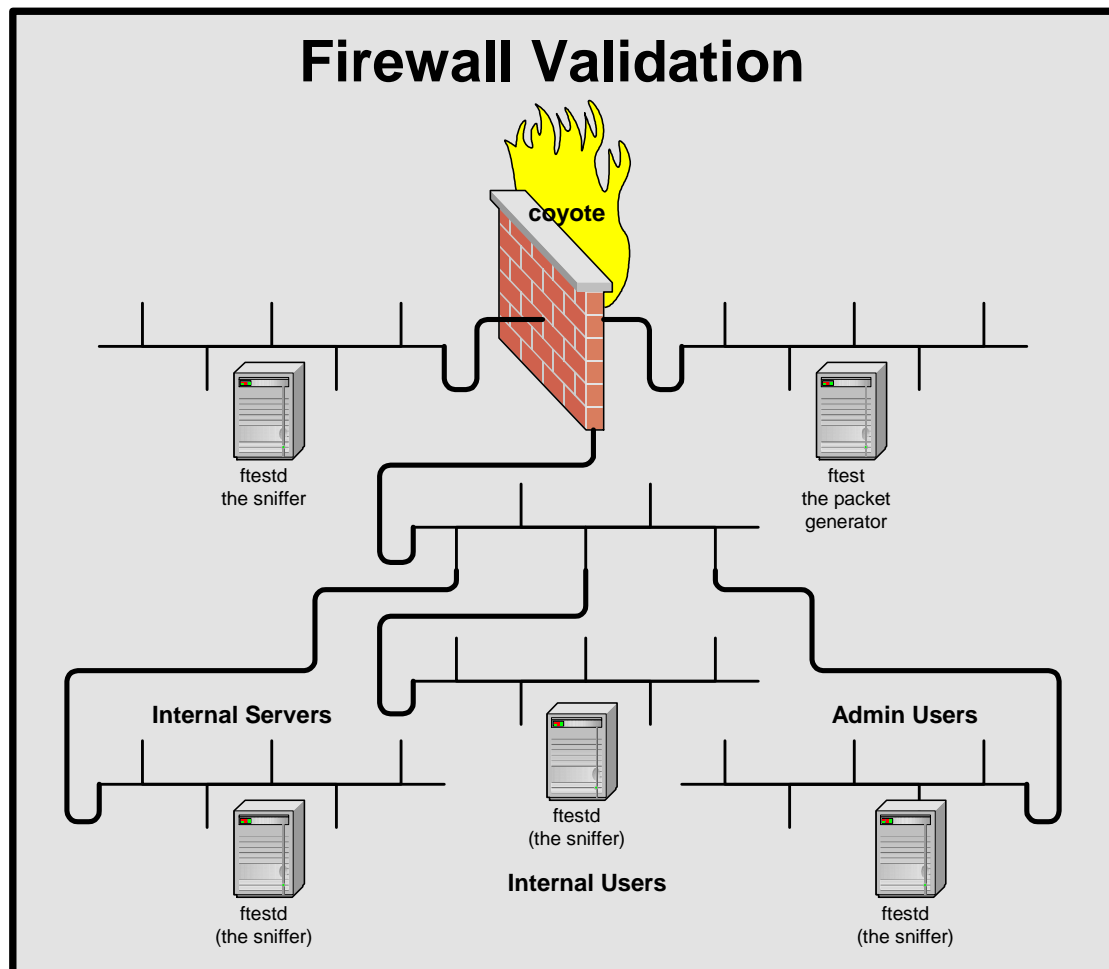
1. Perform a quick Nmap scan on each firewall interface for UDP, TCP, ICMP.
2. Detailed testing of traffic traversing the firewall on each firewall interface using ftester.

For the 2<sup>nd</sup> technique we need to prepare two hosts, a traffic generator and a sniffer. Fortunately there are already several tools available to do this. Firewalk and ftester<sup>10</sup> are a couple that are available. For our assessment we will use ftester. Fttester has the ability to generate real TCP/IP packets. It can be used to test the stateful capabilities of firewalls. The two scripts that encompass fttester were written by Andrea Barisani who was, “was tired of doing this by hand (with packet-crafting tools and tcpdump).”

Fttester allows one to specify source addresses in CIDR format. It has the ability to catch these spoofed addresses since it puts a marker in the packets it generates, to be able to identify them. The definitions of the packets to be sent



are specified in `ftest.conf`. The perl script `ftest` injects the packets and the listener `ftestd` captures them.



## 1.2 Timing Considerations

Typically, the best time to do testing is during non-business hours. However when your business is on the Internet there are no non-business hours. Therefore all business operations will be switched to the hot-site before we proceed. Once this has been accomplished we may disconnect the hot-site from the Internet and proceed with our testing plan. That takes care of our customers. To minimize the impact on our users at the main site we will do our testing during off hours.

## 1.3 Estimate Costs and Effort

Doing full Nmap scans of every port, is very time consuming. For example, if we perform an Nmap stealth scan in normal scanning mode, on the network in

question, we could scan 1600 TCP ports in about 1700 seconds. So that scan takes approximately 30 minutes for one scan to one IP. In order to perform this scan for the screened subnet, which contains 30 IP addresses would require 15 hours.

The UDP scans require even more time to complete. Another limitation of the scanning is that the scans are from one source address, so the results are for that host only. We could scan from one to many, however this is even more time consuming. As for scanning all 65535 ports available on a host, while this level of diligence is commendable, we will instead do quick nmap scans of the privileged ports, just to verify nothing is out of the ordinary.

Since we know the firewall policy we will concentrate on the stateful properties of the allowed traffic. Fortunately we have a tool that allows us to do this. So our level of effort will be to install the software on several hosts on different subnets, then run the various tests for each subnet. An estimate of 20 to 30 hours of work to setup and perform the testing seems reasonable.

#### **1.4 Identify Risks**

One of the biggest risks is the loss of revenue due to business loss caused by downtime of the site. Another risk is the firewall may crash during the process exposing us to external attacks or we may disrupt service by generating excess traffic. The switch over to the hot-site addresses these risks during testing operations at the main-site.

## **2 Validation**

### **2.1 How the Validation was Accomplished**

For each rule we need to determine if it works the way we expect. We begin with Nmap scans from an external host to the screened subnet and the firewall itself. Then we will proceed to Nmap scans from the Internal Admin, Internal User and Internal Server subnets to the firewall. We will scan the screened subnet and the firewall itself.

We will use ftester to test the statefulness of the rules and to verify that the rules actually work. It would be nice if some one would write a script to take the firewall rules and convert them to the configuration files needed to run ftester and then generate a report based on the results.

### **2.2 Tools and Commands**

The Nmap scan results will vary depending on the permissions of the scanning host. Some hosts are allowed more access than others, for example the firewall management station has more access to the firewall than other hosts. While performing these scans it is a good idea to keep an eye on the firewall logs. The logs will indicate success or failure of the connection attempts and provide correlation of the results of the Nmap scans.

The scans we will perform for each of the subnets will be of the following format:

```
Nmap -sS -P0 -v xxx.xxx.xxx.xxx  
Nmap -sU -p0 -v xxx.xxx.xxx.xxx  
Nmap -sP -v xxx.xxx.xxx.xxx
```

We will perform a TCP, UDP and ICMP scan in that order. The scans are in the Appendix.

Our next set of tests will determine if the rules are working the way they are supposed to. Whether or not they are allowing the traffic they are supposed to. We will use ftester to perform these tests.

### Internal Users Subnet

To test every possible host on the Internal Users Subnet, we can start ftstd on the external host. Then run ftest -f fttest.conf -v with an fttest.conf file like below. The -d option specifies a delay of 1 millisecond between packet injections. This configuration generates SYN packets on ports 1 through 1025 from every host on the Internal Admin.

```
flags: -e ttl1 -d 0.01  
192.168.23.0/24:1025:85.112.229.66:1-1025:S:TCP:0
```

This will test outbound connection attempts from the privileged ports on every host on that subnet. There are 254 hosts and 1025 ports, which will be 260350 SYN packets sent. With a delay of 1 millisecond each source host take slightly under 30 seconds to complete its run. So we can run the entire test in approximately 2 hours.

This scan completed with the listener catching no packets. We ran the same test on the Internal Admin and Server Subnets with the same results. Obviously these results also depend on the permissions of the host running the packet injector. The Internal Firewalls were also opened for this testing with an outbound any any rule.

### Allowed Outbound Traffic from Internal Users Subnet

The fttest.conf file shown below, tests the allowed connections from the Internal Users Subnet. The allowed outbound traffic is ftp, http and https from the proxy. We ran the test in connection spoofing mode, spoofing the proxy's source address. The other allowed outbound traffic is from SurfControl to update its database.

```
weston:[/usr/local/fttester-0.9]$more ftest  
flags: -e ttl1 -d 0.01  
#http to proxy
```

```
connect=192.168.23.7:1025:64.112.229.131:80:AP:TCP:0
#ftp to proxy
connect=192.168.23.7:1025:64.112.229.131:21:AP:TCP:0
#https to Proxy
connect=192.168.23.7:1025:64.112.229.131:443:AP:TCP:0
#SurfControl
connect=192.168.23.11:1025:216.251.249.236:80:AP:TCP:0
weston:[/usr/local/ftester-0.9]$
```

Now when we run ftest with with this configuration the output is shown below.

```
weston:[/usr/local/ftester-0.9]$./ftest -f ftest -v
Overriding command-line flags => flags: -e ttl1 -d 0.01
Sent Syn Probe => 192.168.23.7:1025 > 64.112.229.131:80 S TCP
Sleeping for 1 seconds
Sent Ack Reply => 192.168.23.7:1025 > 64.112.229.131:80 A TCP
3 - 192.168.23.7:1025 > 64.112.229.131:80 AP TCP
Sent Syn Probe => 192.168.23.7:1025 > 64.112.229.131:21 S TCP
Sleeping for 1 seconds
Sent Ack Reply => 192.168.23.7:1025 > 64.112.229.131:21 A TCP
7 - 192.168.23.7:1025 > 64.112.229.131:21 AP TCP
Sent Syn Probe => 192.168.23.7:1025 > 64.112.229.131:443 S TCP
Sleeping for 1 seconds
Sent Ack Reply => 192.168.23.7:1025 > 64.112.229.131:443 A TCP
11 - 192.168.23.7:1025 > 64.112.229.131:443 AP TCP
Sent Syn Probe => 192.168.23.11:1025 > 216.251.249.236:80 S TCP
Sleeping for 1 seconds
Sent Ack Reply => 192.168.23.11:1025 > 216.251.249.236:80 A TCP
15 - 192.168.23.11:1025 > 216.251.249.236:80 AP TCP
weston:[/usr/local/ftester-0.9]$
```

There was problem testing the allowed traffic using ftester with the proxy. I was unable to get the firewall to accept the packets I generated. Even though it should have passed them. The proxy server implementation works. I tried connect spoofing mode with a destination outside the firewall. Another interesting problem surfaced during the testing of proxy rules. No packets were logged on firewall for the attempted http connection.

The connections through the proxy failed, although the proxy itself works. The SurfControl connection was accepted, we got a reply because we forgot to set the ttl with the `-t` option. The reply was dropped because it was out of state. I believe the problems in testing are related to the SurfControl implementation, but did not have time to investigate further. The logged packets from the firewall are shown below. They are color coded to show how they relate to the configuration file.

Number: 46524  
Date: 2May2004  
Time: 6:28:46  
Product: VPN-1 & FireWall-1  
Interface: eth2  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Drop  
Service: ftp (21)  
Source: http-ftp-proxy (192.168.23.7)  
Destination: maverick31.sans.org (64.112.229.131)  
Protocol: tcp  
Source Port: Remote\_Storm (1025)  
Information: TCP packet out of state: First packet isn't SYN  
tcp\_flags: SYN-ACK

Number: 46525  
Date: 2May2004  
Time: 6:28:46  
Product: VPN-1 & FireWall-1  
Interface: eth2  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Drop  
Service: https (443)  
Source: http-ftp-proxy (192.168.23.7)  
Destination: maverick31.sans.org (64.112.229.131)  
Protocol: tcp  
Source Port: Remote\_Storm (1025)  
Information: TCP packet out of state: First packet isn't SYN  
tcp\_flags: SYN-ACK

Number: 46562  
Date: 2May2004  
Time: 6:41:57  
Product: VPN-1 & FireWall-1  
Interface: eth2  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: http (80)  
Source: SurfControl (192.168.23.11)  
Destination: www.surfcontrol.com (216.251.249.236)  
Protocol: tcp  
Rule: 28

NAT rule number: 7  
NAT additional rule number: 0  
Source Port: Remote\_Storm (1025)  
XlateSrc: coyote.giac.net (85.112.229.58)  
XlateSPort: 11463

Number: 46563  
Date: 2May2004  
Time: 6:42:00  
Product: VPN-1 & FireWall-1  
Interface: eth0  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Drop  
Service: 11463  
Source: www.surfcontrol.com (216.251.249.236)  
Destination: coyote.giac.net (85.112.229.58)  
Protocol: tcp  
Source Port: http (80)  
Information: TCP packet out of state: First packet isn't SYN  
tcp\_flags: SYN-ACK

For the next test we will use the configuration file below. This will test the stateful properties of the outbound rules for the Internal User Subnet. It does this by sending out of state packets, which should be dropped by the firewall.

```
weston:[/usr/local/ftester-0.9]$more ftest1
flags: -e ttl1 -d 0.01
#http to proxy
192.168.23.7:1025:192.168.20.1:80:SA:TCP:0
#ftp to proxy
192.168.23.7:1025:64.112.229.131:21:SA:TCP:0
#https to Proxy
192.168.23.7:1025:64.112.229.131:443:SA:TCP:0
#SurfControl
192.168.23.11:1025:216.251.249.236:80:SA:TCP:0
weston:[/usr/local/ftester-0.9]$
```

The output from the run is shown below.

```
weston:[/usr/local/ftester-0.9]$./ftest -f ftest1 -v
Overriding command-line flags => flags: -e ttl1 -d 0.01
1 - 192.168.23.7:1025 > 192.168.20.1:80 SA TCP 0
2 - 192.168.23.7:1025 > 64.112.229.131:21 SA TCP 0
3 - 192.168.23.7:1025 > 64.112.229.131:443 SA TCP 0
4 - 192.168.23.11:1025 > 216.251.249.236:80 SA TCP 0
```

```
weston:[usr/local/ftester-0.9]$
```

The expected outcome on the firewall is seen below in this snippet of the firewall log.

```
Drop http http-ftp-proxy coyote.giac.net Remote_Storm TCP packet out of state: First packet isn't SYN; tcp_flags: SYN-ACK
Drop ftp http-ftp-proxy maverick31.sans.org Remote_Storm TCP packet out of state: First packet isn't SYN; tcp_flags: SYN-ACK
Drop https http-ftp-proxy maverick31.sans.org Remote_Storm TCP packet out of state: First packet isn't SYN; tcp_flags: SYN-ACK
Drop http SurfControl www.surfcontrol.com Remote_Storm TCP packet out of state: First packet isn't SYN; tcp_flags: SYN-ACK
```

### Allowed Inbound Traffic to Internal Users Subnet

The only allowed inbound traffic for this subnet is from the Web Servers on the screened subnet to the MiddleWare Server. The fttest.conf file to test this and its run and the firewall log entry is shown below.

```
story:[usr/local/ftester-0.9]$more mid
flags: -e ttl1 -d 0.01
#from Web Server to MiddleWare
85.112.229.5:1025:192.168.23.9:1494:S:TCP:0
story:[usr/local/ftester-0.9]$./fttest -f mid -v
Overriding command-line flags => flags: -e ttl1 -d 0.01
1 - 85.112.229.5:1025 > 192.168.23.9:1494 S TCP 0
story:[usr/local/ftester-0.9]$
```

```
Number:      46647
Date:        2May2004
Time:        8:04:19
Product:     VPN-1 & FireWall-1
Interface:   eth1
Origin:      coyote.giac.net (85.112.229.58)
Type:        Log
Action:      Accept
Service:     winframe (1494)
Source:      GIAC_Web_Server (85.112.229.5)
Destination: Middle_Ware (192.168.23.9)
Protocol:    tcp
Rule:        7
Source Port: Remote_Storm (1025)
```

The corresponding stateful test is shown below. As expected, the packet is dropped as the first packet is not a SYN.

```
story:[usr/local/ftester-0.9]$more mid
flags: -e ttl1 -d 0.01
#from Web Server to MiddleWare
85.112.229.5:1025:192.168.23.9:1494:SA:TCP:0
story:[usr/local/ftester-0.9]$./fttest -f mid -v
```

```
Overriding command-line flags => flags: -e ttl1 -d 0.01  
1 - 85.112.229.5:1025 > 192.168.23.9:1494 SA TCP 0  
story:[/usr/local/ftester-0.9]$
```

```
Number:      46652  
Date:        2May2004  
Time:        8:08:31  
Product:     VPN-1 & FireWall-1  
Interface:   eth1  
Origin:      coyote.giac.net (85.112.229.58)  
Type:        Log  
Action:      Drop  
Service:     winframe (1494)  
Source:      GIAC_Web_Server (85.112.229.5)  
Destination: Middle_Ware (192.168.23.9)  
Protocol:    tcp  
Source Port: Remote_Storm (1025)  
Information: TCP packet out of state: First packet isn't SYN  
             tcp_flags: SYN-ACK
```

## Internal Server Subnet

### Allowed Outbound Traffic from Internal Server Subnet

The allowed traffic will include SMTP from Exchange to the Mail-Relay. The DNS traffic from the Internal-DNS to the External-DNS. The configuration file, run and firewall log entries are shown below.

```
story:[/usr/local/ftester-0.9]$more server  
flags: -e ttl1 -d 0.01  
#external dns to internal dns  
85.112.229.13:1025:192.168.21.12:53::UDP:0  
#Mail-Relay to Exchange  
85.112.229.9:1025:192.168.21.12:25:S:TCP:0  
story:[/usr/local/ftester-0.9]$
```

```
story:[/usr/local/ftester-0.9]$./ftest -f server -v  
Overriding command-line flags => flags: -e ttl1 -d 0.01  
1 - 85.112.229.13:1025 > 192.168.21.12:53 UDP 0  
2 - 85.112.229.25:1025 > 192.168.21.23:25 S TCP 0  
story:[/usr/local/ftester-0.9]$
```

```
Number:      46967  
Date:        2May2004  
Time:        11:44:42  
Product:     VPN-1 & FireWall-1
```



Interface: eth1  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: domain-udp (53)  
Source: External\_DNS (85.112.229.13)  
Destination: Internal\_DNS (192.168.21.12)  
Protocol: udp  
Rule: 4  
Source Port: 1025

Number: 46968  
Date: 2May2004  
Time: 11:44:42  
Product: VPN-1 & FireWall-1  
Interface: eth1  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: smtp (25)  
Source: Mail\_Relay (85.112.229.9)  
Destination: Exchange (192.168.21.23)  
Protocol: tcp  
Rule: 19  
Source Port: Remote\_Storm (1025)

### Allowed Inbound Traffic to Internal Server Subnet

The allowed traffic will include SMTP from the Mail-Relay to Exchange. The DNS traffic from the External-DNS to the Internal-DNS will be allowed. The configuration file, run and firewall log entries are shown below.

```
sundance:[/usr/local/ftester-0.9]#more out
flags: -e ttl1 -d 0.01
#internal dns to external dns
192.168.21.12:1025:85.112.229.13:53::UDP:0
#Exchange to Mail-Relay
192.168.21.23:1025:85.112.229.9:25:S:TCP:0
sundance:[/usr/local/ftester-0.9]#
```

```
sundance:[/usr/local/ftester-0.9]#./ftest -f out -v
Overriding command-line flags => flags: -e ttl1 -d 0.01
1 - 192.168.21.12:1025 > 85.112.229.13:53 UDP 0
2 - 192.168.21.23:1025 > 85.112.229.9:25 S TCP 0
sundance:[/usr/local/ftester-0.9]#
```

Number: 17  
Date: 2May2004  
Time: 12:14:29  
Product: VPN-1 & FireWall-1  
Interface: eth2  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: domain-udp (53)  
Source: Internal\_DNS (192.168.21.12)  
Destination: External\_DNS (85.112.229.13)  
Protocol: udp  
Rule: 3  
Source Port: 1025

Number: 16  
Date: 2May2004  
Time: 12:14:12  
Product: VPN-1 & FireWall-1  
Interface: eth2  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: smtp (25)  
Source: Exchange (192.168.21.23)  
Destination: Mail\_Relay (85.112.229.9)  
Protocol: tcp  
Rule: 19  
Source Port: Remote\_Storm (1025)

## Internal Admin Subnet

### Allowed Outbound Traffic from Internal Admin Subnet

The allowed traffic from the Internal Admin Subnet includes SSH to the firewall (coyote), SSH to the screened subnet, SSH to the border router (wineta), RADIUS to wineta and the WAP, and ntp to wineta. SNMP gets from the SNMP manager. The control traffic from the firewall MGMT station is also allowed from this subnet.

The configuration file, run and the firewall log entries for this test are shown below. The ntp packet appears out of order, since it is encrypted.

```
flags: -e ttl1 -d 0.01  
#ssh to firewall  
connect=192.168.22.22:1025:192.168.20.1:22:AP:TCP:0  
#ssh to wineta
```

```
192.168.22.22:1025:85.112.229.45:22:S:TCP:0
#ssh to wap
192.168.22.22:1025:85.112.229.226:22:S:TCP:0
#ssh to screened subnet
192.168.22.22:1025:85.112.229.17:22:S:TCP:0
#ntp to wineta-lo0
192.168.22.1:123:85.112.229.45:123::UDP:0
#fw mgmt station to firewall
connect=192.168.22.24:1025:192.168.20.1:18192:UR:TCP:0
connect=192.168.22.24:1025:192.168.20.1:18191:UR:TCP:0
#snmp manager to wineta-lo0
192.168.22.23:1025:85.112.229.45:161::UDP:0
```

```
weston:[/usr/local/ftester-0.9]$. /ftest -f admin -v
Overriding command-line flags => flags: -e ttl1 -d 0.01
Sent Syn Probe => 192.168.22.22:1025 > 192.168.20.1:22 S TCP
Sleeping for 1 seconds
Sent Ack Reply => 192.168.22.22:1025 > 192.168.20.1:22 A TCP
3 - 192.168.22.22:1025 > 192.168.20.1:22 AP TCP
5 - 192.168.22.22:1025 > 85.112.229.45:22 S TCP 0
6 - 192.168.22.22:1025 > 85.112.229.226:22 S TCP 0
7 - 192.168.22.22:1025 > 85.112.229.17:22 S TCP 0
6 - 192.168.22.1:123 > 85.112.229.45:123 UDP 0
Sent Syn Probe => 192.168.22.24:1025 > 192.168.20.1:18192 S TCP
Sleeping for 1 seconds
Sent Ack Reply => 192.168.22.24:1025 > 192.168.20.1:18192 A TCP
11 - 192.168.22.24:1025 > 192.168.20.1:18192 UR TCP 0
Sent Syn Probe => 192.168.22.24:1025 > 192.168.20.1:18191 S TCP
Sleeping for 1 seconds
Sent Ack Reply => 192.168.22.24:1025 > 192.168.20.1:18191 A TCP
15 - 192.168.22.24:1025 > 192.168.20.1:18191 UR TCP 0
16 - 192.168.22.23:1025 > 85.112.229.45:161 UDP 0
weston:[/usr/local/ftester-0.9]$.
```

```
Number: 46687
Date: 2May2004
Time: 8:36:46
Product: VPN-1 & FireWall-1
Interface: eth2
Origin: coyote.giac.net (85.112.229.58)
Type: Log
Action: Accept
Service: ssh (22)
Source: 192.168.22.22
Destination: coyote.giac.net (192.168.20.1)
```

Protocol: tcp  
Rule: 27  
NAT rule number: 7  
NAT additional rule number: 0  
Source Port: Remote\_Storm (1025)

Number: 46690  
Date: 2May2004  
Time: 8:36:48  
Product: VPN-1 & FireWall-1  
Interface: eth2  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Encrypt  
Service: ssh (22)  
Source: 192.168.22.22  
Destination: wineta\_lo0 (85.112.229.45)  
Protocol: tcp  
Rule: 23  
Source Port: Remote\_Storm (1025)  
Destination Key ID: 0xa4bb30ce  
Encryption Scheme: IKE  
VPN Peer Gateway: wineta-fa0-1 (85.112.229.57)  
Encryption Methods: ESP: 3DES + SHA1

Number: 46691  
Date: 2May2004  
Time: 8:36:48  
Product: VPN-1 & FireWall-1  
Interface: eth2  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: ssh (22)  
Source: 192.168.22.22  
Destination: WAP (85.112.229.226)  
Protocol: tcp  
Rule: 27  
Source Port: Remote\_Storm (1025)

Number: 46692  
Date: 2May2004  
Time: 8:36:48  
Product: VPN-1 & FireWall-1  
Interface: eth2  
Origin: coyote.giac.net (85.112.229.58)

Type: Log  
Action: Accept  
Service: ssh (22)  
Source: 192.168.22.22  
Destination: Reverse\_http\_Proxy (85.112.229.17)  
Protocol: tcp  
Rule: 27  
Source Port: Remote\_Storm (1025)

The two packets below consisting of the test of the firewall management station to the firewall are dropped. This has to do with the fact that the communication with the two hosts is encrypted and requires a preshared key for the communication to succeed. Referred to as by Checkpoint as Secure Internal Communication (SIC).

Number: 46693  
Date: 2May2004  
Time: 8:36:48  
Product: VPN-1 & FireWall-1  
Interface: eth2  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Drop  
Service: CPD\_amon (18192)  
Source: 192.168.22.24  
Destination: coyote.giac.net (192.168.20.1)  
Protocol: tcp  
Rule: 31  
Source Port: Remote\_Storm (1025)

Number: 46694  
Date: 2May2004  
Time: 8:36:49  
Product: VPN-1 & FireWall-1  
Interface: eth2  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Drop  
Service: CPD (18191)  
Source: 192.168.22.24  
Destination: coyote.giac.net (192.168.20.1)  
Protocol: tcp  
Rule: 31  
Source Port: Remote\_Storm (1025)

Number: 46695

Date: 2May2004  
Time: 8:38:17  
Product: VPN-1 & FireWall-1  
Interface: eth2  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Encrypt  
Service: ntp-udp (123)  
Source: crowsfoot-fa0-13 (192.168.22.1)  
Destination: wineta\_lo0 (85.112.229.45)  
Protocol: udp  
Rule: 24  
Source Port: ntp-udp (123)  
Destination Key ID: 0xa4bb30ce  
Encryption Scheme: IKE  
VPN Peer Gateway: wineta-fa0-1 (85.112.229.57)  
Encryption Methods: ESP: 3DES + SHA1

Number: 46769  
Date: 2May2004  
Time: 9:47:14  
Product: VPN-1 & FireWall-1  
Interface: eth2  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Encrypt  
Service: snmp-read (161)  
Source: 192.168.22.23  
Destination: wineta\_lo0 (85.112.229.45)  
Protocol: udp  
Rule: 23  
Source Port: 1025  
Destination Key ID: 0x56e4577d  
Encryption Scheme: IKE  
VPN Peer Gateway: wineta-fa0-1 (85.112.229.57)  
Encryption Methods: ESP: 3DES + SHA1

### Allowed Inbound Traffic to Admin Subnet

The allowed traffic to the Admin Subnet consists of SNMP RADIUS and Syslog from the border router (wineta) and the WAP. The configuration file, the run and the firewall log entries are shown below. Since the tunnel is up I am able to spoof traffic from another host across it. This is rather disturbing. Fortunately the border router ACL's will stop this. An attacker could possibly exploit this if the service that is tunneled has vulnerability. I can't use ftester to test the stateful properties since this is UDP.

```
[root@mexicanhat ftester-0.9]# more admin
flags: -e ttl1 -d 0.01
#RADIUS from WAP
85.112.229.226:1025:192.168.22.12:1812::UDP:0
#RADIUS from wineta-lo0
85.112.229.45:1025:192.168.22.12:1812::UDP:0
#syslog from wineta-lo0
85.112.229.45:1025:192.168.22.22:514::UDP:0
#snmp-trap from wineta-lo0
85.112.229.45:1025:192.168.22.23:162::UDP:0
[root@mexicanhat ftester-0.9]#
```

```
[root@mexicanhat ftester-0.9]# ./ftest -f admin -v
Overriding command-line flags => flags: -e ttl1 -d 0.01
1 - 85.112.229.226:1025 > 192.168.22.12:1812 UDP 0
2 - 85.112.229.45:1025 > 192.168.22.12:1812 UDP 0
3 - 85.112.229.45:1025 > 192.168.22.22:514 UDP 0
4 - 85.112.229.45:1025 > 192.168.22.23:162 UDP 0
[root@mexicanhat ftester-0.9]#
```

```
Number:      46775
Date:        2May2004
Time:        9:53:50
Product:     VPN-1 & FireWall-1
Interface:   eth0
Origin:      coyote.giac.net (85.112.229.58)
Type:        Log
Action:      Accept
Service:     NEW-RADIUS (1812)
Source:      WAP (85.112.229.226)
Destination: Internal_RADIUS (192.168.22.12)
Protocol:    udp
Rule:        26
Source Port: 1025
```

```
Number:      46776
Date:        2May2004
Time:        9:53:50
Product:     VPN-1 & FireWall-1
Interface:   eth0
Origin:      coyote.giac.net (85.112.229.58)
Type:        Log
Action:      Decrypt
Service:     NEW-RADIUS (1812)
Source:      wineta_lo0 (85.112.229.45)
Destination: Internal_RADIUS (192.168.22.12)
```

Protocol: udp  
Rule: 22  
Source Port: 1025  
Source Key ID: 0x88ee6538  
Encryption Scheme: IKE  
VPN Peer Gateway: wineta-fa0-1 (85.112.229.57)  
Encryption Methods: ESP: 3DES + SHA1

Number: 46777  
Date: 2May2004  
Time: 9:53:50  
Product: VPN-1 & FireWall-1  
Interface: eth0  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Decrypt  
Service: syslog (514)  
Source: wineta\_lo0 (85.112.229.45)  
Destination: 192.168.22.22  
Protocol: udp  
Rule: 21  
Source Port: 1025  
Source Key ID: 0x88ee6538  
Encryption Scheme: IKE  
VPN Peer Gateway: wineta-fa0-1 (85.112.229.57)  
Encryption Methods: ESP: 3DES + SHA1

Number: 46778  
Date: 2May2004  
Time: 9:53:50  
Product: VPN-1 & FireWall-1  
Interface: eth0  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Decrypt  
Service: snmp-trap (162)  
Source: wineta\_lo0 (85.112.229.45)  
Destination: 192.168.22.23  
Protocol: udp  
Rule: 21  
Source Port: 1025  
Source Key ID: 0x88ee6538  
Encryption Scheme: IKE  
VPN Peer Gateway: wineta-fa0-1 (85.112.229.57)  
Encryption Methods: ESP: 3DES + SHA1



## Screened Subnet

### Allowed Outbound Traffic from Screened Subnet

The allowed outbound traffic consists of the Terminal Services server to the Internal Proxy, Internal Servers, Exchange and Internal-DNS. The Mail-Relay is also allowed to connect to Exchange. The configuration file, run and the firewall log entries are below.

```
story:[/usr/local/ftester-0.9]$more ftest.conf
flags: -e ttl1 -d 0.01
#TS Server to proxy
85.112.229.25:1025:192.168.23.7:443:S:TCP:0
#TS Server to proxy
85.112.229.25:1025:192.168.23.7:80:S:TCP:0
#TS Server to proxy
85.112.229.25:1025:192.168.23.7:21:S:TCP:0
#TS Server to Internal Servers
85.112.229.25:1025:192.168.21.21:137::UDP:0
85.112.229.25:1025:192.168.21.21:138::UDP:0
85.112.229.25:1025:192.168.21.21:139:S:TCP:0
85.112.229.25:1025:192.168.21.21:445:S:TCP:0
#TS Server to Exchange
85.112.229.25:1025:192.168.21.23:135:S:TCP:0
#TS Server to internal DNS
85.112.229.25:1025:192.168.22.12:53::UDP:0
story:[/usr/local/ftester-0.9]$
```

```
story:[/usr/local/ftester-0.9]$./ftest -f ftest.conf -v
Overriding command-line flags => flags: -e ttl1 -d 0.01
1 - 85.112.229.25:1025 > 192.168.23.7:443 S TCP 0
2 - 85.112.229.25:1025 > 192.168.23.7:80 S TCP 0
3 - 85.112.229.25:1025 > 192.168.23.7:21 S TCP 0
4 - 85.112.229.25:1025 > 192.168.21.21:137 UDP 0
5 - 85.112.229.25:1025 > 192.168.21.21:138 UDP 0
6 - 85.112.229.25:1025 > 192.168.21.21:139 S TCP 0
7 - 85.112.229.25:1025 > 192.168.21.21:445 S TCP 0
8 - 85.112.229.25:1025 > 192.168.21.23:135 S TCP 0
9 - 85.112.229.25:1025 > 192.168.22.12:53 UDP 0
story:[/usr/local/ftester-0.9]$
```

Number: 41912  
Date: 1May2004  
Time: 10:55:39  
Product: VPN-1 & FireWall-1

Interface: eth1  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: https (443)  
Source: Terminal\_Services\_Server (85.112.229.25)  
Destination: http-ftp-proxy (192.168.23.7)  
Protocol: tcp  
Rule: 14  
Source Port: Remote\_Storm (1025)

Number: 41913  
Date: 1May2004  
Time: 10:55:39  
Product: VPN-1 & FireWall-1  
Interface: eth1  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: http (80)  
Source: Terminal\_Services\_Server (85.112.229.25)  
Destination: http-ftp-proxy (192.168.23.7)  
Protocol: tcp  
Rule: 14  
Source Port: Remote\_Storm (1025)

Number: 41915  
Date: 1May2004  
Time: 10:55:39  
Product: VPN-1 & FireWall-1  
Interface: eth1  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: nbname (137)  
Source: Terminal\_Services\_Server (85.112.229.25)  
Destination: 192.168.21.21  
Protocol: udp  
Rule: 15  
Source Port: 1025

Number: 41917  
Date: 1May2004  
Time: 10:55:39  
Product: VPN-1 & FireWall-1  
Interface: eth1

Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: nbdatagram (138)  
Source: Terminal\_Services\_Server (85.112.229.25)  
Destination: 192.168.21.21  
Protocol: udp  
Rule: 15  
Source Port: 1025

Number: 41918  
Date: 1May2004  
Time: 10:55:39  
Product: VPN-1 & FireWall-1  
Interface: eth1  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: nbssession (139)  
Source: Terminal\_Services\_Server (85.112.229.25)  
Destination: 192.168.21.21  
Protocol: tcp  
Rule: 15  
Source Port: Remote\_Storm (1025)

Number: 41919  
Date: 1May2004  
Time: 10:55:39  
Product: VPN-1 & FireWall-1  
Interface: eth1  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: microsoft-ds (445)  
Source: Terminal\_Services\_Server (85.112.229.25)  
Destination: 192.168.21.21  
Protocol: tcp  
Rule: 15  
Source Port: Remote\_Storm (1025)

Number: 41920  
Date: 1May2004  
Time: 10:55:39  
Product: VPN-1 & FireWall-1  
Interface: eth1  
Origin: coyote.giac.net (85.112.229.58)

Type: Log  
Action: Accept  
Service: dce-rpc (135)  
Source: Terminal\_Services\_Server (85.112.229.25)  
Destination: Exchange (192.168.21.23)  
Protocol: tcp  
Rule: 16  
Source Port: Remote\_Storm (1025)

Number: 41921  
Date: 1May2004  
Time: 10:55:39  
Product: VPN-1 & FireWall-1  
Interface: eth1  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: domain-udp (53)  
Source: Terminal\_Services\_Server (85.112.229.25)  
Destination: Internal\_RADIUS (192.168.22.12)  
Protocol: udp  
Rule: 17  
Source Port: 1025

Now we can test the stateful properties of these rules. If we modify them slightly to be a SYN-ACK packet they should be dropped by the firewall. The configuration file, run and expected result are shown below.

```
story:[/usr/local/ftester-0.9]$more ftest.conf
flags: -e ttl1 -d 0.01
#TS Server to proxy
85.112.229.25:1025:192.168.23.7:443:SA:TCP:0
#TS Server to proxy
85.112.229.25:1025:192.168.23.7:80:SA:TCP:0
#TS Server to proxy
85.112.229.25:1025:192.168.23.7:21:SA:TCP:0
#TS Server to Internal Servers
85.112.229.25:1025:192.168.21.21:139:SA:TCP:0
85.112.229.25:1025:192.168.21.21:445:SA:TCP:0
#TS Server to Exchange
85.112.229.25:1025:192.168.21.23:135:SA:TCP:0
story:[/usr/local/ftester-0.9]$
```

```
story:[/usr/local/ftester-0.9]$./ftest -f ftest.conf -v
Overriding command-line flags => flags: -e ttl1 -d 0.01
1 - 85.112.229.25:1025 > 192.168.23.7:443 SA TCP 0
```

```
2 - 85.112.229.25:1025 > 192.168.23.7:80 SA TCP 0
3 - 85.112.229.25:1025 > 192.168.23.7:21 SA TCP 0
4 - 85.112.229.25:1025 > 192.168.21.21:139 SA TCP 0
5 - 85.112.229.25:1025 > 192.168.21.21:445 SA TCP 0
6 - 85.112.229.25:1025 > 192.168.21.23:135 SA TCP 0
story:[/usr/local/ftester-0.9]$
```

Firewall snippet showing expected result of out of state packets.

Terminal_Services_Server	http-ftp-proxy	Remote_Storm	TCP packet out of state: First packet isn't SYN; tcp_flags: SYN-ACK
Terminal_Services_Server	http-ftp-proxy	Remote_Storm	TCP packet out of state: First packet isn't SYN; tcp_flags: SYN-ACK
Terminal_Services_Server	http-ftp-proxy	Remote_Storm	TCP packet out of state: First packet isn't SYN; tcp_flags: SYN-ACK
Terminal_Services_Server	192.168.21.21	Remote_Storm	TCP packet out of state: First packet isn't SYN; tcp_flags: SYN-ACK
Terminal_Services_Server	Exchange	Remote_Storm	TCP packet out of state: First packet isn't SYN; tcp_flags: SYN-ACK

### Allowed Inbound Traffic from External to Screened Subnet

The allowed traffic from external hosts to the screened subnet consists of DNS to the External-DNS, SMTP from Postini Mail Services and Customer access to the Reverse Proxy. There is also VPN traffic from SecureClient to the Terminal Services Server and Site-to-Site VPN's from Partners and Suppliers to the ftp Server. We will deal with the VPN traffic in the next section. The configuration file for the allowed inbound traffic, the run and the firewall log entries are shown below.

```
[root@mexicanhat ftester-0.9]# more ftest.conf
#DNS
85.112.229.66:1025:85.112.229.13:53:S:TCP:0
#Mail Relay Postini Spoofed Source
connect=63.240.181.100:1025:85.112.229.9:25:UR:TCP:0
#Reverse Proxy
85.112.229.66:1025:85.112.229.17:443:S:TCP:0
85.112.229.66:1025:85.112.229.17:80:S:TCP:0
[root@mexicanhat ftester-0.9]#
```

```
[root@mexicanhat ftester-0.9]# ./ftest -f ftest.conf -v
1 - 85.112.229.66:1025 > 85.112.229.13:53 S TCP 0
Sent Syn Probe => 63.240.181.100:1025 > 85.112.229.9:25 S TCP
Sleeping for 1 seconds
Sent Ack Reply => 63.240.181.100:1025 > 85.112.229.9:25 A TCP
4 - 63.240.181.100:1025 > 85.112.229.9:25 UR TCP 0
6 - 85.112.229.66:1025 > 85.112.229.17:443 S TCP 0
7 - 85.112.229.66:1025 > 85.112.229.17:80 S TCP 0
[root@mexicanhat ftester-0.9]#
```

```
Number:      40509
Date:        1May2004
Time:        3:01:26
```

Product: VPN-1 & FireWall-1  
Interface: eth0  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: domain-tcp (53)  
Source: 85.112.229.66  
Destination: External\_DNS (85.112.229.13)  
Protocol: tcp  
Rule: 5  
Source Port: Remote\_Storm (1025)

Number: 40515  
Date: 1May2004  
Time: 3:03:02  
Product: VPN-1 & FireWall-1  
Interface: eth0  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: smtp (25)  
Source: list.postini.com.mail7.psmtp.com (63.240.181.100)  
Destination: Mail\_Relay (85.112.229.9)  
Protocol: tcp  
Rule: 20  
Source Port: Remote\_Storm (1025)

Number: 40510  
Date: 1May2004  
Time: 3:01:27  
Product: VPN-1 & FireWall-1  
Interface: eth0  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: https (443)  
Source: 85.112.229.66  
Destination: Reverse\_http\_Proxy (85.112.229.17)  
Protocol: tcp  
Rule: 6  
Source Port: Remote\_Storm (1025)

Number: 40511  
Date: 1May2004  
Time: 3:01:27

Product: VPN-1 & FireWall-1  
Interface: eth0  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Accept  
Service: http (80)  
Source: 85.112.229.66  
Destination: Reverse\_http\_Proxy (85.112.229.17)  
Protocol: tcp  
Rule: 6  
Source Port: Remote\_Storm (1025)

We can test the stateful properties by slightly modifying the configuration file. The configuration, run and firewall log entries are shown below.

```
[root@mexicanhat ftester-0.9]# more fttest.conf
flags: -e ttl1 -d 0.01
#DNS
85.112.229.66:1025:85.112.229.13:53:AP:TCP:0
85.112.229.66:53:85.112.229.13:53::UDP
#Mail Relay
85.112.229.66:1025:85.112.229.9:25:AP:TCP:0
#Reverse Proxy
85.112.229.66:1025:85.112.229.17:443:AP:TCP:0
85.112.229.66:1025:85.112.229.17:80:AP:TCP:0
85.112.229.66::85.112.229.17:::ICMP:3:5
[root@mexicanhat ftester-0.9]#
```

```
[root@mexicanhat ftester-0.9]# ./fttest -f fttest.conf -v
Overriding command-line flags => flags: -e ttl1 -d 0.01
1 - 85.112.229.66:1025 > 85.112.229.13:53 AP TCP 0
2 - 85.112.229.66:53 > 85.112.229.13:53 UDP
3 - 85.112.229.66:1025 > 85.112.229.9:25 AP TCP 0
4 - 85.112.229.66:1025 > 85.112.229.17:443 AP TCP 0
5 - 85.112.229.66:1025 > 85.112.229.17:80 AP TCP 0
6 - 85.112.229.66 > 85.112.229.17 ICMP 3 5
[root@mexicanhat ftester-0.9]#
```

Number: 466661  
Date: 29Apr2004  
Time: 12:25:03  
Product: VPN-1 & FireWall-1  
Interface: eth0  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Drop

Service: domain-tcp (53)  
Source: 85.112.229.66  
Destination: External\_DNS (85.112.229.13)  
Protocol: tcp  
Source Port: Remote\_Storm (1025)  
Information: TCP packet out of state: First packet isn't SYN  
tcp\_flags: PUSH-ACK

Number: 466662  
Date: 29Apr2004  
Time: 12:25:03  
Product: VPN-1 & FireWall-1  
Interface: eth0  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Drop  
Service: domain-udp (53)  
Source: 85.112.229.66  
Destination: External\_DNS (85.112.229.13)  
Protocol: udp  
Rule: 30  
Source Port: domain-udp (53)

Number: 466663  
Date: 29Apr2004  
Time: 12:25:03  
Product: VPN-1 & FireWall-1  
Interface: eth0  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Drop  
Service: smtp (25)  
Source: 85.112.229.66  
Destination: Mail\_Relay (85.112.229.9)  
Protocol: tcp  
Source Port: Remote\_Storm (1025)  
Information: TCP packet out of state: First packet isn't SYN  
tcp\_flags: PUSH-ACK

Number: 466664  
Date: 29Apr2004  
Time: 12:25:03  
Product: VPN-1 & FireWall-1  
Interface: eth0  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log



Action: Drop  
Service: https (443)  
Source: 85.112.229.66  
Destination: Reverse\_http\_Proxy (85.112.229.17)  
Protocol: tcp  
Source Port: Remote\_Storm (1025)  
Information: TCP packet out of state: First packet isn't SYN  
tcp\_flags: PUSH-ACK

Number: 466665  
Date: 29Apr2004  
Time: 12:25:03  
Product: VPN-1 & FireWall-1  
Interface: eth0  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Drop  
Service: http (80)  
Source: 85.112.229.66  
Destination: Reverse\_http\_Proxy (85.112.229.17)  
Protocol: tcp  
Source Port: Remote\_Storm (1025)  
Information: TCP packet out of state: First packet isn't SYN  
tcp\_flags: PUSH-ACK

Number: 466666  
Date: 29Apr2004  
Time: 12:25:03  
Product: VPN-1 & FireWall-1  
Interface: eth0  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Drop  
Source: 85.112.229.66  
Destination: Reverse\_http\_Proxy (85.112.229.17)  
Protocol: icmp  
Information: ICMP: Source Route Failed  
ICMP Type: 3  
ICMP Code: 5  
message\_info: ICMP errors are not allowed

We will show a couple more stateful test and their results since we are all setup.

```
[root@mexicanhat ftester-0.9]# more fttest.conf
flags: -e ttl1 -d 0.01
#DNS
```

```
85.112.229.66:1025:85.112.229.13:53:AS:TCP:0
#Mail Relay
85.112.229.66:1025:85.112.229.9:25:AS:TCP:0
#Reverse Proxy
85.112.229.66:1025:85.112.229.17:443:AS:TCP:0
85.112.229.66:1025:85.112.229.17:80:AS:TCP:0
[root@mexicanhat ftester-0.9]#
```

```
⊙ Drop domain-tcp 85.112.229.66 External_DNS TCP packet out of state: First packet isn't SYN; tcp_flags: SYN-ACK
⊙ Drop smtp 85.112.229.66 Mail_Relay TCP packet out of state: First packet isn't SYN; tcp_flags: SYN-ACK
⊙ Drop https 85.112.229.66 Reverse_http_Proxy TCP packet out of state: First packet isn't SYN; tcp_flags: SYN-ACK
⊙ Drop http 85.112.229.66 Reverse_http_Proxy TCP packet out of state: First packet isn't SYN; tcp_flags: SYN-ACK
```

```
[root@mexicanhat ftester-0.9]# more fttest.conf
flags: -e ttl1 -d 0.01
#DNS
85.112.229.66:1025:85.112.229.13:53:UR:TCP:0
#Mail Relay
85.112.229.66:1025:85.112.229.9:25:UR:TCP:0
#Reverse Proxy
85.112.229.66:1025:85.112.229.17:443:UR:TCP:0
85.112.229.66:1025:85.112.229.17:80:UR:TCP:0
[root@mexicanhat ftester-0.9]#
```

```
⊙ Drop domain-tcp 85.112.229.66 External_DNS TCP packet out of state: First packet isn't SYN; tcp_flags: RST-URG
⊙ Drop smtp 85.112.229.66 Mail_Relay TCP packet out of state: First packet isn't SYN; tcp_flags: RST-URG
⊙ Drop https 85.112.229.66 Reverse_http_Proxy TCP packet out of state: First packet isn't SYN; tcp_flags: RST-URG
⊙ Drop http 85.112.229.66 Reverse_http_Proxy TCP packet out of state: First packet isn't SYN; tcp_flags: RST-URG
```

## VPN's

Some of the allowed VPN traffic includes Syslog, SNMP and RADIUS to or from the border router, which we have covered already. Other VPN traffic includes SecureClient to the Terminal Services Server. The database synchronization from main-ste to hot-site is over a VPN. Finally partners and suppliers connect to the ftp Server via a site-to-site VPN.

To test VPN traffic using fttester we need to have a tunnel established. It has no mechanism for establishing a VPN connection. Our Secureclient implementation can be established from anywhere. So the only thing to be tested for inbound connections is whether or not the user is authenticated and that they are restricted to the Terminal Services port on the Terminal Services Server.

Below are the firewall log entries for establishing a SecureClient connection to the Firewall and logging in to the Policy Server. The tunnel is initiated (133 and 134) with a shared secret (132) that the client receives from the Firewall module through a topology download. The client then receives authorization in the form of a login from the policy server (135 and 136).

Number: 132  
Date: 2May2004  
Time: 13:44:37  
Product: VPN-1 & FireWall-1  
Interface: daemon  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Login  
Source: HDQNETJRAUSERAL (64.112.229.131)  
Destination: coyote.giac.net (85.112.229.58)  
User: jimrauser  
Encryption Scheme: IKE  
Encryption Methods: 3DES,IKE,SHA1  
Information: reason: Client Encryption: Authenticated by Internal  
Password

Number: 133  
Date: 2May2004  
Time: 13:44:38  
Product: VPN-1 & FireWall-1  
Interface: daemon  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Key Install  
Source: HDQNETJRAUSERAL (64.112.229.131)  
Destination: coyote.giac.net (85.112.229.58)  
User: jimrauser  
Encryption Scheme: IKE  
VPN Peer Gateway: HDQNETJRAUSERAL (64.112.229.131)  
IKE Initiator Cookie: d8f7f98ebfec786d  
IKE Responder Cookie: 803b4793f3b0b4e4  
IKE Phase2 Message ID: fd17bb05  
Encryption Methods: 3DES + SHA1, Internal Password  
Information: IKE: Main Mode completion.

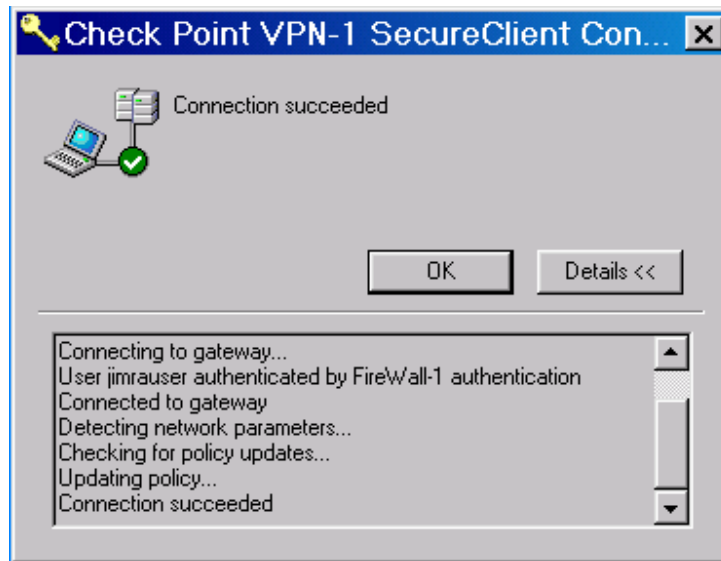
Number: 134  
Date: 2May2004  
Time: 13:44:39  
Product: VPN-1 & FireWall-1  
Interface: daemon  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Key Install  
Source: HDQNETJRAUSERAL (64.112.229.131)  
Destination: coyote.giac.net (85.112.229.58)

User: jimrauser  
Source Key ID: 0xffaa78d7  
Destination Key ID: 0x4df62cb0  
Encryption Scheme: IKE  
VPN Peer Gateway: HDQNETJRAUSERAL (64.112.229.131)  
IKE Initiator Cookie: d8f7f98ebfec786d  
IKE Responder Cookie: 803b4793f3b0b4e4  
IKE Phase2 Message ID: 8b3b4c99  
Encryption Methods: ESP: 3DES + SHA1  
Information: IKE: Quick Mode completion  
IKE IDs: subnet: 0.0.0.0 (mask= 0.0.0.0) and  
host: 64.112.229.131

Number: 135  
Date: 2May2004  
Time: 13:44:39  
Product: VPN-1 & FireWall-1  
Interface: daemon  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Login  
Source: HDQNETJRAUSERAL (64.112.229.131)  
Destination: coyote.giac.net (85.112.229.58)  
User: jimrauser  
Encryption Scheme: IKE  
Start Time: 2May2004 13:44:39  
Information: VPN internal source IP: HDQNETJRAUSERAL  
reason: connected to gateway

Number: 136  
Date: 2May2004  
Time: 13:44:45  
Product: Policy Server  
Origin: coyote.giac.net (85.112.229.58)  
Type: Log  
Action: Login  
Source: HDQNETJRAUSERAL (64.112.229.131)  
Destination: coyote.giac.net (85.112.229.58)  
User: jimrauser  
Information: PS: User jimrauser logged on to Policy Server.

The corresponding dialog box for the login and policy download is shown below.



I did not have enough gear to be to test the site-to-site VPN's. If one did you could attempt to establish a VPN with the firewall module. It should not work since you would need to know the pre-shared keys and the encryption parameters. Even if you knew this, the encryption rules are written to only all the establishment of a VPN with certain sites.

## 3 Results

### 3.1 Analysis

The Nmap scans and the tests with ftester, clearly show that the firewall is allowing the traffic it is supposed to. The Nmap scans show that the firewall is denying the traffic that is not permitted. The stateful capabilities of the firewall are also working properly as evidenced from ftester tests. The SecureClient application only works for authorized users although Username/Password is weak. The only thing that really seemed odd, was my ability to spoof UDP traffic across a tunnel (from the border router) that was already up.

This certainly could be exploitable. The border router ACL's were protecting against this for our configuration. However, I was unable to test this from another site-to-site VPN due to lack of gear. I don't think this is limited to UDP. It should also work with TCP but I did not test that.

### 3.2 Recommendations for Improvement

Using some form of strong authentication could further protect the SecureClient implementation. Either a token system or client side certificates could be added to the current configuration for further security. The same could be said of the WAP. We could use client-side certificates here and implement EAP-TLS instead of EAP-TTLS. Also our site-to-site VPN's could use certificates instead of pre-shared keys.

© SANS Institute 2004, Author retains full rights

## Assignment 4 Design Under Fire

### 1 Select a Network

For this portion of the practical I chose Miles Parkins<sup>11</sup> firewall design. [http://www.giac.org/practical/GCFW/Miles\\_Parkin\\_GCFW.pdf](http://www.giac.org/practical/GCFW/Miles_Parkin_GCFW.pdf). He had several diagrams that showed traffic flow, Diagram 2-5 is the design as a whole. This is shown below.

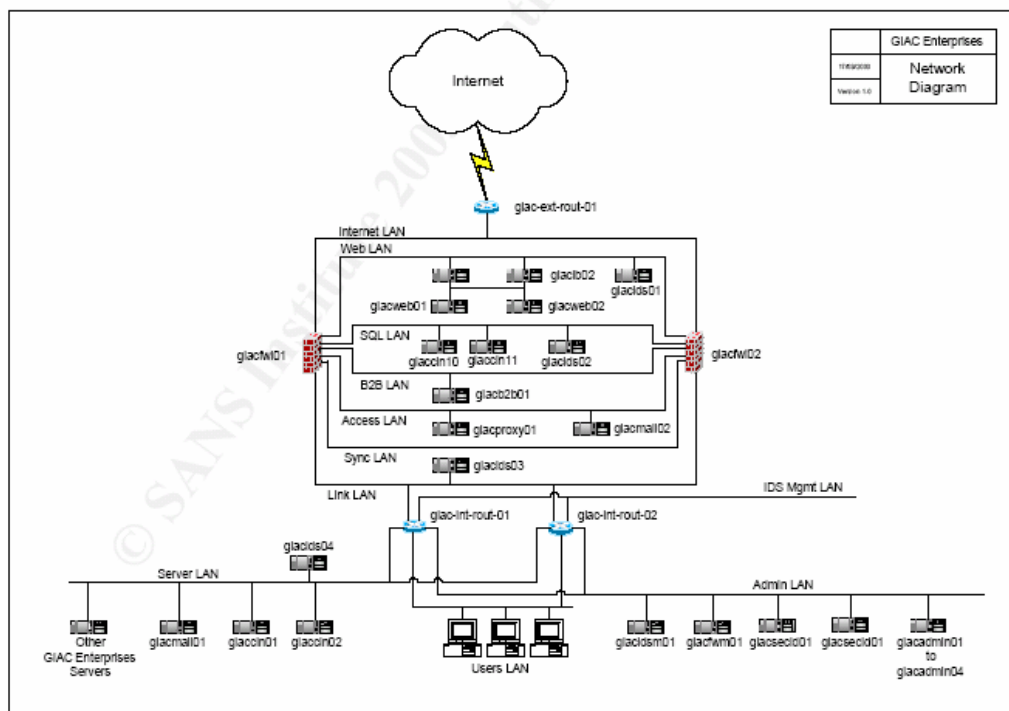


Diagram 2-5 Complete Network Diagram

His design incorporates the border router, screened subnet, a clustered primary firewall and internal routers. The internal network is segregated into users, servers and admin LAN's. A Trend Micro proxy is used for outbound web/ftp access. There is a mail relay with a connection to the internal mail server. Also there is a web server on the screened subnet with, a connection to the internal database. There is no split-DNS; there is no internal DNS. The web proxy and mail relay are allowed outbound DNS.

## 2 Firewall Attack

### 2.1 Vulnerability

The firewall was described as CheckPoint SecurePlatform NG with AI, running on an Intel platform. No mention of the current Hot Fix Accumulator is made. We will assume it was patched to the current version at that time. This would likely be Hotfix Accumulator 317. He is running a proxy for outbound web traffic so how does that get out without using the http security server? We will come back to this, for now lets look at what we can do with the given policy.

The firewall accepts control connections from the management station. Possibly we could use this to spoof the firewall module in to thinking we are the management station. I don't know how an attacker would get the management stations IP address, but it is possible. However to get the replies back to the attacker, would involve source routing. Source routing is disabled on the border router and access lists are also in place to prevent spoofing internal addresses. So that avenue of attack looks to be closed. Other possibilities to try would be connections at boot time but they are also not allowed by the policy, so they also appear to be out of the question.

Returning to the question of the http security server. There are known vulnerabilities for the http security server. The firewall policy shows he is not running it. However, I believe Trend's InterScan VirusWall which he is using, requires you to define a "next Proxy". For this to work the firewall becomes the "next Proxy" thus the need to run an http security server. Also the security servers run on port 80 by default. The \$FWDIR/conf/fwauth.conf file would needed to have been modified to get an http security server running on port 1080, so I don't think this is an actual working policy. I need to verify this.

If we proceed from this assumption, that the firewall rules are not correct and the web proxy would need a security server for a working policy. Since this paper was written, there are known vulnerabilities for the http security server.

On February 4<sup>th</sup> 2004, Internet Security Systems<sup>12</sup> released the following advisory: "ISS X-Force has discovered a flaw in the HTTP Application Intelligence component of Firewall-1. Application Intelligence is a relatively recent addition to the Firewall-1 product line and functions as an application proxy between untrusted networks and network servers for the purpose of detecting and preventing potential attacks. The vulnerabilities also exist within the HTTP Security Server application proxy that ships with all versions of Firewall-1 (including those prior to Application Intelligence releases). The affected components contain several remotely exploitable format string vulnerabilities."

So it would appear even if the security server were not running this is exploitable. That is, if the AI http component is enabled. This cannot be determined from the practical, since it is not mentioned. However, I believe the default is for it to be enabled.



## 2.2 Attack

A simple port scan would tell us this is a CheckPoint firewall since it is listening on port 264 for topology downloads. This is allowed on the access list on the border router and has to be listening for SecureClient to work. So taking the path of least resistance we would launch our format string attack against the web server behind the firewall with the above mentioned exploit. The CVE CAN-2004-0039<sup>13</sup> describes this as, “multiple format sting vulnerabilities in the HTTP Application Intelligence (AI) component in Check Point Firewall-1 NG-AI R55 and R54, and Check Point Firewall-1 HTTP Security Server included with NG FP1, FP2, and FP3 allows remote attackers to execute arbitrary code via HTTP requests that cause format string specifiers to be used in an error message, as demonstrated using the scheme of a URI.”

So now we need to find out the details of the exploit. A search of the web turned up little of value. CheckPoint claims the exploit is in theory only, while ISS claim to have proof of concept code, however they state on some platforms it is not trivial. I can't seem to find any exploit code to test it out on my test network. It most likely works and the exploit code will probably leak out in the future.

The attack is based on the fact that an attacker can partially specify the format string to the sprintf() function call. Using this the attacker can overwrite memory and execute code of their choice.

## 2.3 Results

A remote attacker may be able to use these attacks to execute commands on the firewall. The most likely result for an unpatched system would be a full compromise. Unsuccessful exploit attempts will disrupt all established HTTP sessions and stop Web traffic momentarily.

## 2.4 Countermeasures

Upload and install the Firewall-1 HTTP Security Hotfix from the Check Point Web site. Turn off security servers and/or the http AI component.

# 3 Distributed Denial of Service Attack

## 3.1 Compromise the Zombies

We want to compromise Windows XP machines since they have support for raw sockets, this will give us the ability to spoof. So as part of our reconnaissance we should fingerprint the OS. We will be looking for unpatched Windows XP DSL/cable modem machines that have been compromised by the worm of the day. Typically these machines will be listening on a backdoor port. So we will scan for these listeners, connect to the backdoor, patch the system and install our bot. Alternatively we could compromise the machines ourselves

but hey, why bother when the worm has already done all the work. The compromise itself could be any of a number of the current exploits available.

### 3.2 DDoS Design

Steve Gibson<sup>14</sup> of Gibson Research states that, "IRC Bots are among the newer breed of Distributed Denial of Service (DDoS) agents deployed by the Internet's most active hackers. Whenever an IRC bot hosting Windows PC is started, the bot waits for the system to finish booting, then connects to a previously designated IRC server. Using a private password key, it joins a secret IRC channel that is not visible to other users of the IRC server ...and awaits commands." With that in mind we will design our attack around IRC bots or zombies.

Upon startup the bots will join the specified IRC channel and await further commands. Evilbot (the bot used against Gibson) will accept commands from anyone on the channel. Slackbot requires a password. So we do need to specify an IRC server for the bots to connect to and for us to control them.

As an example, these are typical commands that can be issued, "The following command on the same IRC channel where Evilbot is would cause a UDP packet flood attack against a certain host "!udp 101.105.201.212 1000 0", 1000 on the line stands for amount of packets that will be sent and 0 on the end of line stands for the delay between each packet. Evilbot can attack by pinging a target host too; it supports four different kinds of ping attacks:

!p4	Sends 10000 64 kbyte ping packets to specified IP
!p3	Sends 1000 64 kbyte ping packets to specified IP
!p2	Sends 100 64 kbyte ping packets to specified IP
!p1	Sends 10 64 kbyte ping packets to specified IP

The amount of pings and the ping size can be configured (for ex. !p4 command could have been set to send 15000 32 byte ICMP packets to a specified host, but defaultly it uses the above values)" excerpted from an article on [www.netsys.com](http://www.netsys.com).<sup>15</sup>

The above demonstrates the ease that an attack may be coordinated once the bots are in place. Hiding the communications channel is another problem since all bots talk to the same place. This is one of its weaknesses since the source of control is easily tracked down. Having the bots connect to multiple IRC servers and/or servers in other countries would slow this process down.

### 3.3 Attack

So as an attacker we could map the live hosts in preparation for our DDoS attack. An attacker can also use ICMP to map past the router since the ACL allows ICMP out. The attacker can also map the access list since the router will send ICMP unreachable for denied items in the access list. Our DDoS options

could also include SYN flooding (depends on the bot) hosts to use up the bandwidth and/or connections to the web server and mail relay. The firewall will allow these and they will reply back. We should use spoofed hosts that are not up but are real live IP's this will use resources on these hosts. Other attacks could include taking out the DNS by sending unsolicited replies from spoofed hosts against the DNS. Additionally we could find a smurf network to use as an amplifier for the DDoS clients.

The external router looks like a good candidate for some fun, since it is the only entry point. The connection speed is relatively slow and is a frame-relay circuit. All that is mentioned is that it is a Cisco 2612 running Cisco IOS 12.2. Also the firewalls are clustered, so attacks against them will probably not be as easy.

An Nmap scan to fingerprint the router will fail since the ACL won't allow the necessary UDP and TCP connections. If we were able to guess the IOS version we would not be able to use many of the exploits since they are also blocked by the ACL. Again, taking the path of least resistance, as an attacker we can't determine the IOS or that it is a Cisco router, but we could surmise this and look for ICMP attacks for IOS 12 and higher. We can determine its address 195.110.67.18 via traceroute since ICMP is not blocked on the device (on a host that uses ICMP for traceroute). So the router itself is also open for DDoS attacks using ICMP.

### 3.4 Results

The results of the attack will be consumption of bandwidth and resources on the attacked hosts. A successful DDoS will prevent access to the Internet through the consumption of these resources. In the classic SYN flood the attacked hosts will send SYN/ACKS and never receive replies. This does not exhaust the resources of the host since newer versions of OS's use SYN cookies to mitigate this, however the bandwidth will be exhausted.

### 3.5 Countermeasures

Insure that there is adequate bandwidth and/or redundant paths. Filter ICMP at border router to stop Smurf amplifier attacks, Deploy an IDS to alert of DDoS. Patch systems regularly. Use egress filters to stop traffic that is not from our network, since zombies generally spoof. Make sure your upstream provider is capable of blocking these attacks and find out whom to contact should one occur.

For a web site, collocation may be the best option. If it gets DDoS'ed your network is still up.

## 4 Internal System Compromise

### 4.1 Target

Compromising an internal host on a network with some security is a difficult task. It will require taking advantage of multiple vulnerabilities and or exploits. Our target will be one of the hosts on the users LAN. We will pattern our attack using the five phases of an attack as described by Ed Skoudis<sup>16</sup>.

### 4.2 Attack

#### Reconnaissance

From the network diagram and descriptions there appears to be no internal DNS. It must be hosted externally. Using a whois lookup we can find the netblock that is registered to GIAC enterprises. This will give us the public IP address space. It should also give us the nameservers that service this address space. We can map this space to hosts by doing DNS forward lookups one at a time, or all at once if they allow zone transfers. This should give us the addresses of the web servers, mail relay and/or any other host in the DNS. We might also get a persons name, address, phone number etc. who is responsible for this domain.

Before doing any scanning we should take our time and do a detailed reconnaissance for any publicly available information. We can begin with web searches on their domain name and their own web site. Any information revealed will be used to generate further searches. We would like to find their addresses, phone contacts, business partners, employee names, e-mail addresses, organizational structure, business functions, SysAdmin names, etc. In general any piece of information we can turn up can aid us in conducting the next phases of the attack.

At this point we should have turned up some contact names and phone numbers that we can use to leverage more information from a called employee.

This can be very involved and proceed over weeks or months as we try to build trust and credibility with the target employee. There are a number of very common pretexts to try which usually revolve around getting a username and/or password. This type of attack or “social engineering” can involve third parties, females, anyone who is skillful at manipulating other people. Gaining a username/password combination in this manner is the easiest way in and should be attempted.

Physical access to their premises is risky but may be attempted if we can ascertain from some of our previous information the level of security on site. If we can gain access, planting a Wireless Access Point on their premises would be highly desirable. If they have a DHCP server the WAP will forward their addresses to us so we can conduct further reconnaissance on their LAN from a remote point. One method of gaining access, includes hiring on as a temp for

them or one of their vendors or business partners who has access, if access can be made to the computer room we may be able to steal a harddrive or other computer gear which may yield further information. Another method of access may be through a tour. Some places offer tours to the public that are usually not very well supervised and may include sensitive areas.

One particular attack comes to mind, if we can insert a boot disk into a server and boot it we may be able to copy the NT SAM database to the disk. Using Lophtrcrack on this off premises will give us all the NT usernames and passwords. If we have a list of employees at this point we may also want to drive by their residences to see if they have any WAP's in use. A break-in of their residence and/or vehicle may also yield a laptop with sensitive information on it. Most places shred their documents these days so we won't consider dumpster diving.

By now we have probably found some valid e-mail addresses and are now ready to try to harvest some username passwords. We will send an e-mail pretending to be from the help desk. We will ask them to visit a web site, which we control. However the link will appear to be the corporate web site. The text of the e-mail attack could be something like this.

**From:** Helpdesk  
**Sent:** Thursday, January 15, 2004 1:24 PM  
**To:** joeuser  
**Subject:** Assistance required

Good afternoon,

In a recent effort to consolidate our online servers we require end user assistance. Please follow the link below and login to Web Access Mail to update your user name and password on the new server. If you have any questions regarding this, please reply to this email and the Help Desk will provide you with support.

Thank you for your time.

GIAC IT Dept.

Click here: <https://owa.giac.com/exchange><sup>17</sup>

Other variations of this attack have been making the rounds, such as sending a user a Trojan claiming it is a patch from Microsoft.

### Scanning

The scanning phase involves using the information gained from the reconnaissance to further map the network. The use of war-dialing tools to find modems, scanning for live hosts, finding the open ports on the hosts, all these techniques as described by Skoudis will further our knowledge of the network we are attacking and any one may reveal a key vulnerability to exploit. Other techniques or tools of the attacker include, fingerprinting the public servers to reveal the OS. Grabbing banners to reveal the versions of the applications they

are hosting. Determining the firewall rules. As an attacker we will want to do all the above and if possible at the same time evade any IDS.

### **Gaining and Keeping Access**

In his book Counter Hack, Ed Skoudis describes two broad categories for gaining access. The Application and operating system attacks are the first category. As an attacker we would want to find try any of the known buffer overflow attacks for the systems we have identified. The other techniques in this category include password and web-based attacks. We could use password guessing and cracking for any known usernames. The cracking mainly involves tools for cracking encrypted passwords, which we may have obtained in one to the two above phases.

The other category Network Attacks includes using sniffers, IP spoofing, session hijacking and tools for establishing backdoors. The sniffers would be used on an already compromised machine to gain more information. IP spoofing with source routing may allow us to impersonate trusted hosts. Session hijacking tools typically utilize ARP spoofing attacks to get a man in the middle session. Establishing backdoors using netcat is typically done on compromised hosts.

### **Covering Tracks**

Once the system is compromised we want to cover our tracks. Usually this is done by utilizing rootkits, and modifying logs. The rootkits modify system utilities that will hide the presence of the attackers tools.

## **4.3 Results**

Gaining access to an internal host will probably only succeed if we are able to gain a username password. Some of the above attacks might have yielded that information. Probably will take a combination of exploited hosts and techniques to gain access to an internal host.

## **4.4 Countermeasures**

### **Recon**

User education is most effective defense against social engineering. All of the users must be trained not to give out sensitive information. Computer racks and rooms need to be locked, with access given only to people who are authorized. Physical security must also be appropriate at the sites premises. It is difficult to prevent against web-based recon, but the appropriate security policies should be put in places to prevent the disclosure of sensitive information about your organization.

### **Scanning**

Remove all modems from your internal network. Harden the systems so unneeded services are not running. Run some of the vulnerability scanners

i.e. run Nessus against your servers. Perform this task on a regular basis with the updated Nessus plugins. The signatures on your IDS also need to be updated regularly

#### Gaining and Keeping Access, Covering Tracks

Once again keeping your systems patched is the best defense against stack-based buffer overflows. A strong password policy and/or a two form factor means of authentication can be very effective against attackers. Employ ingress filters to defeat spoofing and turn off source routing on the border router. A tool like Arpwatch can help guard against ARP spoofing. An automated Nmap scan can watch for new ports listening on hosts and alert us to compromises. File integrity checkers can be used to watch for alteration of system files, which is commonly done when rootkits are installed.

© SANS Institute 2004, Author retains full rights.

## Appendix

### IP Addressing

<b>Main Site</b>		
Internal Addresses		
Internal Server Subnet		192.168.21.0/24
Internal DNS		192.168.21.12
Exchange		192.168.21.23
MySql DB		192.168.21.25
Admin Users Subnet		192.168.22.0/24
RADIUS Server		192.168.22.12
Syslog Server		192.168.22.22
SNMP Manager		192.168.22.23
FW-1 Mgmt Station		192.168.22.24
Internal Users Subnet		192.168.23.0/24
http/ftp proxy		192.168.23.7
MiddleWare Server		192.168.23.9
SurfControl		192.168.23.11
Screened Subnet		
coyote-eth1		85.112.229.1
Web Server		85.112.229.5
Mail Relay		85.112.229.9
External DNS		85.112.229.13
Reverse Proxy		85.112.229.17
FTP Server		85.112.229.21
Terminal Services Server		85.112.229.25
Point to Point		
coyote-eth2	(to crowsfoot-fa0/1)	192.168.20.1/30
crowsfoot-l0		192.168.20.5
crowsfoot-fa0/1	(to coyote-eth2)	192.168.20.2/30
crowsfoot-fa0/2	(to sundance untrust)	192.168.20.9/30
crowsfoot-fa0/13	(to littlehoop untrust)	192.168.20.13/30
Border Router		
wineta-l0		85.112.229.45
wineta-s3/0	(to hotsite chugwater-s3/0)	85.112.229.49
wineta-fa0/1	(to coyote-eth0)	85.112.229.57
wineta-fa0/0	(to Internet)	85.112.229.61
WAP	(to Wineta-e1/0)	85.112.229.225
coyote-eth0	(to border router wineta-fa0/0)	85.112.229.226
wineta-e1-0	(to WAP)	85.112.229.53
wineta-e1/1	(to mexicanhat)	85.112.229.65
WAP Users Subnet		85.112.229.224/27



© SANS Institute 2004, Author retains full rights.

## Firewall Policy

NO.	SOURCE	DESTINATION	SERVICE	ACTION	TRACK	INSTALL ON
<b>Firewall Implied Rules Replacement (Rules 1-2)</b>						
1	* Any	coyote.giac.net	TCP FW1_pslogon_NG UDP FW1_scv_keep_alive UDP IKE UDP tunnel_test TCP FW1_topo	accept	- None	* Policy Targets
2	Mgmt_Station	coyote.giac.net	TCP CPD_amon TCP CPD	accept	- None	* Policy Targets
<b>Internal and External DNS (Rules 3-5)</b>						
3	Internal_DNS	External_DNS	TCP domain-tcp UDP domain-udp	accept	- None	* Policy Targets
4	External_DNS	GIAC_Internal_Networks	TCP domain-tcp UDP domain-udp	accept	- None	* Policy Targets
5	GIAC_Internal_Networks	External_DNS	TCP domain-tcp UDP domain-udp	accept	- None	* Policy Targets
<b>Customer Access (Rules 6-7)</b>						
6	* Any	Reverse_http_Proxy	TCP https TCP http	accept	Log	* Policy Targets
7	GIAC_Web_Servers	Middle_Ware	TCP winframe	accept	Log	* Policy Targets
<b>Internal Users http and ftp Access (Rules 8-12)</b>						
8	http-ftp-proxy	* Any	HTTP http->SurfControl_Allow_URI	accept	Log	* Policy Targets
9	http-ftp-proxy	* Any	HTTP http->Surf_Control_Block_URI	drop	Log	* Policy Targets
10	http-ftp-proxy	* Any	HTTP http->Coyote_http_URI_Proxy	accept	Log	* Policy Targets
11	http-ftp-proxy	* Any	HTTP https->Coyote_https_URI_Proxy	accept	Log	* Policy Targets
12	http-ftp-proxy	* Any	FTP ftp->Coyote_ftp_Proxy	accept	Log	* Policy Targets
<b>Mobile and Teleworker Access (Rules 13-17)</b>						
13	MobileUser@Any	Terminal_Services_Server	TCP MS-Terminal-Services	Client Encrypt	Log	* Policy Targets
14	Terminal_Services_Server	http-ftp-proxy	TCP http TCP https TCP ftp	accept	Log	* Policy Targets
15	Terminal_Services_Server	Internal_Servers	NBT TCP microsoft-ds	accept	Log	* Policy Targets
16	Terminal_Services_Server	Exchange	TCP dce-rpc TCP rpc1189 TCP rpc1314	accept	Log	* Policy Targets
17	Terminal_Services_Server	Internal_DNS	TCP domain-tcp UDP domain-udp	accept	Log	* Policy Targets
<b>Partners and Suppliers Access (Rule 18)</b>						
18	Partners_and_Suppliers	ftp_Server	TCP ftp	Encrypt	Log	* Policy Targets
<b>E-mail Access (Rules 19-20)</b>						
19	Exchange Mail_Relay	Exchange Mail_Relay	TCP smtp	accept	Log	* Policy Targets
20	Mail_Relay Postini_Mail_Services	Postini_Mail_Services Mail_Relay	TCP smtp	accept	Log	* Policy Targets

James D. Rauser  
 GIAC Certified Firewall Analyst (GCFW) – Practical Assignment – v 2.0

NO.	SOURCE	DESTINATION	SERVICE	ACTION	TRACK	INSTALL ON
<b>Border Router VPN (Rules 21-24)</b>						
21	wineta_lo0	GIAC_Internal_Admin	UDP syslog UDP snmp-trap	Encrypt	Log	* Policy Targets
22	wineta_lo0	Internal_RADIUS	UDP NEW-RADIUS UDP NEW-RADIUS-ACCT	Encrypt	Log	* Policy Targets
23	GIAC_Internal_Admin	wineta_lo0	TCP ssh UDP snmp	Encrypt	Log	* Policy Targets
24	crowsfoot-fa0-13	wineta_lo0	UDP ntp-udp	Encrypt	Log	* Policy Targets
<b>Database Sync (Rule 25)</b>						
25	Main_Site_DB Hot_Site_DB	Hot_Site_DB Main_Site_DB	TCP mysql	Encrypt	Log	* Policy Targets
<b>Misc and Admin Access (Rules 26-30)</b>						
26	coyote_eth2 wAP	Internal_RADIUS	UDP NEW-RADIUS UDP NEW-RADIUS-ACCT	accept	Log	* Policy Targets
27	GIAC_Internal_Admin	GIAC_Screened_Subnet coyote_eth2 wAP	TCP ssh	accept	Log	* Policy Targets
28	SurfControl http-ftp-proxy	www.trendmicro.com www.surfcontrol.com	TCP http	accept	Log	* Policy Targets
29	coyote.giac.net	* Any	* Any	accept	Log	* Policy Targets
30	* Any	* Any	* Any	drop	Log	* Policy Targets

NO.	ORIGINAL PACKET			TRANSLATED PACKET			INSTALL ON
	SOURCE	DESTINATION	SERVICE	SOURCE	DESTINATION	SERVICE	
1	GIAC_Internal_Networks	GIAC_Screened_Subnet	* Any	Original	Original	Original	* Policy Targets
2	GIAC_Internal_Admin	wAP	* Any	Original	Original	Original	* Policy Targets
3	GIAC_Internal_Admin	wineta_lo0	TCP ssh	Original	Original	Original	* Policy Targets
4	GIAC_Internal_Admin	wineta_lo0	UDP snmp	Original	Original	Original	* Policy Targets
5	crowsfoot-fa0-13	wineta_lo0	UDP ntp-udp	Original	Original	Original	* Policy Targets
6	Main_Site_DB	Hot_Site_DB	TCP mysql	Original	Original	Original	* Policy Targets
7	GIAC_Internal_Networks	* Any	* Any	coyote.giac.net	Original	Original	* Policy Targets

## Component Descriptions

### Primary Firewall/VPN

CheckPoint NG with Application Intelligence (R55)  
Hotfix 017 - Build 009

SecurePlatform 2.4.9-42cp

#### Mgmt Station

CheckPoint SmartConsole R55

Microsoft Windows 2000  
5.00.2195  
Service Pack 4

Security Role: Perimeter Security

### Secondary Firewalls

Netscreen-5GT

ScreenOS 5.0.0r4.1 (Firewall+VPN)

Security Role: Defense in depth and separation of resources.

### Border Router

Cisco 3640 Series

Cisco Internetwork Operating System Software  
IOS (tm) 3600 Software (C3640-IK9S-M), Version 12.2(11)T, RELEASE  
SOFTWARE (fc1) Copyright (c) 1986-2002 by cisco Systems, Inc.

Security Role: Perimeter Security. Since it is the main gateway into our networks we use its filtering capabilities to supplement our defense in depth strategy.

### Reverse-Proxy

Squid Proxy Server Version 2.5

Fedora Core 2.4.22-1.2115.nptl

Security Role: Perimeter Security, Defense in Depth

### http-ftp-proxy

Trend Micro InterScan VirusWall NT 3.52

Microsoft Windows 2000  
5.00.2195  
Service Pack 4

Security Role: Separation of Resources

#### Mail Relay

Trend Micro InterScan VirusWall NT 3.52

Microsoft Windows 2000  
5.00.2195  
Service Pack 4

Security Role: Separation of Resources

#### Internal DNS

Microsoft Windows DNS

Microsoft Windows 2000  
5.00.2195  
Service Pack 4

Security Role: Separation of Resources

#### External DNS

BIND 9.2.2-P3

Fedora Core 2.4.22-1.2115.nptl

Security Role: Separation of Resources

#### Intrusion Detection

Snort 2.1.1

Fedora Core 2.4.22-1.2115.nptl

Security Role: Identify attacks, suspicious activity and vulnerabilities.

#### RADIUS

Funk Software Steel-Belted Radius Release 4.7

Microsoft Windows 2000  
5.00.2195  
Service Pack 3

Security Role: Perimeter Security. SecureClient uses this for authentication as does our 802.1x client for layer 2 access to the wireless.

#### EAP-Client

Funk Software Odyssey Client Ver. 2.8

Microsoft Windows 2003

Security Role: Perimeter Security. Protects against unauthorized access to the wireless.

#### WAP

Cisco Aironet 350 Series Wireless Access Point

Cisco Internetwork Operating System Software  
IOS (tm) C350 Software (C350-K9W7-M), Version 12.2(13)JA1, EARLY  
DEPLOYMENT RELEASE SOFTWARE (fc1)TAC Support:  
<http://www.cisco.com/tac>  
Copyright (c) 1986-2003 by cisco Systems, Inc.

Security Role: Separation of Resources. It accomplishes this by keeping the roaming laptops off our Internal Networks. Instead assigning them to an external wireless network. This WAP's security role also includes Perimeter Security.

## Wap Configuration

```
sh conf
Using 4322 out of 32768 bytes
!
version 12.2
no service pad
service timestamps debug datetime msec show-timezone
service timestamps log datetime msec show-timezone
service password-encryption
service linenumber
!
hostname WAP
!
logging buffered 16000 debugging
logging console critical
enable secret 5 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!
username XXXXXXXXXX privilege 15 password 7 XXXXXXXXXXXXXXXX
clock timezone CST -6
clock summer-time cdt recurring
ip subnet-zero
no ip source-route
ip domain name giac.net
ip dhcp excluded-address 85.112.229.224
ip dhcp excluded-address 85.112.229.225
ip dhcp excluded-address 85.112.229.226
ip dhcp bootp ignore
!
ip dhcp pool default
    network 85.112.229.224 255.255.255.224
    domain-name giac.net
    default-router 85.112.229.225
    netbios-node-type h-node
    lease 8
!
ip ssh time-out 60
ip ssh authentication-retries 2
aaa new-model
!
aaa group server radius rad_eap
    server 192.168.22.12 auth-port 1812 acct-port 1813
!
aaa group server radius rad_mac
!
aaa group server radius rad_acct
```

```
!  
aaa group server radius rad_admin  
  server 192.168.22.12 auth-port 1812 acct-port 1813  
!  
aaa group server tacacs+ tac_admin  
!  
aaa group server radius rad_pmip  
!  
aaa group server radius dummy  
!  
aaa group server radius rad_eap2  
  server 192.168.22.12 auth-port 1812 acct-port 1813  
!  
aaa authentication login eap_methods group rad_eap  
aaa authentication login mac_methods local  
aaa authentication login eap_methods2 group rad_eap2  
aaa authentication login radius-login group radius local  
aaa authorization exec default if-authenticated local  
aaa authorization ipmobile default group rad_pmip  
aaa accounting exec default start-stop group radius  
aaa accounting network default start-stop group radius  
aaa accounting network acct_methods start-stop group rad_acct  
aaa session-id common  
!  
bridge irb  
!  
!  
interface Dot11Radio0  
  no ip address  
  no ip route-cache  
  !  
  encryption key 1 size 128bit 7 290DFAA0A3D887F9B798DAF8E0CD transmit-  
key  
  encryption mode wep mandatory  
  !  
  ssid f57e6wphwpp  
    authentication open eap eap_methods2  
    authentication network-eap eap_methods2  
    infrastructure-ssid  
  !  
  speed basic-1.0 basic-2.0 basic-5.5 basic-11.0  
  rts threshold 2312  
  station-role root  
  bridge-group 1  
  bridge-group 1 subscriber-loop-control  
  bridge-group 1 block-unknown-source
```



```
no bridge-group 1 source-learning
no bridge-group 1 unicast-flooding
bridge-group 1 spanning-disabled
!
interface FastEthernet0
ip address 85.112.229.226 255.255.255.224
no ip redirects
no ip unreachable
no ip proxy-arp
no ip route-cache
speed 10
full-duplex
no cdp enable
bridge-group 1
no bridge-group 1 source-learning
bridge-group 1 spanning-disabled
!
interface BV11
ip address 85.112.229.226 255.255.255.224
no ip route-cache
!
ip default-gateway 85.112.229.225
no ip http server
ip http help-path
http://www.cisco.com/warp/public/779/smbiz/prodconfig/help/eag/ivory/1100
ip radius source-interface FastEthernet0
access-list 182 permit tcp 192.168.22.0 0.0.0.255 any
access-list 182 deny ip any any log
no cdp run
radius-server host 192.168.22.12 auth-port 1812 acct-port 1813 key 7
XXXXXXXXXXXXXXXXXX
radius-server retransmit 1
radius-server deadtime 5
radius-server attribute 32 include-in-access-req format %h
radius-server key 7 XXXXXXXXXXXXXXXXXXXXXXXXXXXX
radius-server authorization permit missing Service-Type
radius-server vsa send accounting
bridge 1 route ip
!
banner login ^C
%%%%%%%%%%
%%%%%%%%%%
%
% !!!!!!!!!!! WARNING !!!!!!!!!!!
% You are attempting to access a protected device! %
% Unauthorized access is prohibited! %
```

```
% Your session is being monitored and logged! %  
% You will be prosecuted for any illegal activity! %  
%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%^C  
!  
line con 0  
exec-timeout 30 0  
login authentication radius-login  
stopbits 1  
line vty 0 4  
access-class 182 in  
exec-timeout 30 0  
login authentication radius-login  
transport input ssh  
line vty 5 15  
access-class 182 in  
exec-timeout 30 0  
login authentication radius-login  
transport input ssh  
!  
end
```

© SANS Institute 2004, Author retains full rights.

## EAP-TTLS RADIUS Server Configuration

### 1. Get certificate from root CA

#### A. Get root CA certificate for RADIUS server (Win2000)

#### B. Get server certificate

Advanced (next)

Request a certificate (advanced request using a form) (Next)

Add name (server and function)

Purpose: Server Authentication Certificate

Key options:

CSP: Microsoft Enhanced Cryptographic Provider v1.0

Key usage: both

Size: 1024

Check mark keys as exportable

Once issued retrieve the key

#### C. Approve Certificate

On CA Server

Use CA tool to approve request

Exit

From browser on (requesting host)

Use browser to check on pending certificate

Install Certificate

Click yes, you should see your new certificate has been installed.

#### D. Server Certificate Setup

Using the browser (on the RADIUS server that issued the request), under personal certificates select the key and export it.

Check export the private key. Personal info exchange PKCS#12 .PFX. Don't enable Strong protection. Type and confirm password. Export to C:\radius\service\cert\radiusservercert.pfx

### 2. Edit files on Radius server C:\radius\service

#### A. certinfo.ini

Certificate\_And\_Private\_Key\_File=c:\radius\service\cert\radiusservercert.pfx

; Specifies the password with which the private key contained in the PKCS#12; file mentioned above was encrypted.

Password=XXXXXXXXXXXX

B. Edit radius.ini

Server\_Certificate\_Info\_File=c:\radius\service\certInfo.ini

C. eap.ini

uncomment the following (not needed in version 4.52)

```
[ttlsauth]
EAP-Only = 1
EAP-Type = TTLS
First-Handle-Via-Auto-EAP = 0
```

D. ttlsauth.aut

A. enable loading of EAP-TTLS module

```
[Bootstrap]
LibraryName=ttlsauth.dll
Enable=1
InitializationString=EAP-TTLS
```

B. [Session\_Resumption]

Specifies the maximum length of time (in seconds) the NAS/AP ; will be instructed to allow the session to persist before the client is asked to re-authenticate. Specifying a 0 will cause the Session-Timeout attribute not to be generated by the plug-in. The default is 0.

```
Session_Timeout = 600
```

C. to enable checklist attributes uncomment in the [Request\_Filters] sections

```
Transfer_Outer_Attribs_to_New = ttls_transfer_outer_to_new
```

Edit filters.ini add a newline and the following section:

```
[ttls_reject]
Allow
Exclude EAP-Message
[ttls_transfer_outer_to_new]
Exclude
Allow NAS-IP-Address
```

4. Stop start RADIUS

activate EAP-TTLS from GUI move to top of the list

activate Windows Domain User

activate Windows Domain Group

5. Create RADIUS clients and users.

## NMAP Scans

### External Host to Firewall

```
[root@mexicanhat root]# nmap -sS -P0 -v 85.112.229.58
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host (85.112.229.58) appears to be up ... good.
Initiating SYN Stealth Scan against (85.112.229.58)
Adding open port 264/tcp
The SYN Stealth Scan took 632 seconds to scan 1601 ports.
Interesting ports on (85.112.229.58):
(The 1600 ports scanned but not shown below are in state: filtered)
Port      State      Service
264/tcp   open       bgmp
Nmap run completed -- 1 IP address (1 host up) scanned in 633 seconds
[root@mexicanhat root]#
```

```
[root@mexicanhat root]# nmap -sU -P0 -v 85.112.229.58
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host (85.112.229.58) appears to be up ... good.
Initiating UDP Scan against (85.112.229.58)
(no udp responses received -- assuming all ports filtered)
All 1468 scanned ports on (85.112.229.58) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 1771
seconds
[root@mexicanhat root]#
```

```
[root@mexicanhat root]# nmap -sP -v 85.112.229.58
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host (85.112.229.58) appears to be down.
Note: Host seems down. If it is really up, but blocking our ping probes, try
-P0
Nmap run completed -- 1 IP address (0 hosts up) scanned in 30 seconds
[root@mexicanhat root]#
```

### External Host to Reverse-Proxy Subnet

The following scan took approximately 16 hours

```
[root@mexicanhat root]# nmap -sS -P0 -v 85.112.229.0/27
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
```

Host (85.112.229.0) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.0)  
The SYN Stealth Scan took 1721 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.0) are: filtered  
Host (85.112.229.1) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.1)  
Adding open port 264/tcp  
The SYN Stealth Scan took 660 seconds to scan 1601 ports.  
Interesting ports on (85.112.229.1):  
(The 1600 ports scanned but not shown below are in state: filtered)

Port	State	Service
264/tcp	open	bgmp

Host (85.112.229.2) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.2)  
The SYN Stealth Scan took 1721 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.2) are: filtered

Host (85.112.229.3) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.3)  
The SYN Stealth Scan took 1721 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.3) are: filtered

Host (85.112.229.4) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.4)  
The SYN Stealth Scan took 1722 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.4) are: filtered

Host (85.112.229.5) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.5)  
The SYN Stealth Scan took 1722 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.5) are: filtered

Host (85.112.229.6) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.6)  
The SYN Stealth Scan took 1722 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.6) are: filtered

Host (85.112.229.7) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.7)  
The SYN Stealth Scan took 1722 seconds to scan 1601 ports.

All 1601 scanned ports on (85.112.229.7) are: filtered

Host (85.112.229.8) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.8)  
The SYN Stealth Scan took 1721 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.8) are: filtered

Host (85.112.229.9) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.9)  
The SYN Stealth Scan took 1721 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.9) are: filtered

Host (85.112.229.10) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.10)  
The SYN Stealth Scan took 1722 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.10) are: filtered

Host (85.112.229.11) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.11)  
The SYN Stealth Scan took 1722 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.11) are: filtered

Host (85.112.229.12) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.12)  
The SYN Stealth Scan took 1722 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.12) are: filtered

Host (85.112.229.13) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.13)  
The SYN Stealth Scan took 1722 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.13) are: filtered

Host (85.112.229.14) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.14)  
The SYN Stealth Scan took 1722 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.14) are: filtered

Host (85.112.229.15) appears to be up ... good.

Initiating SYN Stealth Scan against (85.112.229.15)  
The SYN Stealth Scan took 1721 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.15) are: filtered

Host (85.112.229.16) appears to be up ... good.

Initiating SYN Stealth Scan against (85.112.229.16)

The SYN Stealth Scan took 1721 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.16) are: filtered

Host (85.112.229.17) appears to be up ... good.

Initiating SYN Stealth Scan against (85.112.229.17)

The SYN Stealth Scan took 431 seconds to scan 1601 ports.

Interesting ports on (85.112.229.17):

(The 1599 ports scanned but not shown below are in state: filtered)

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

Host (85.112.229.18) appears to be up ... good.

Initiating SYN Stealth Scan against (85.112.229.18)

The SYN Stealth Scan took 1722 seconds to scan 1601 ports.

All 1601 scanned ports on (85.112.229.18) are: filtered

Host (85.112.229.19) appears to be up ... good.

Initiating SYN Stealth Scan against (85.112.229.19)

The SYN Stealth Scan took 1722 seconds to scan 1601 ports.

All 1601 scanned ports on (85.112.229.19) are: filtered

Host (85.112.229.20) appears to be up ... good.

Initiating SYN Stealth Scan against (85.112.229.20)

The SYN Stealth Scan took 1722 seconds to scan 1601 ports.

All 1601 scanned ports on (85.112.229.20) are: filtered

Host (85.112.229.21) appears to be up ... good.

Initiating SYN Stealth Scan against (85.112.229.21)

The SYN Stealth Scan took 1722 seconds to scan 1601 ports.

All 1601 scanned ports on (85.112.229.21) are: filtered

Host (85.112.229.22) appears to be up ... good.

Initiating SYN Stealth Scan against (85.112.229.22)

The SYN Stealth Scan took 1721 seconds to scan 1601 ports.

All 1601 scanned ports on (85.112.229.22) are: filtered

Host (85.112.229.23) appears to be up ... good.

Initiating SYN Stealth Scan against (85.112.229.23)

The SYN Stealth Scan took 1721 seconds to scan 1601 ports.



All 1601 scanned ports on (85.112.229.23) are: filtered

Host (85.112.229.24) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.24)  
The SYN Stealth Scan took 1722 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.24) are: filtered

Host (85.112.229.25) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.25)  
The SYN Stealth Scan took 1722 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.25) are: filtered

Host (85.112.229.26) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.26)  
The SYN Stealth Scan took 1722 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.26) are: filtered

Host (85.112.229.27) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.27)  
The SYN Stealth Scan took 1721 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.27) are: filtered

Host (85.112.229.28) appears to be up ... good.  
Initiating SYN Stealth Scan against (85.112.229.28)  
The SYN Stealth Scan took 1721 seconds to scan 1601 ports.  
All 1601 scanned ports on (85.112.229.28) are: filtered

.....

Internal Admin Network to Firewall

```
[root@windriver root]# nmap -sS -P0 -v 192.168.20.1
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host (192.168.20.1) appears to be up ... good.
```

```
Initiating SYN Stealth Scan against (192.168.20.1)
Adding open port 264/tcp
Adding open port 22/tcp
The SYN Stealth Scan took 462 seconds to scan 1601 ports.
Interesting ports on (192.168.20.1):
(The 1599 ports scanned but not shown below are in state: filtered)
Port      State  Service
22/tcp    open   ssh
264/tcp   open   bgmp
Nmap run completed -- 1 IP address (1 host up) scanned in 462 seconds
[root@windriver root]#

[root@windriver root]# nmap -sU -P0 -v 192.168.20.1
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host (192.168.20.1) appears to be up ... good.
Initiating UDP Scan against (192.168.20.1)
(no udp responses received -- assuming all ports filtered)
All 1468 scanned ports on (192.168.20.1) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 1772
seconds
[root@windriver root]#

[root@windriver root]# nmap -sP -v 192.168.20.1
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host (192.168.20.1) appears to be down.
Note: Host seems down. If it is really up, but blocking our ping probes, try
-P0
Nmap run completed -- 1 IP address (0 hosts up) scanned in 30 seconds
[root@windriver root]#
```

#### Internal User Network to Firewall

```
weston:[/usr/local/rpm]$nmap -sS -P0 -v 192.168.20.1
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-04-26
16:50 CDT
Host 192.168.20.1 appears to be up ... good.
Initiating SYN Stealth Scan against 192.168.20.1 at 16:50
Adding open port 264/tcp
Adding open port 21/tcp
Adding open port 80/tcp
The SYN Stealth Scan took 90 seconds to scan 1659 ports.
Interesting ports on 192.168.20.1:
(The 1655 ports scanned but not shown below are in state: filtered)
PORT      STATE SERVICE
```

```
21/tcp open  ftp
80/tcp open  http
264/tcp open bgmp
443/tcp closed https
Nmap run completed -- 1 IP address (1 host up) scanned in 89.814
seconds
```

```
weston:[/usr/local/rpm]$nmap -sU -P0 -v 192.168.20.1
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-04-26
17:31 CDT
Host 192.168.20.1 appears to be up ... good.
Initiating UDP Scan against 192.168.20.1 at 17:31
(no udp responses received -- assuming all ports filtered)
All 1478 scanned ports on 192.168.20.1 are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 1785.183
seconds
weston:[/usr/local/rpm]$
```

```
weston:[/usr/local/rpm]$nmap -sP -v 192.168.20.1
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-04-26
16:58 CDT
Host 192.168.20.1 appears to be down.
Note: Host seems down. If it is really up, but blocking our ping probes, try
-P0
Nmap run completed -- 1 IP address (0 hosts up) scanned in 12.044
seconds
weston:[/usr/local/rpm]$
```

### Screened Subnet to Firewall

```
[root@teton root]# nmap -sS -P0 -v 85.112.229.1
Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-04-25
18:14 CDT
Host 85.112.229.1 appears to be up ... good.
Initiating SYN Stealth Scan against 85.112.229.1 at 18:14
Adding open port 264/tcp
The SYN Stealth Scan took 106 seconds to scan 1657 ports.
Interesting ports on 85.112.229.1:
(The 1655 ports scanned but not shown below are in state: filtered)
PORT      STATE SERVICE
53/tcp    closed domain
264/tcp   open  bgmp
```

```
Nmap run completed -- 1 IP address (1 host up) scanned in 106.689 seconds
```

```
[root@teton root]# nmap -sU -P0 -v 85.112.221.1  
Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-04-26 18:08 CDT  
Host 85.112.221.1 appears to be up ... good.
```

```
Initiating UDP Scan against 85.112.221.1 at 18:08
```

```
(no udp responses received -- assuming all ports filtered)  
All 1478 scanned ports on 85.112.221.1 are: filtered  
Nmap run completed -- 1 IP address (1 host up) scanned in 1784.800 seconds  
[root@teton root]#
```

```
[root@teton root]# nmap -sP -v 85.112.221.1  
Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-04-26 18:06 CDT  
Host 85.112.221.1 appears to be down.  
Note: Host seems down. If it is really up, but blocking our ping probes, try -P0  
Nmap run completed -- 1 IP address (0 hosts up) scanned in 12.062 seconds  
[root@teton root]#
```

© SANS Institute 2004

## References

- <sup>1</sup> Funk Software. Steel Belted RADIUS Enterprise Edition Administrative Guide. 2003. Funk Software Inc.
- <sup>2</sup> Center for Internet Security. Gold Standard Benchmark for Cisco IOS. Sept. 2003. URL: [www.cisecurity.org](http://www.cisecurity.org).
- <sup>3</sup> Northcutt, Stephen. Inside Network Perimeter Security: The Definitive Guide to Firewalls, VPNs, Routers and Intrusion Detection Systems. New Riders Publishing. July 2003.
- <sup>4</sup> Ziegler, Robert. Linux Firewalls. 2<sup>nd</sup> Edition. New Riders Publishing. 2002.
- <sup>5</sup> Brenton, Chris Hunt, Cameron. Mastering Network 2<sup>nd</sup> Security Edition. Sybex Inc. 2003.
- <sup>6</sup> Center for Internet Security. Router Auditing Tool and ncat\_config. Sept. 2003. URL: [www.cisecurity.org](http://www.cisecurity.org).
- <sup>7</sup> National Security Agency. Router Security Configuration Guide Sept. 2002
- <sup>8</sup> Cisco Systems Inc. Configuring an IPSEC Tunnel Between a Cisco Router and a CheckPoint NG. URL <http://www.cisco.com/warp/public/707/ipsec-checkpt.pdf>
- <sup>9</sup> Cisco Systems Inc. Triple DES Encryption for IPSEC.  
URL  
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t2/3desips.pdf>
- <sup>10</sup> Barisani Andrea. Firewall Tester. June 2003.  
URL: <http://www.infis.univ.trieste.it/~lcars/ftester>
- <sup>11</sup> Parkin, Miles. GIAC Certified Firewall Analyst Practical Assignment v2.0. Nov. 2003. URL [http://www.giac.org/practical/GCFW/Miles\\_Parkin\\_GCFW.pdf](http://www.giac.org/practical/GCFW/Miles_Parkin_GCFW.pdf)
- <sup>12</sup> Internet Security Systems. Security Advisory. Checkpoint Firewall-1 HTTP Parsing Format String Vulnerabilities. Feb. 2004.  
URL <http://xforce.iss.net/xforce/alerts/id/162>
- <sup>13</sup> The MITRE Corporation. “Common Vulnerabilities and Exposures (CVE).” CAN-2004-0039. Jan. 2004.  
URL: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0039>

<sup>14</sup> Gibson, Steve. The Strange Tale of the Denial of Service Attacks Against GRC.com. June 2001. URL: <http://grc.com/dos/grcdos.htm>

<sup>15</sup> hypnosis@mbnet.fi From an article on [www.netsys.com](http://www.netsys.com)  
URL: <http://www.netsys.com/library/papers/ddos-ircbot.txt>

<sup>16</sup> Skoudis, Ed. Counter Hack A Step by Step Guide to Computer Attacks and Effective Defenses. 2002 Prentice Hall.

<sup>17</sup> lss-com. From a recent pen test. January 15, 2004.

© SANS Institute 2004, Author retains full rights