Global Information Assurance Certification Paper

Security Proposal for GIAC Enterprises
SANS Firewalls, Perimeter Protection, and VPNs
GCFW Practical Assignment Version 1.6a
Stig Ravdal

Assignment 1 – Security Architecture (15 points)

Define a security architecture for GIAC Enterprises, an e-business which deals in the online sale of fortune cookie sayings.

Your architecture **must** consider access requirements (and restrictions) 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);
- GIAC Enterprises (the employees located on GIAC's internal network).

You **must** explicitly define how the business operations of GIAC Enterprises will take place. How will each of the groups listed above connect to or communicate with GIAC Enterprises? How will GIAC employees access the outside world? What services, protocols, or applications will be used?

Defining what type of access is required and why is a critical part of this assignment. If you have not thought through how this access will take place, you will not be able to adequately define your security policy and ACLs/rulesets later in the paper.

In designing your architecture, you **must** include the following components:

- filtering routers;
- firewalls;
- VPNs to business partners.

Your architecture **may** also include the following optional components if they are appropriate to your design:

- internal firewalls (are internal firewalls appropriate for additional, layered protection; to segment internal networks…?);
- secure remote access (is additional remote access required by administrators, salespeople, telecommuters…?).

Include a diagram or set of diagrams that shows the layout of GIAC Enterprises' network and the location of each component listed above. Provide the specific brand and version of each perimeter defense component used in your design. Finally, include an explanation that describes the purpose of each component, the security function or role it carries out, and how the placement of each component on the network allows it to fulfill this role.

The network can be as complex or as simple as you like as long as it meets the functional requirements that you define according to the guidelines given above. The important thing is not how elaborate your network is, but that your design actually works.

GIAC Enterprises

GIAC Enterprises is a small business that sells fortune cookie sayings. It employs a total of 53 people that handle all the activities of the company ranging from office administration, accounting and marketing to sales, printing and IT. The company netted a total of three million dollars in profits for the year 2001. Growth has been stable, but decreasing over the past several years, something that is not unusual for a company of this size operating in a very competitive market.

In an effort to reinvent the company the executive management of GIAC Enterprises came up with the enitiative campaign intended to make the growth curve steeper and take the company into the 21st century. The main thrust of the campaign is to establish an Internet and World Wide Web presence that allows for automated, secure and efficient transactions of fortune cookie sayings sold in bulk and submission of such sayings from a network of business partners. This initiative will allow the company to gain an advantage on its competitors is by increasing the means by which it can produce, market and sell products without incurring significant new operational costs.

To this end GIAC Enterprises has hired a security-consulting firm called OpenSecure, Inc. that is known for its secure, rapidly deployable and cost-efficient network designs.

Business Requirements

The following are the four main business requirements of the enitiative campaign:

Customers: Purchasing Online Fortunes in Bulk

Customers will be able to purchase fortune cookie sayings in bulk via automated online ordering. Using a web site as the interface customers can submit orders securely across the Internet using Secure Sockets Layer (SSL) and https. Customers have a choice of having their orders serviced in one of two ways: 1) On first order the customer receives a userID and random character password that to download the pre-formatted bulk fortune sayings via https/download. 2) The fortunes are printed and shipped within 2 days of order receipt (There are small margins to be earned on printing and shipping).

Authors: Suppliers of Fortunes

GIAC Enterprises relies on several freelance authors and subcontractors for the creation of new fortune cookie sayings. Working independently they do not have network infrastructures that will be connecting to GIAC's network. All access will be made via the Internet. Authors submitting fortune sayings by copying and pasting them into forms the web server using HTTPS.

Partners: Resell Fortunes and Translate Into Other Languages

GIAC Enterprises has several national and international partners that resell and translate the fortune cookie sayings into various languages. Thus, they require secure access to the sayings stored in GIAC Enterprises' database systems. To enable partners to securely retrieve the data stored in the GIAC databases partners will be required to implement network-to-network VPN connections with GIAC Enterprises. Using user accounts with read only permission they can retrieve the data packages. Which vendor a partner uses for their VPN solution is up to the partner as long as it is compatible and conforms with the system and standards set forth by GIAC Enterprises.

Employees: GIAC Employees Accessing Internet Resources

Employees of GIAC Enterprises are encourage to use the world wide web (WWW) and other Internet resources to perform research and accomplish other business related tasks. Several people in the sales team travel and will require network access from ephemeral remote locations. Additionally, employees may periodically work from home. To accommodate remote users accessing the network remote-to-site VPN tunneling will be employed.

Functional Requirements

To achieve the business goals set forth in the e-nitiative a suite services have to be offered over the internet as well as some underlying supporting services and systems. These have to be incorporated in the design.

Services that interact with Customers, Partners and Employees:
- Mail (SMTP, Exchange)
- Web (http, https)
- IPSec VPN (AH, IKE, ESP)
- Databases (indirectly through web site)

Underlying services necessary to support the network and perimeter:
- Domain Name Service (DNS - 53)
- Mail Relay (SMTP)
- Proxy
- Secure Sockets Layer (SSL - 443)

Services that support perimeter operation
- SSH (22)

Note: FTP is a service that is commonly available in many corporate networks. However, after discussing the matter with OpenSecure, Inc., GIAC Enterprises agreed that the risks outweighed the benefits of allowing the use of this service when there are many good alternatives. Thus, without a strong business case FTP service will not be offered.

GIAC Enterprises Information System Policy

1) Systems are intended for business use.
2) Activities that are inappropriate are prohibited.
   Note: While this is a policy that could be enforced the executive administration believes that strict monitoring of users activities on the web is not necessary. Spot checks will suffice to remind users of the policy and the executive administration believes that the relationship between the network stewards and its netizens benefits the honor system. Content monitoring is an administrative burden that GIAC Enterprises would like to avoid.
3) No modems are allowed to be connected to systems connected to the network.
4) Only those device explicitly controlled by the MIS department of GIAC enterprises shall have remote access via VPN or SSH to the corporate network (including the perimeter devices). Those systems that shall be secured with a personal firewall (software or hardware deemed acceptable by MIS), virus protection suite and VPN client controlled and configured by MIS. The only exception to this policy is for established partners that are given limited access to the network via VPN.
5) All systems accessing public or third party networks (including access to customer networks while onsite) are required to have a personal firewall and virus protection before such access is granted.
6) Any systems directly connected to the corporate LAN must be cleared with MIS. This includes all visitors.
7) Any new systems, applications or services must be cleared for use by MIS before connecting to the network or being installed on a system.

Guiding Principles

The following guiding principles will be followed where possible when designing the network architecture and perimeter defense systems (note the order of the principles does not imply a strict relation of importance).

- *Defense in depth*: There is no silver bullet to secure a business network infrastructure. GIAC Enterprises will employ layers of security devices that can slow an attack sufficiently enough to be noticed and to give the incident response team time enough to notify, evaluate and activate countermeasures before data is compromised or destroyed.
- *Compartmentalized Network Segments.* The entire network infrastructure will be compartmentalized to mitigate the effects of a successful perimeter breach.
- *Explicitly Deny*: Only services that are explicitly required for business functionality are available. All other services will be denied. Err on the side of stricter security.
- *It's All About The Bottom Line*: Cost and functionality are always factors in any proposed security measures that affect the business. The security risks must be weighed against the cost and affect on functionality.

However, upper management approval and signoff is required to change policy.

- *KISS –Keep It Simple Stupid*: Keep the infrastructure as simple as possible because this reduces management overhead (=costs), and it also reduces the risks of loopholes in the design.
- *Security is a process*: Regular, frequent audits, reviews and updates of all policies and systems discussed herein are necessary to maintain the desired level of security of the network infrastructure. That includes keeping all systems updated with the latest patches,
- *Be a Good Internet Neighbor*: Everyone on the Internet is part of a greater whole where everyone shares resources. To the best of our abilities, we shall not allow our systems and resources to be used to attack, harm or in any other way deny other Internetizens full access to the Internet (simply put we will perform egress filtering and keep our systems from be used to stage attacks against others).
- *Scalability*: As the main purpose of this business initiative is to produce a significant increased growth it is necessary that in mind for the design. This means that the scalability must also play a part when deciding what components are to be used. Growth can be expected in the following main areas: Bandwidth requirements, number of partners, and volume of traffic.
- *Open Source*: Where possible and practical we will use Open Source tools/systems to accomplish the component objectives of the security architecture. Open Source systems provide in many cases, more granular control of the rule sets and logging features, they are tried and tested by the masses, they do not suffer from "proprietary vulnerabilities", offera large support communities and keep the total cost of ownership low.

Security Domains

From the business requirements we have distinguished four major and four minor *security domains* that we will use as a basis to define relationships between those entities (Figure 1). Conceptually this makes the task of defining the rules governing traffic flow between all elements in the design easier.

Internet

This is a major component of the design as it will be the medium or pathway by which information flows to the company, to consumers, between the company and its partners and to some degree between the company and its employees. The most important aspect of this security domain is that it is untrusted.

Partner Network

Another major component of the design is the partner(s) network(s) (Note: For

the purpose of the design there is only one partner network, however in reality there may be many of these but they will all be handled similarly from a functional and implementation perspective). Partners are afforded a certain degree of trust and but because their systems are beyond the control GIAC Enterprises control caution is warranted.

Customer and Authors

Customers are a very important component in the design overall as the whole initiative is driven by creating more customers. This component of the architecture is the one we have the least control over but it also requires no direct access to the corporate network. Authors are functionally treated in the same way as customers.

Corporate Network

The corporate is the most important component of the whole design. It consist of several subcomponents:

1. DMZ
   This is the network segment between the border router and delimits the part of the network over which we have control. There is one access point to and from the Internet (Note: The access point can consist of more than one Pipe and ISP using core switches and routers with load balancing). There is some degree of control of what type and bandwidth of traffic is allowed to flow over the border router(s) but these are in the strictest sense not security devices.
2. Service Network
   This network segment is publicly accessible to only those services required by the new business model. The public and secure web site and secure transactions on the web-site. Additionally the domain name servers are hosted on this segment.
3. LAN
   This is the core of the corporate network. This is where most employee workstations, productions servers and other networked equipment are located.
4. Restricted LAN
   This is a restricted area of the LAN where the financial databases and systems are kept. Thus protecting from intentional or accidental access by employees and/or partners.

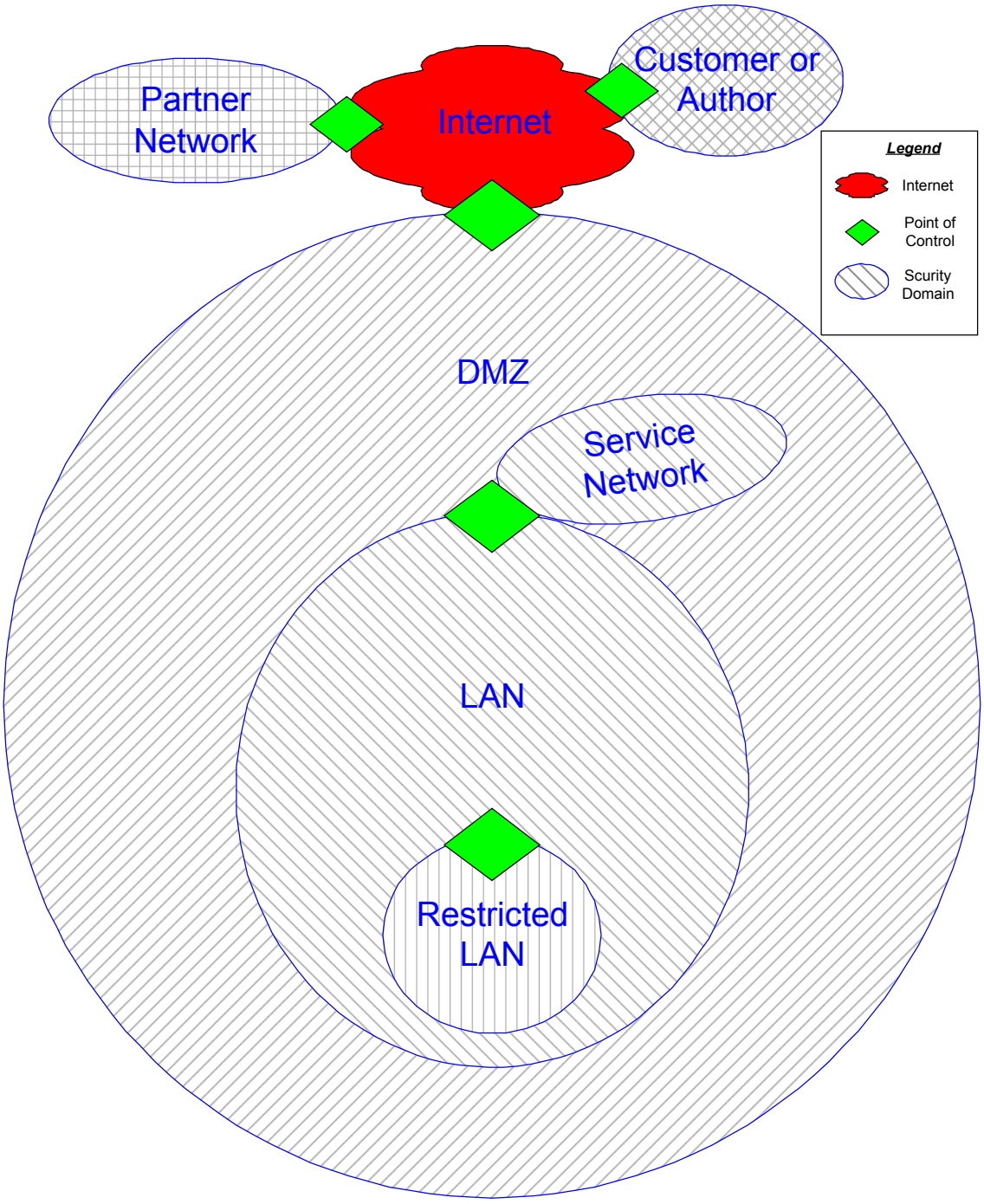


Figure 1: The security domains of the secure network and perimeter design. The green diamonds illustrate where the domains interact and where devices that control traffic flow in both directions must be implemented.

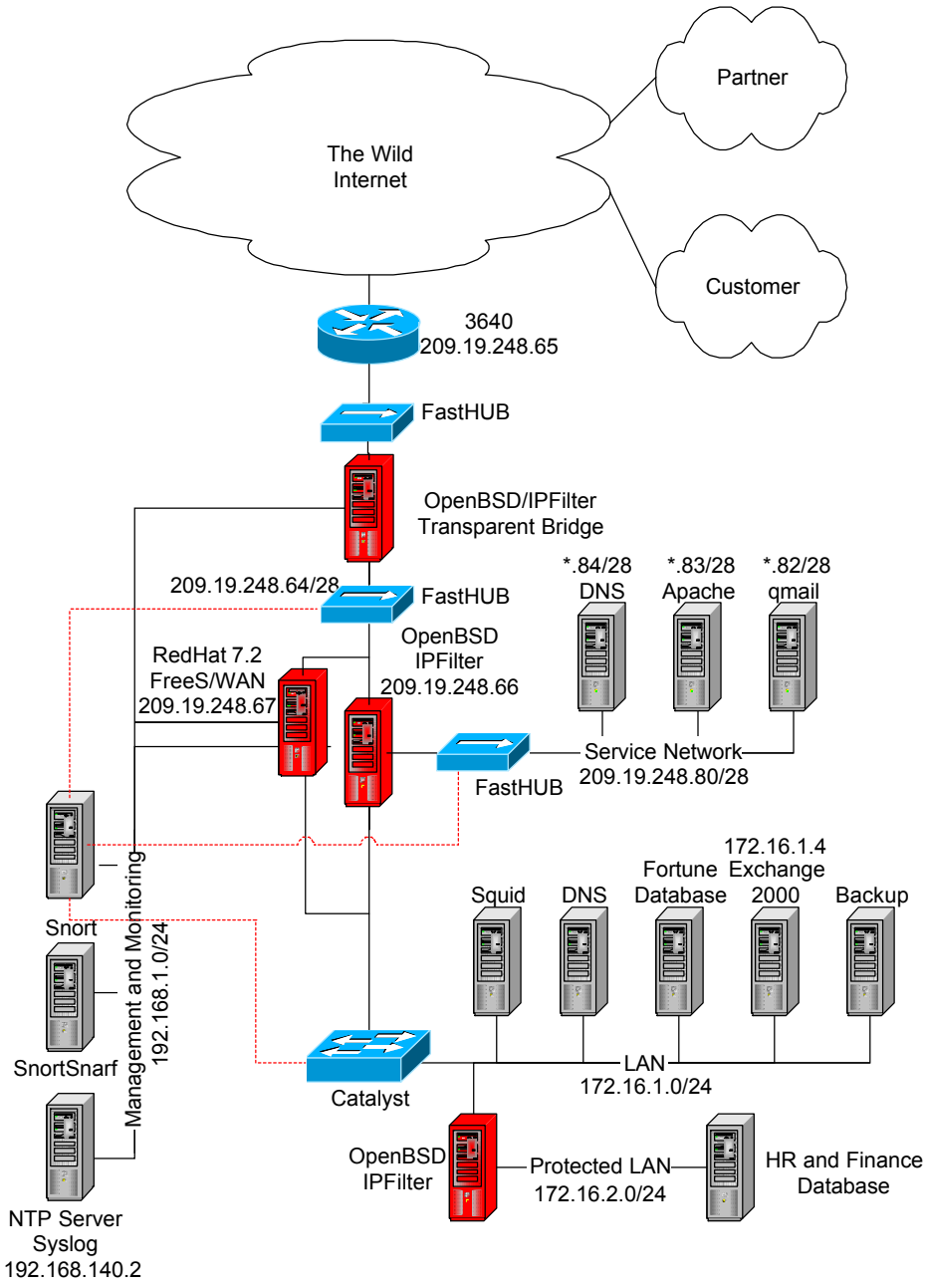


Figure 2: Diagram showing the layout of GIAC Enterprises network and perimeter defenses. IP network addresses for each segment have been included and additionally the IP address of the untrusted (public) side of the primary Firewall and VPN Gateway. For brevity several network components are not included on this diagram.

Description of Components

Border Router

This device delimits the boundary between the Internet and GIAC Enterprises network. As the first line of defense in the perimeter and a critical component of

the business model, it will be a Cisco 3640 router running the current version of IOS (12.2 as of this writing). The product is known to be reliable, it scales well with any organization, it offers a solid feature set and Cisco has excellent technical support. An added advantage with Cisco products is that it isn't hard to find skilled people to administer them.

Transparent Filtering Bridge

A transparent filtering bridge based on OpenBSD 3.0 and IPFilter will be implemented between the border router and the first firewall. The bridge is implemented is as a packet filtering firewall to augment the filtering the router performs. In the event that the router should fail this device will be able to block undesirable packets. Additionally, it provides redundancy for the primary firewall n the event of a failure or if servicing is required nad it can quickly be converted to take the place of the primary firewall. One interface is configured with an IP address and connected to the management network. SSH will be used for remote management.

Primary Firewall

The primary firewall in the perimeter will be based on OpenBSD 3.0 and IPFilter 3.4.22. OpenBSD is "secure by default"…..
IPFilter has since been replaced with PacketFilter PF in OpenBSD versions 3.0 and later. PF is similar to IPF but for this implementation we believe it is prudent to allow PF to stand the test of time and get additional feedback from the open source community before implementing it as part of a secure perimeter. One interface is configured with an IP address and connected to the management network. SSH will be used for remote management.

VPN

For VPN access to the network the solution calls for IPSec tunneling using OpenSource that supports IKE, AH and ESP and that narrows the choice down to FreeS/WAN (current version is 1.95 but we will be using 1.94 for a while yet). It will run on an Intel PC with RedHat Linux 7.2 (2.4.16 kernel) hardened using Bastille Linux hardening scripts. One interface is configured with an IP address and connected to the management network. SSH will be used for remote management.

The VPN will operating in tunnel mode which means that a new IP header is created for each packet encapsulating (hiding) the real IP header as the packet traverses the untrusted networks, compromise of any given connection key will only compromise the data protected by that key and future connections remain relatively secure. Additionally, encapsulating security payload (ESP) provides encryption of the entire packet payload. The VPN will accept packets with any source IP address using IKE (UDP port 500) and ESP (IP protocol 50) and so the filtering bridge will be configured to allow these connections to be

established.

Partners with VPN will be advised to implement the same solution as GIAC Enterprises but they will be free to use any vendor they wish so long as it is compatible with FreeS/WAN and conforms to our security standards (OpenBSD, FreeBSD and Cisco are known to work well).

Remote users (road warriors) will use SSH Sentinel (v1.2) VPN client from SSH Communications Security (http://www.ssh.com). Split-horizon is never used even though it may take better advantage of the available bandwidth. The performance penalty and inconvenience to the end user is a small price to pay for the added security. The user will therefore be forced to disconnect from the VPN to use resources outside the network.

Internal Firewall

The internal firewall is based on the same configuration as the Primary Firewall. The only difference is that the rule set is more limited because it is only controlling access from one LAN segment to another. However, because of the sensitive nature of the data behind this device it will not be connected to the management segment. SSH will be used for remote management.

Other Components

1. Cisco Switches/Hubs
   Internally on the network we will use Catalyst 2900 series switches (only those directly connected to the perimeter are shown). Catalyst 2900 series switches are reasonable and sufficient for the requirements of a network of this size. Hubs will be used on the public networks and in the DMZ. The choice of hubs instead of switches is to make it easier to sniff the outside network without this being detected should a hacker manage to compromise a system on the service network. Hubs makes it nearly impossible for an intruder to determine if the network is being sniffed because a hub forwards packets out on all active interfaces and on a switch it is possible to determine that a network sniffer was present from the switch statistics that indicating a passively listening port (the mirror port).

2. DNS
   The design includes a split DNS setup to protect the trusted segments from being mapped out: Internal systems will query the internal name server whereas external hosts will query a different name server on the service network. The external system will not permit zone transfers or service request over 500k such that TCP port 53 will not be required. BIND 9.2.0 will serve as the domain name server and it will be running as *nobody* in a chroot jail. A chroot jail is a directory that BIND resides in and once it is running there it will not be able to access files outside the

jail at all. Therefore some additional requisite files need to be provisioned for (alternatively you can create statically link the binaries at build time).

```
# cd /chroot/named/lib
# cp -p /lib/libc-2.*.so .
# ln -s libc-2.*.so libc.so.6
# cp -p /lib/ld-2.*.so .
# ln -s ld-2.*.so ld-linux.so.2
```

3. Intrusion Detection

   To augment our ability to detect and intrusion we will be using Snort 1.8.3 with SnortSnarf 020126.1 for viewing alerts running on a hardened RedHat 7.2 Linux PC. The sensor will be configured with three sniffing interfaces that connect to 1) the FastHub between the transparent bridge and the Primary Firewall, 2) the FastHub on the Service Network and 3) on the mirror port of the switch immediately inside the Primary Firewall (and VPN Gateway). A fourth interface will be used to connect the Snort system to the management network (an entirely separate network segment used for managing the security devices on the perimeter).

   To avoid detection by a seasoned hacker the Snort sensor will connect be connected to the hubs and switch with a straight-through sniff-only cable will be connected made like this: Hub/Switch side: pin 1 [TxData +] is shorted onto pin 3 [RxData +] and Pin 2 [TxData -] is shorted onto Pin 6 [RxData -] this tricks the hub/switch into thinking that the port is active because a circuit is created that allows the “alive” signal from the hub/switch on the transmit pair to be returned to the hub/switch on the receive pair which give is the signal required to keep the port in the active state (Fig. 1a). On the IDS end of the cable only the Receive pair (Pins 3 and 6) is connected (Fig 1b).

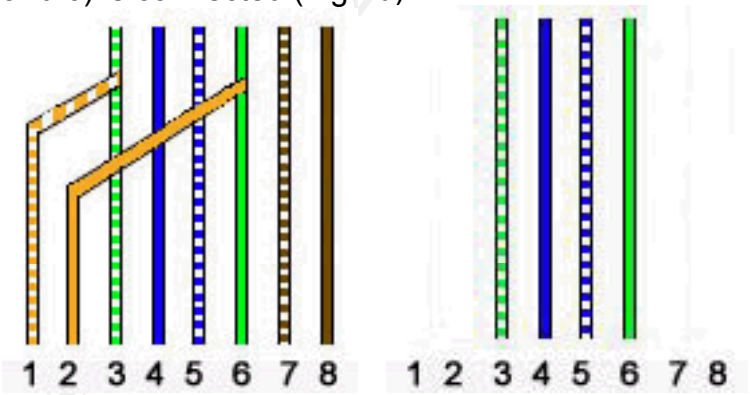


Figure 3: Diagram showing sniffing or read-only cable wiring schematic for both ends of the cable (courtesy Iron Comet Consulting http://www.ironcomet.com/ethernet.shtml). The left side depicts the wiring of the RJ-45 plug that connects to the hub or switch. The right side of the diagram depicts the wiring of the RJ-45 plug that goes into the promiscuous NIC on the IDS system.

4. NTP server will reside on the management segment and provide time-

synchronized time to the devices that log events. These include the Transparent Bridge, the Primary Firewall, the VPN Gateway and IDS system . The Syslog service will run on the same system as well.

5. Apache is the web server on the Service Network. It runs on a hardened (Bastille Linux) RedHat 7.2 Linux system. SSH is used for remote management.
6. Qmail is used as an SMTP relay/proxy where mail can be scrubbed for viruses. The MX record for the domain points to this device. Sophos MailMonitor is used to scrub the email. Mail is then forwarded to the Exchange system on the trusted network.
7. SQUID 2.4 acts as a web proxy agent for partners accessing data over the VPN connection.

Assignment 2 – Security Policy (35 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 Access Control List (ACL), firewall ruleset, IPSec policy, etc. (as appropriate) for the specific component used in your architecture. For each component, be sure to consider the access requirements for internal users, customers, suppliers, and partners that you defined in Assignment 1. The policies you define should accurately reflect those business needs as well as appropriate security considerations.

You **must** include the complete policy (explicit ACLs, ruleset, IPSec policy) in your paper. It is not enough to simply state "I would include ingress and egress filtering…" etc. The policies may be included in an Appendix if doing so will help the "flow" of the paper.

(Special note on 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.)

In addition, for **one** of the three security policies defined above, you **must** incorporate a tutorial on how to implement the policy. Use screen shots, network traffic traces, firewall log information, and/or URLs to find further information to clarify your instructions. Be certain to include the following:

A general explanation of the syntax or format of the ACL, filter, or rule for your device.

A general description of each of the parts of the ACL, filter, or rule.

An general explanation of how to apply a given ACL, filter, or rule.

For each ACL, filter, or rule in your security policy, describe:

- the service or protocol addressed by the rule, and the reason this service might be considered a vulnerability.
- Any relevant information about the behavior of the service or protocol on the network.
- If the **order** of the rules is important, include an explanation of why certain rules must come before (or after) other rules.

Select three sample rules from your policy and explain how you would test each rule to make sure it has been applied and is working properly.

Be certain to point out any tips, tricks, or potential problems ("gotchas").

Assignment 2: Security Policy

Border Router

In order to stop certain attacks and also reduce the number of packets that the primary firewall must process, Access Control Lists (ACLs) are applied to the serial interface of the router (untrusted network) for *ingress filtering* and on the Ethernet interface (DMZ) for *egress filtering*. *Standard* ACLs are used for ingress filtering because it produces the least CPU load on the router. Although there are added benefits of using *extended* ACLs and the *established* criterion could also have been used, because it is still possible for some tools such as nmap to penetrate this ACL by using the ACK and/or RST flags.

To further enhance performance, it is wise to periodically review the packet drop statistics and use those statistics as a basis to sort the most frequently used ACLs to the top. However, for brevity, the rules are ordered by IP address in this policy. As a rule, un-routable or un-assigned IP addresses are not logged because it is impossible to determine the source. However, it may be necessary to turn logging on periodically to analyze a specific problem.

Ingress Filtering

```
Router(config)#
       int serial0
       ip access-group 11 in
       no ip-source route
       exit

sh access-list 11
! Historical broadcast
access-list 11 deny 0.0.0.0 0.255.255.255
! IANA Reserved
access-list 11 deny 1.0.0.0 0.255.255.255
access-list 11 deny 2.0.0.0 0.255.255.255
access-list 11 deny 5.0.0.0 0.255.255.255
access-list 11 deny 7.0.0.0 0.255.255.255
! Private unroutable IPs RFC 1918
access-list 11 deny 10.0.0.0 0.255.255.255
access-list 11 deny 172.16.0.0 0.15.255.255
access-list 11 deny 192.168.0.0 0.0.255.255
! IANA TEST-NET IPs draft-manning-dsua – documentation network
access-list 11 deny 192.0.2.0 0.0.0.255
! Loopback
access-list 11 deny 127.0.0.0 0.255.255.255
! IANA Link Local Networks – autoconfiguration when no DHCP available
access-list 11 deny 169.254.0.0 0.0.255.255
! Sun Microsystems private cluster interconnects /23
access-list 11 deny 204.152.64.0  0.0.1.255
! IANA class D multicast IPs 224-239
access-list 11 deny 224.0.0.0 31.255.255.255
! IANA class E multicast IPs
access-list 11 deny 240.0.0.0 7.255.255.255
! Broadcast address
```

```
access-list 11 deny 255.255.255.255 0.0.0.0
! Serial interface of border router
access-list 11 deny 209.19.199.107 0.0.0.63
! GIAC IP Addresses
access-list 11 deny 209.19.248.64 0.0.0.31
! gotomypc.com
access-list 11 deny 65.251.224.169 0.255.255.255
access-list 11 permit any
```

Egress Filtering

In line with our guiding principle of being a good Internet neighbor we will use egress filtering to ensure that no illegitimate packets pass through our border router to the Internet.

```
Router(config)#
       int eth0
       ip access-group 111 in
       no ip-source route
       exit

sh access-list 111
! permit traffic from GIAC public IPs but log
access-list 111 permit 209.19.248.64 0.0.0.31 log
! block access to gotomypc.com and log
access-list 111 deny ip 65.251.224.169 0.0.0.0  any log
! deny all other packets and log
access-list 111 deny ip any any log
```

Additional Configuration

Some additional settings are prudent to secure the router.

```
Router(config)#
```

- Secure access with one way MD-5 hash on password:
  `enable secret`
- Encrypts password on the router:
  `service password-encryption`
- Restrict telnet access from firewall only:
  ```
  access-list 12 permit 209.19.248.66 0.0.0.31
  linr vty 0 4
  access-class 12 in
  ```
- Turn on buffered logging so that it can be viewed with the show log command:
  `logging buffered`
- Enable logging to syslog server:
  `logging 209.19.248.66`
- Drop all packets with source-route flag set:
  `no ip source-route`

- Disable finger service:
  ```
  no service finger
  ```
- Block Cisco Discovery Protocol (CDP):
  ```
  no cdp run
  ```
- Disable small ports below 20 for TCP and UDP:
  ```
  no service tcp-small-servers
  no service udp-small-servers
  ```
- Disable administrative http server running on router:
  ```
  no ip http server
  ```
- Disable boot protocol:
  ```
  no ip bootp server
  ```
- Banner warning:
  ```
  banner /
  WARNING:  This is a private network. Unauthorized access will be prosecuted
  to the fullest extent of the law.
  /
  ```

Interface specific settings:

```
Router(config-if)#
int eth0
```
or
```
int serial0
```

- Disable directed broadcast (RFC 2644). Default on 12.x, but to be sure it will be set on all interfaces:
  ```
  sh ip interface brief
  no ip directed-broadcast
  ```
- Disable SNMP traps:
  ```
  no snmp
  ```
- Disable CDP on interface:
  ```
  no CDP enable
  ```
- Prevent inverse mapping of the network:
  ```
  no ip unreachables
  ```
- Prevent attacker from using our router for redirects:
  ```
  no ip redirects
  ```

Transparent Filtering Bridge

implement the system as a black box in your perimeter solution by configuring the system as an Ethernet bridge: Typically, a bridge is used to make one continuous Ethernet segment from two separate Ethernet segments that may be located in different buildings, exceed the maximum wire lengths or be of different speeds. Because the bridge operates on layer 2 of the ISO stack it does not produce a hop in the route and thus it is invisible[1] to any systems transmitting across the bridge. Several operating systems (e.g. Linux and the

[1] Machines that are connected to the same segment as the device would be able to see it by doing netstat, however as long as no hacker is able to compromise the border router or the device connected to the inside interface of the transparent bridge (typically a switch, hub or other firewall) the device is “invisible.”

UNIX-like BSDs) can be used to create a bridge from a single PC with two or more network adapters. By combining the bridging functionality with a firewall that operates at layer 3 of the OSI model create a transparent filtering bridge.

There are several situations in which a transparent filtering bridge can be quite useful as a component of your perimeter defense system. For instance, if the subnet provided by your ISP is too small to subnet further the use of a filtering bridge provides an additional security device without creating subnet headaches. Additionally, there are scenarios where a firewall or perimeter is implemented after a class C network has been in use for some time and so using a filtering bridge alleviates the issues of re-issuing a non-routable class C IP address range.

We will be setting this bridge up with three identical network adapter cards two of which will be used I the bridge and one will be use for management (this latter interface will only participate on the management and monitoring network segment).

Port 51 (AH); port 50 (ESP) and UDP 500 (IKE)
As this system is almost identical to the primary firewall that is used for the tutorial, only the additional configuration and settings are covered in this part. Ipnat will not be used so portmap and nat will be turned off:

- `/etc/hostname.dc0 # you can use autosensing instead of 100BaseT`
  ```
  inet media 100BaseT
  up
  ```
- `/etc/hostname.dc1`
  ```
  inet media 100BaseT
  up
  ```
- Edit `/etc/hostname.dc2`
  ```
  inet media 100BaseT
  up
  ```
- Create `/etc/bridgename.bridge0`
  ```
  add dc0 add dc1 up
  ```

Primary Firewall – security tutorial

Implementing a Security Policy on an OpenBSD Firewall.

Preamble

This tutorial describes configuring a PC running the OpenBSD (v 3.0) operating system with IPFilter (v 3.4.22). OpenBSD/IPFilter firewall is a good choice for a firewall because OpenBSD's goal is "to be the NUMBER ONE in the industry for security" (www.openbsd.org). To that end OpenBSD.org audits all files in the OS and packages line by line to find any security holes and flaws before releasing a new distribution. In addition to the major code review, nonessential services are disabled by default when the operating system is installed. These criteria makes OpenBSD an ideal OS candidate for a security device and it is

one of the reasons that several security product vendors also use it as the underlying OS of their security products.

Installation and configuration of the underlying operating system

For the installation of the operating system we follow the standard installation procedure that is documented at http://www.openbsd.org/faq/faq4.html and therefore step-by-step instructions are not included in this tutorial. Note however, that only the "required" files plus the compiler (comp30.tgz) will be installed on the system (http://www.openbsd.org/faq/faq4.html#FilesNeeded). The compiler is necessary to create a custom kernel (see below).

To use this system as a security device it is prudent perform some additional hardening tweaks and also remove unneeded files. Hardening the system we follow the guidelines that George Schaffer has graciously provided to the OpenBSD community at http://geodsoft.com/howto/harden/bsdhardn.htm. There is also a checklist on the site that makes the entire hardening process highly repeatable. Adapt the checklist to your specific requirements.

The last configuration step is to customize the kernel for use as a firewall. Even though OpenBSD is "secure by default," the kernel is installed in a generic form. A custom kernel only loads those devices necessary for operation. There are many good sources on how to compile the custom kernel but here again we employ the guides provided by Mr. Scaffer http://geodsoft.com/howto/harden/OpenBSD/kernel.htm. The last step we perform use the "detar.sh" shell script from Wes Sonnenreich http://www.openlysecure.org/openbsd/scripts/detar-script.html to uninstall the comp30.tgz disk set. We do not want to leave the compiler on the firewall because if the system were ever to be compromised you essentially provide the attacker with a toolset. Although the attacker may be able to upload their own tools it requires more steps and additional time which may be that small edge you need to thwart a successful attack.

The last item is to install SSH (v 2.0) in order to remotely manage the system securely.

IPFilter

IP Filter (ipf) consists of a suite of files (ipf, ipnat and ipstat) that control *whether or not* the underlying OS should route packets from one interface to another. Ipf processes a rule file (ipf.rules) that contains all ACLs. Unlike NetFilter, IP Filter distinguishes between packet filtering and forwarding (NAT or masquerading) that the ipnat program instead handles. Lastly, the ipstat program maintains the state table to keep track of established communications sessions. At boot time, the programs ipf, ipnat and ipstat are loaded into memory along with the associated rule sets (ipf.rules and ipnat.rules). When a packet arrives at one of the interfaces on the firewall the first step is that ipnat.rules is checked to see if translation of the destination address is required.

Next, the state table is checked to determine if the packet is part of an established session. The last step is to compare the packet with the ACLs in ipf.rules and if necessary update the state table (if it's a new session and the ACL calls for maintaining state).

Before proceeding, a few additional utilities are worth mentioning: *Ipmon* reads log files saved from /dev/ipl (and /dev/ipnat) and is necessary to forward log files to syslog (ipmon –s). *Ipfstat* reports on packet filter statistics and filter lists and is very useful for troubleshooting purposes or to performance-tune the firewall rule set. Lastly, *ipftest* is useful for testing the effects of a rule set on test packets and should be used to test a new rule set before it is implemented.

IP Filter operates by the "last-fit" packet-to-rule-matching mode (compare with NetFilter "first-fit"). Ipnat however, follows the principle of first-fit in matching. As the name implies last-fit mode operates by comparing all packets with all rules and making a decision to pass or drop the packet according to the last rule to match. It is therefore important to setup rules such that they are ordered from general to specific otherwise you may shoot yourself in the foot. Consider this simple scenario: The first rule states to "block all packets", the second rule states "pass DNS queries." All packets match the first rule but only DNS queries match the second rule and are passed through the firewall. If the rule order were reversed nothing would get through the firewall because all packets would match the last rule and be blocked.

The pros of this type of rule matching mode is that the rule set will remain fairly easy to read even if it is complex and potentially preventing some serious "gotchas". The obvious disadvantage however, is that the firewall consumes a large amount of finite resources (CPU and memory) for each packet to be compared with each rule. On a firewall with a large rule set and a high volume of traffic the resulting overhead could prove to be a bottleneck in the perimeter defense system.

The keyword "quick" can be used to overcome the potential issue of high overhead because it short-circuits the rule set so that any remaining rules in the rule set are not compared with the packet at hand. Additionally, the keyword "head" is used to order rules into "groups." Use in conjunction, these three important keywords (quick, head and group) can turn an IP Filter firewall into an efficient stateful packet filtering system as this tutorial will hopefully convey.

Definitions of common syntax and some options used in a rule set are listed here:
block – drop the packet. Can be used with return-rst (reset), return-cmp(<type>) or return-icmp-as-dest(<type>). E.g.: return-icmp(3) = network unreachable.

- pass – pass the packet
- log – log the packet (note this action is taken immediately and is independent of whether or not the packet is passed or discarded by another rule in the rule set)
- quick – perform *the* action (e.g. pass or block) and override all remaining rules in the rule set.
- on – used to specify interface name.

- proto – option to filter by port or protocol (the most common protocol have keywords e.g. icmp, tcp, udp etc.)
- head/group (used together). A new group is created with *head* and all packets matching the head rule will be processed by rules of the specified *group*.
- state – keeps status of a communications session (TCP, UDP or ICMP). Remaining packet-filtering rules are not compared.
- frags – keeps status of fragmented packets.
- Ipopts and short – used to match against TCP header options and short fragments
- flags (FSRPAU) – Used with TCP filtering to match flags set in the TCP header.
- icmp-type – used with proto icmp. Use numbers or abbreviations of ICMP type.
- map – map one address to another.
- map-block – maps one network into another network and port range and can be used to compress a larger network into a smaller one by using IP/ports.

Note: When "keep state" is used, all packets are first compared with entries in the state table before being matched with rules in the rule set and if a state match is found the rule set is overridden.

Setting up the firewall rule set

It is necessary to change some configurations to the OS and initialize devices. I is assumed that the devices were detected and therefore it wont be necessary to create the "hostname.*" files (informational comments follow "#"). Edit the following files:

- /etc/rc.conf

```
ipfilter=yes
ipnat=yes
inetd=no # it is undesirable to run any services on this system
portmap=yes
```

- /etc/sysctl.conf

```
net.inet.ip.forwarding=1
```

Setup the NATing rules:
Create ipnat.rules file. Include the following lines in the file (they are wrapped here but would each appear on one line in ipnat.rules):

```
# redirect packets received on dc1/209.19.248.81, port 25,
to 172.16.1.smtp-inside, port 25.
rdr dc1 209.19.248.smtp-inside/32 port 25 -> 172.16.1.smtp-
inside port 25

# redirect packets received from the border router to the
firewall on port 514 (syslog) to management network syslog
server
```

```
rdr dc0 209.19.248.66/32 port 514 -> 192.168.1.syslog port
514 udp

# map all internal traffic to 209.19.248.97/28 using 252
ports to each IP address.  Each internal IP is limited to
252 simultaneous connections by this method of NAT.
map dc0 172.16.1.0/24 -> 209.19.248.97/28 portmap tcp/udp
auto
map dc0 172.16.1.0/24 -> 209.19.248.97/28
```

The remainder of this tutorial focuses on configuring the firewall rule set.

For ease of interpreting the firewall rule set we include comments and explanations throughout the rule set:

```
#######################################
# Primary Firewall Rule Set
# dc0 = external interface   209.19.248.66/27
# dc1 = service network      209.19.248.81/28
# dc2 = LAN                  172.16.1.1/24
# dc3 = management interface 192.168.1.1/24
# lo0 = loopback adapter (localhost)
# border router =            209.19.248.65
```

The default policy is to explicit deny all packets according to our guiding principles therefore we start by blocking all traffic. We have also included the "log" option which allows us to review our rules after some time of operation as well as periodically; any packets that are not matched by any other rule will be logged. Packets that show excessive logging may indicate that the rule set is misconfigured and may need some adjustment. Once you are comfortable that the rule set is performing as it should, you can review the logs to see who is knocking at your door (confer with the IDS system as well) and adjust the rule set if necessary.

```
# Default policy in place:
block in log all
```

Accept all packets coming from the internal interface. (lo0 is the loopback adapter).

```
# Loopback interface
pass in  on lo0 all
pass out on lo0 all
```

To make the rules easier to interpret and manage they are divided into groups based on the on which interface they come in to:

```
# Split filtering into groups by incoming interface:
block in  quick on dc0 all head 100
block in  quick on dc1 all head 200
block in  quick on dc2 all head 300
```

```
block in  quick on dc3 all head 400

# Split filtering into groups by outgoing interface:
block out quick on dc0 all head 1000
block out quick on dc2 all head 3000
```

All packets that enter the firewall on the DMZ side are handled by these rules:

```
#####################################################################
##
# dc0 Filtering (group 100 and 1000)
#####################################################################
##

  # Inbound (group 100)

  # Ingress filtering of spoofed source address and other malicious
packets
  block in quick from 0.0.0.0/7       to any group 100
  block in quick from 2.0.0.0/8       to any group 100
  block in quick from 5.0.0.0/8       to any group 100
  block in quick from 10.0.0.0/8      to any group 100
  block in quick from 23.0.0.0/8      to any group 100
  block in quick from 27.0.0.0/8      to any group 100
  block in quick from 31.0.0.0/8      to any group 100
  block in quick from 69.0.0.0/8      to any group 100
  block in quick from 70.0.0.0/7      to any group 100
  block in quick from 72.0.0.0/5      to any group 100
  block in quick from 82.0.0.0/7      to any group 100
  block in quick from 84.0.0.0/6      to any group 100
  block in quick from 88.0.0.0/5      to any group 100
  block in quick from 96.0.0.0/3      to any group 100
  block in quick from 127.0.0.0/8     to any group 100
  block in quick from 128.0.0.0/16    to any group 100
  block in quick from 128.66.0.0/16   to any group 100
  block in quick from 169.254.0.0/16  to any group 100
  block in quick from 172.16.0.0/12   to any group 100
  block in quick from 191.255.0.0/16  to any group 100
  block in quick from 192.0.0.0/19    to any group 100
  block in quick from 192.0.48.0/20   to any group 100
  block in quick from 192.0.64.0/18   to any group 100
  block in quick from 192.0.128.0/17  to any group 100
  block in quick from 192.168.0.0/16  to any group 100
  block in quick from 197.0.0.0/8     to any group 100
  block in quick from 201.0.0.0/8     to any group 100
  block in quick from 204.152.64.0/23 to any group 100
  block in quick from 219.0.0.0/8     to any group 100
  block in quick from 220.0.0.0/6     to any group 100
  block in quick from 224.0.0.0/3     to any group 100

  # Block our own address space from the outside
  block in quick from 209.19.248.64/27 to any group 100

  # Block IP fragments
  block in quick all with frag group 100

  # Block short IP fragments
  block in quick proto tcp all with short group 100
```

```
  # Block all packets with IP options using "ipopts" option
  block in quick all with ipopts group 100

  # Block OS fingerprinting by scanners (e.g. nmap)
  Block in quick proto tcp all flags FUP group 100

  # Block all packets to network and broadcast address to prevent a
Smurf attack
  block in log quick from any to 209.19.248.64/27 group 100
  block in log quick from any to 209.19.248.95/27 group 100

  ##################################################################
  # Filtering subdivided by protocol (note the interface is implied):

  # TCP traffic
  block in quick proto tcp  all head 110 group 100
      pass in from any to 209.19.248.84/28 port = 25    group 110
      pass in from any to 209.19.248.83/28  port = 80    group 110

  # UDP traffic
  block in quick proto udp all head 130 group 100
      pass in from any to 209.19.248.82/28 port = 53 keep state
group 130
      pass in from 209.19.248.65 to 209.19.248.66/28 port = 514 keep
state    group 130

  # ICMP traffic
  block in quick proto icmp all head 120 group 100
      # Allow only type 0, 3, and 11 ICMP
      pass in from any to 209.19.248.80/28 icmp-type 0  keep state
group 120
      pass in from any to 209.19.248.80/28 icmp-type 3  keep state
group 120
      pass in from any to 209.19.248.80/28 icmp-type 8  keep state
group 120
      pass in from any to 209.19.248.80/28 icmp-type 11 keep state
group 120


  # Outbound (group 1000)

  # Egress filtering here
  block out quick from 0.0.0.0/7       to any group 1000
  block out quick from 2.0.0.0/8       to any group 1000
  block out quick from 5.0.0.0/8       to any group 1000
  block out quick from 10.0.0.0/8      to any group 1000
  block out quick from 23.0.0.0/8      to any group 1000
  block out quick from 27.0.0.0/8      to any group 1000
  block out quick from 31.0.0.0/8      to any group 1000
  block out quick from 69.0.0.0/8      to any group 1000
  block out quick from 70.0.0.0/7      to any group 1000
  block out quick from 72.0.0.0/5      to any group 1000
  block out quick from 82.0.0.0/7      to any group 1000
  block out quick from 84.0.0.0/6      to any group 1000
  block out quick from 88.0.0.0/5      to any group 1000
  block out quick from 96.0.0.0/3      to any group 1000
  block out quick from 127.0.0.0/8     to any group 1000
```

```
  block out quick from 128.0.0.0/16    to any group 1000
  block out quick from 128.66.0.0/16   to any group 1000
  block out quick from 169.254.0.0/16  to any group 1000
  block out quick from 172.16.0.0/12   to any group 1000
  block out quick from 191.255.0.0/16  to any group 1000
  block out quick from 192.0.0.0/19    to any group 1000
  block out quick from 192.0.48.0/20   to any group 1000
  block out quick from 192.0.64.0/18   to any group 1000
  block out quick from 192.0.128.0/17  to any group 1000
  block out quick from 192.168.0.0/16  to any group 1000
  block out quick from 197.0.0.0/8     to any group 1000
  block out quick from 201.0.0.0/8     to any group 1000
  block out quick from 204.152.64.0/23 to any group 1000
  block out quick from 219.0.0.0/8     to any group 1000
  block out quick from 220.0.0.0/6     to any group 1000
  block out quick from 224.0.0.0/3     to any group 1000

  # Block IP fragments (state table should keep track of fragments on
stateful traffic)
  block out quick all with frag group 1000

  # Block short IP fragments
  block out quick proto tcp all with short group 1000

  # Block all packets with IP options using "ipopts" option
  block out quick all with ipopts group 1000

  # Block OS fingerprinting by scanners (e.g. nmap)
  Block out quick proto tcp all flags FUP group 1000

######################################################################
##
# dc1 (groups 200 and 2000)
######################################################################
##

  # Inbound  (group 200)

  pass in quick from 209.19.248.84/28 to any proto tcp port = 25 keep
state group 200
  pass in quick from 209.19.248.84/28 to 209.19.248.81/28 proto tcp
port = 25 keep state group 200
  block in from any to any log

######################################################################
##
# dc2 (groups 300 and 3000)
######################################################################
##

  # Inbound  (group 300)


  pass in quick proto tcp from 172.16.1.0/24 to any port = 80 flags S
keep state keep frags group 300
  block in  log quick all group 300

  # ICMP traffic
```

```
  block in quick proto icmp all head 320 group 300
      # Allow only type 0, 3, and 11 ICMP
      pass in from any to 209.19.248.80/28 icmp-type 0  keep state
group 320
      pass in from any to 209.19.248.80/28 icmp-type 3  keep state
group 320
      pass in from any to 209.19.248.80/28 icmp-type 8  keep state
group 320
      pass in from any to 209.19.248.80/28 icmp-type 11 keep state
group 320

  # Outbound (group 3000)

  pass out quick on dc2 proto tcp from any keep state flags S keep
frags  group 3000
  block out log quick all                                    group
3000

  # Nothing from the managment network is allowed through to the
trusted network
  block out log quick any from dc3 group 3000

##############################################################################
##
# dc3 (groups 400)
##############################################################################
##

  # Inbound  (group 400)

  # remote management via SSH
  pass in from 192.168.1.2/24 to 192.168.1.1/24   proto tcp port = 22
group 400
  pass in from 192.168.1.2/24 to 209.19.248.65/27 proto tcp port = 22
group 400
  pass in from 192.168.1.2/24 to 209.19.248.84/28 proto tcp port = 22
group 400
  pass in from 192.168.1.2/24 to 209.19.248.83/28 proto tcp port = 22
group 400
  pass in from 192.168.1.2/24 to 209.19.248.82/28 proto tcp port = 22
group 400

  # ICMP traffic
  block in quick proto icmp all head 420 group 400
      # Allow only type 0, 3, and 11 ICMP
      pass in from any to any icmp-type = 0  keep state group 420
      pass in from any to any icmp-type = 3  keep state group 420
      pass in from any to any icmp-type = 8  keep state group 420
      pass in from any to any icmp-type = 11 keep state group 420
```

Drop all packets originating from illegal IP addresses (note: it is very important to review this list frequently when it includes unassigned IP addresses as these may be assigned at a later point). We are using the “quick” option which forces IPFilter to take action at that rule and stop processing the rule base further. We have also chosen to *not* log any of these unroutable IP addresses because we cannot determine the origin of such packets however, at times (check with the blocked packet statistics) it may be prudent to “turn on logging” and review the

activity of such packets unless you are doing that with an IDS instead.

- Unroutable IP addresses

```
# Deny packets originating outside and with illegal IPs
# Unroutable IP addresses
block in quick on dc0 from 0.0.0.0/32 to any

block in quick on dc0 from 10.0.0.0/8 to any
block in quick on dc0 from 172.16.0.0/12 to any
block in quick on dc0 from 192.168.0.0/16 to any
block in quick on dc0 from 192.0.2.0/32 to any
block in quick on dc0 from 224.0.0.0/3 to any
block in quick on dc0 from 127.0.0.0/8 to any
block in quick on dc0 from 169.254.0.0/16 to any
block in quick on dc0 from 192.168.0.0/16 to any
block in quick on dc0 from 255.255.255.255/32 to any
```

- Internal public IP addresses (e.g. firewall, DMZ, service networks) originating from the outside has to be spoofed traffic so these will be discarded.

```
# bbb
block in …
```

Allow only certain types of ICMP packets. Type 0 (echo reply) and 11 (time exceeded) are useful for traceroute and ping to work. ICMP type 3 is port unreachable.

```
   # Allow only type 0, 3, and 11 ICMP
pass in quick on dc0 proto icmp all icmp-type 0
   pass in quick on dc0 proto icmp all icmp-type 3
   pass in quick on dc0 proto icmp all icmp-type 11
```

Block fragments

- Regular fragments are blocked with the “frag” option

```
# Block IP fragments
block in quick all with frag
```

- Short fragments which are most likely malicious

```
# Block short IP fragments
block in quick proto tcp all with short
```

Block all packets with IP options set

```
# Block all packets with IP options using “ipopts” option
block in quick all with ipopts
```

Block nmap scanning using Fin, Urg and Push typically used for OS fingerprinting.

```
# Block OS fingerprinting by scanners (e.g. nmap)
Block in quick proto tcp all flags FUP
```

To prevent the Smurf attack incoming packets to the network and broadcast address have to be stopped. Directed broadcasts are off by default but you may want to check this just to be on the safe side.

```
# Block all packets to network and broadcast address to prevent
# a Smurf attack
block in log quick on dc0 to 209.19.248.64/27
block in log quick on dc0 to 209.19.248.95/27
```

To prevent the network from being spoofed with our own IP addresses they will be blocked as well.

```
# Block our own public IP addresses inside of this firewall
block in quick on dc0 from 209.19.248.66/27
block in quick on dc0 from 209.19.248.67/27
block in quick on dc0 from 209.19.248.68/27
```

Note: IP Filter has in addition to the "active" rule set, ipf can maintain an inactive rule set that is useful for debugging problems, serving as a default standby and as a way to test new rule sets before they are implemented. We will always make changes to the inactive rule set and test these before implementing new rules.

VPN Gateway

Secure access to GIAC Enterprises network from the partner network will be configured as follows. VPNs with partners will use RSA keys for authentication and will operate in tunnel mode thereby completely encapsulating the original IP header and payload. Assuming the necessary legalese paperwork is in order the ipsec utility is run with the rsasigkey option to generate a public-private key pair. Copy and paste the private key into the ipsec.secrets file (it has been generated in the correct format):

```
209.19.248.67:  rsa {
        # 2048 bits, Wed Feb 27 13:23:48 2002
        # for signatures only, UNSAFE FOR ENCRYPTION
        # pubkey=0x0e05j48w35w985uf45...
       Modulus: 0xc86c20cf1a86f11abb82d1f10...
       PublicExponent: 0x03
       # everything after this point is secret
         PrivateExponent: 0x881c59fdf38ab105c8c77d23...
        Prime1: 0x d43cb2b53321d50e34ca921e0...
        Prime2: 0xd5a9108453321d43cb2b...
        Exponent1: 0x8d914e70b5c59fdf8a4a38d9...
```

```
 Exponent2: 0x8e70b5ad8d9142168d7dcc7...
Coefficient: 0xa8453321d10c13e98d98...
```

Next copy and paste the public key into the ipsec.conf file and assign it to the leftrsasigkey. From the partner get the their public rsasigkey and assign it to the rightrsasigkey. The /etc/ipsec.conf file contains the configuration settings of the VPN Gateway. The first section distinguished by *config setup* consists of settings for the Gateway itself and any global settings for the listed tunnels:

```
# VPN Gateway configuration
config setup
      # %defaultroute uses machine gateway
      interfaces=%defaultroute
      # Debugging is off
      klipsdebug=none
      plutodebug=none
      # Use "auto=" in tunnels (see below) to control startup
      plutoload=%search
      plutostart=%search

# Global configuration for all tunnels
conn %default
      # Retry rate for key negotiations (0 = continuous).
      keyingtries=0
      # Autokeying with IKE
      keyexchange=ike
      keylife=1h
      authby=rsasig
```

For each GIAC partner with a VPN connection there will be a section denoted by `conn <tunnel_name>` as the one shown here:

```
# GIAC partner tunnel mode, RSA keys
conn GIACpartner1-VPN
      type=tunnel
      # Left security Gateway (GIAC), trusted subnet
      # Next hop (e.g. border router)
      left=209.19.248.67
      leftnexthop=209.19.248.65
      leftsubnet=172.16.1.0/24
      leftrsasigkey=0x0e05j48w35w985uf45...
      leftid=209.19.248.67
      # Right security Gateway (GIAC), trusted subnet
      # Next hop (e.g. border router)
      right=<Partner1_VPN_untrusted_interface>
      rightnexthop=<Partner1_borderrouter>
      rightsubnet=<Partner1_subnet>
      rightrsasigkey=0x0f985cf34dj45334ca5o...
      righted=<partner1_VPN_IP>
      # connection is not started at startup but standby
      authby=rsasig
      auto=start
```

FreeS/WAN uses a notion of left and right sides to distinguish between the two ends of a VPN tunnel. The choice is arbitrary and in this scenario it is the GIAC VPN Gateway that is the left side (for the partner it is they that are the leftside). Each tunnel has an IP address, next hop and trusted (private) subnet for the left side which is the configured the same way for each tunnel as it is the VPN Gateway itself. The right side is the partner network and this information will have to be provided by the partner for each GIACpartner-VPN tunnel configured in the ipsec.conf file.

IKE is the default and only method of key exchange supported by FreeS/WAN. We set the key life to 1 hour such that every hour the keys will be renegotiated with perfect forward secrecy enabled (it may be necessary to adjust this setting somewhat to accommodate lengthy data transfer sessions).

Access to the network by GIAC Enterprises road warriors is very similar although it is not possible to know what IP address they will be connecting from and so the setup is slightly different to accommodate the unknown IP addresses:

```
conn GIACremote_1
type=tunnel
   # Left security Gateway (GIAC), subnet behind it
   # Next hop toward it
   left=209.19.248.67
   leftnexthop=209.19.248.65
   leftsubnet=172.16.1.0/24
   right=0.0.0.0
   rightnexthop=<Partner1_borderrouter>
   rightsubnet=<Partner1_subnet>
   keyexchange=ike
   keylife=1h
   authby=rsasig
   leftrsasigkey=<pasted here>
   rightrsasigkey=0x0e05j48w35w985uf45...
   auto=add
```

Although it is not covered in detail here, the underlying NetFilter (IPTables) firewall accepts packets with any source IP address, UDP protocol and destination port 500 (IKE) and/or ESP (IP protocol 50). However, only connections from systems that can authenticate (private-public key pair) will be able to establish sessions. On the trusted network side of the VPN Gateway the decrypted packets from partner networks may only access the SQUID proxy to access the data on the fortune sayings database. Each partner has a fixed number of users with read access to the data on the system. GIAC road warriors can only connect to Exchange for email using Outlook in the corporate/workgroup mode and the intranet portal that provides access to the resources the road warriors require. No other access to the network is allowed from systems connecting over the VPN and this limits any impact on GIAC

Enterprises network and data that a compromise of a VPN tunnel or partner site can have.

Assignment 3 – Audit Your Security Architecture (25 points)

You have been asked to conduct a technical audit of the primary firewall (described in Assignments 1 and 2) for GIAC Enterprises. In order to conduct the audit, you will need to:

1. Plan the audit. Describe the technical approach you recommend to assess the firewall. 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. Conduct the audit. Using the approach you described, validate that the primary firewall is actually implementing GIAC Enterprises' 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. Evaluate the audit. 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 you must annotate/explain the output.

Assignment 3: Auditing the Security Architecture

Preamble

The final part of implementing a security policy and architecture is to audit the solution in its entirety ensuring that it enforces the security policy; packets that shold get dropped are dropped, packets that whould get through do get rhough; and that information about a packet gets logged when it issupoosed to.and that it doesn't inadvertently deny business critical services. If all or parts of the security policy have been incorrectly implemented the audit should uncover this. This section will focus on the primary firewall.

Planning

The security policy stipulates what type of packets should traverse the primary firewall, what information should be returned based on a request and when packet information and statistics should be logged. To test that policy really does this we need to emulate the type of traffic that the security policy is designed to handle.

The firewall audit will be performed in two phases. The first phase will concentrate on testing the firewall rule sets by attempting connections to public and non-public services. It is important to note however that this phase of the audit shall not be intrusive. With permission from the Director of MIS the first phase will be completed during normal operating hours. The second phase will test how well the firewall stands up to an active attack where it is flooded with packets. The second phase will be conducted outside of normal business hours Saturday) because it is expected to be disruptive potentially breaking something on the firewall, and anyone attempting to make connections to or from the Internet will most likely have problems doing so.

It is expected that this effort will take require two people and approximately one day and one Saturday. At a rate of $1500/day and $2300/Saturday in consulting fees, the cost of this effort is approximately $7600.

Tools

The audit will consist of two laptops running RedHat 7.2 Linux, one on either side of the firewall. The first laptop (Lin1) attached to the hub in the DMZ, will serve as a user or attacker emulating packets coming from the Internet. The other laptop (Lin2) will be attached to hub immediately inside the firewall on the service network and capture packets that pass through. The roles will be reversed to test functionality in both directions. Next, the second laptop will be connected to the mirror port on the switch immediately inside the firewall on the LAN and the procedure will be repeated in both directions. Finally, the second laptop will be connected to the hub on management network and the procedure will be repeated again (in both directions).

- Ping to test connectivity.
- Nmap (by Fyodor) will be used to scan the firewall, perform OS fingerprinting of the firewall itself and of devices on the service network, and to create packets of various kinds that will test that packets are blocked when they should be.
- Telnet will be used to initiate connections to the mail system and SSH.
- Nslookup to test DNS.
- Ethereal will be used to capture and review the packets.

Phase I

From DMZ:

1. The first test we do is to ping all the public services from Lin1 that GIAC offers and that should respond: http, https, DNS and SMTP. Initially Lin2 running ethereal shows the echo request coming in, but Lin1 only gets "Destination host unreachable." A quick review of the firewall rule set reveals that ICMP type 0 was not permitted to pass through the firewall. As soon as the rule set has been amended all services reply indicating that they are up and that ICMP echo reply is properly passing through the firewall.
2. Using telnet a connection is established with mail.GAICEnt.com on port 25. The connection is established without problems and it is possible to send a test message to hostmaster@GIACEnt.com using "mail from:" and "rcpt to:"
3. To verify that the router is successful in updating syslog we check log files on the syslog system to see that the routers log files have been uploaded. They were not uploaded. Some quick troubleshooting narrowed down the problem: it was not the firewall that prevented syslog updates from the router but rather the syslog server had not been configured to accept syslog from other systems. To enable the syslog server to receive log files from the router some edits had to made to: /etc/rc.d/init.d/syslog. After that was completed the logs were updated as expected. Thus we know that the firewall allows syslog updates from the router and that NAT functions properly (DMZ interface of firewall port 514 is NATed to syslog server).
4. Nslookup is used to test that DNS is functioning properly. By setting the "server" to ns.GIACEnt.com the IP addresses for www.giacent.com, ns.giacent.com and mail.giactent.com were successfully resolved.
5. Using Nmap to scan the public server IP addresses to verify that *only* http, https, DNS and SMTP respond as listening:

```
[Lin1]# nmap -v -oN ServiceAudit.log 209.19.248.84
Starting nmap V. 2.54BETA22 ( www.insecure.org./nmap/ )
No tcp, udp, or ICMP ....  nmap assumes vanilla tcp connect
scan...
Host (209.19.248.84) appears to be up ... good.
Initiating Connect() Scan against (209.19.248.84):
```

```
Adding TCP port 80 (state open).
Adding TCP port 443 (state open).
The Connect() Scan took 0 seconds to scan 1542 ports.
Interesting ports on  (209.19.248.84):
(The 1540 ports scanned but not shown below are in
state:filtered)
Port        State       Service
80/tcp      open        http
443/tcp     open        https

Nmap run complete -- 1 IP address (1 host up) scanned in 21
seconds

[Lin1]# nmap -v -sU -oN ServiceAudit.log 209.19.248.83
Starting nmap V. 2.54BETA22 ( www.insecure.org./nmap/ )
Host (209.19.248.84) appears to be up ... good.
Initiating UDP Scan against (209.19.248.83):
The UDP Scan took 5 seconds to scan 1453 ports.
Interesting ports on  (209.19.248.83):
(The 1452 ports scanned but not shown below are in
state:filtered)
Port        State       Service
53/udp      open        dns

Nmap run complete -- 1 IP address (1 host up) scanned in 25
seconds

[Lin1]# nmap -v -oN ServiceAudit.log 209.19.248.82
Starting nmap V. 2.54BETA22 ( www.insecure.org./nmap/ )
No tcp, udp, or ICMP .... nmap assumes vanilla tcp connect
scan...
Host (209.19.248.84) appears to be up ... good.
Initiating Connect() Scan against (209.19.248.82):
Adding TCP port 25 (state open).
The Connect() Scan took 0 seconds to scan 1542 ports.
Interesting ports on  (209.19.248.82):
(The 1541 ports scanned but not shown below are in
state:filtered)
Port        State       Service
25/tcp      open        http

Nmap run complete -- 1 IP address (1 host up) scanned in 23
seconds
```

Only those services that are supposed to be listening are indeed listening.

From Service Network

1. To test that the SMTP server on the service network could connect outbound for mail delivery from GIAC Enterprises, a telnet session was established from SMTP server to Lin1 on port 25. The test was sufficient to verify that the firewall rule set was allowing outbound SMTP although no mail was really sent.
2. Inbound SMTP from the SMTP relay server to the internal SMTP server (firewall interface dc1 listening on port 25 is NATed to the internal mail

server) was tested by using telnet on the mail relay server to connect to the internal SMTP server on port 25. The test was successful and again revealed that NAT also was working as it was supposed to.
3. To ensure that the firewall rule set only allowed the SMTP relay to establish connections with the internal mail server a connection attempt was made from Lin2 using telnet. This failed.

From LAN

1. Outbound access on ports 80 and 443 were tested using Mozilla to connect to www.netscape.com and access Netscape mail (uses https). Both tests were successful.

From Management Network

1. SSH to the firewall, DNS, Web and mail relay was tested by opening a connections to each of these systems on port 22 using telnet. All connection attempts were successful (output for firewall shown):

```
telnet 192.168.1.1 22
Trying 192.168.1.1…
Connected to 192.168.1.1.
Escape character is '^]'.
SSH-1.99-OpenSSH_2.9p2
```

Evaluation and Recommendations

1. It was necessary to make some minor changes to the firewall rule set in order for ICMP type 0 messages to be returned to the sender.
2. A review of the syslog server also revealed that it was not receiving updates from the router which at first suggested that the firewall rule set (or NAT) was improperly configured, however it turned out to be that the syslog server had not been configured to receive remote updates and once that was corrected the router log files showed up on the syslog server.
2. When connecting to both the mail relay and the website it was determined that both systems provided too much information about the services that were offered. A few quick edits of configuration files shored up this hole.

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. Research and design **two of the following three types of attacks** against the architecture:

An attack against the firewall itself. Research and describe at least **three** vulnerabilities that have been found for the type of firewall chosen for the design. Choose **one** of the vulnerabilities, design an attack based on the vulnerability, and explain the results of running that attack against the firewall.

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.

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.

Your attack information should be detailed - include the specifics of how the attack would be carried out. Do not simply say "I would exploit the vulnerability described in Vendor Security Bulletin XXX". What commands would you use to carry out the attack? Are exploit tools or scripts available on the Internet? What additional steps would you need to take prior to conducting the attack (reconnaissance, determining internal network layout, determining valid account name.)? Would any of your methods be noticed (log files, IDS.)? What "stealth" techniques could you employ to avoid detection? What countermeasures would help prevent your attack from succeeding?

If it is possible to carry out the attack on a test system, include screen shots, log files, etc. as appropriate to illustrate your methods.

In designing your attacks, keep the following in mind:

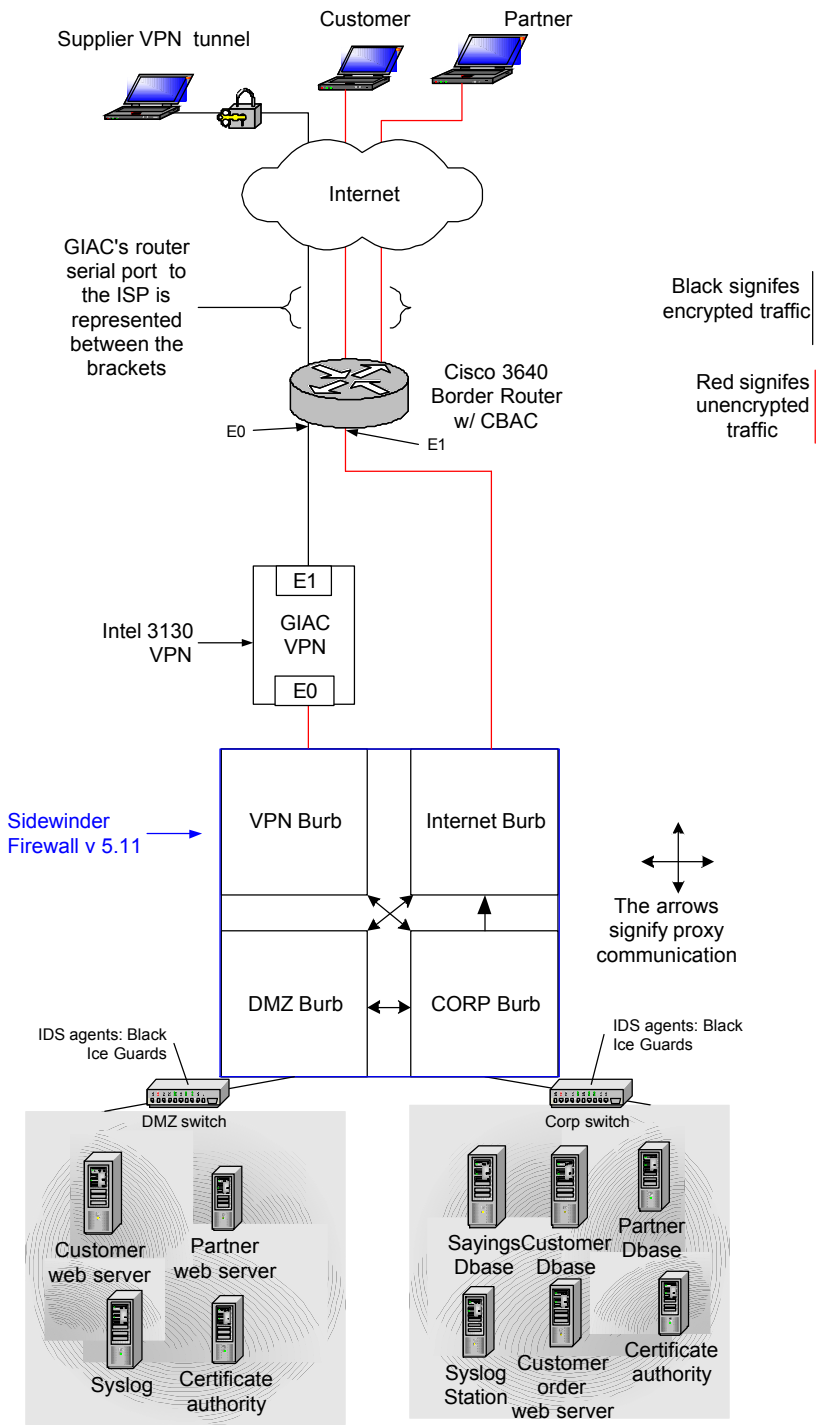
- The attack should be **realistic.** The purpose of this exercise is for the student to clearly demonstrate that they understand that firewall and perimeter systems are not magic "silver bullets" immune to all attacks.
- The attack should be **reasonable.** The firewall does not necessarily have to be impenetrable (perfectly configured with all of the up-to-the-minute patches installed). However, you should not assume that it is an unpatched, out-of-the-box firewall installed on an unpatched out-of-the-box OS. (Remember, you designed GIAC Enterprises' firewall; would you install a system like that?)
- You **must** supply documentation (e.g., a URL to the security bulletin, bugtraq archive, or exploit code used) for any vulnerability you use in your attack.
- The attack does not necessarily have to succeed (though a successful attack is often the more interesting approach). If, given the perimeter and

νετωork configuration you have described above, the attack would fail, you can describe this result as well.

Assignment 4: Design Under Fire

For this section of the practical I have chosen to attack Henry Guzman's practical found at http://www.giac.org/GCFW.php/Henry_Guzman_GCFW.zip with a denial of service attack and a compromise of a system through the firewall.

GIAC Enterprises Security Archetecture Overview



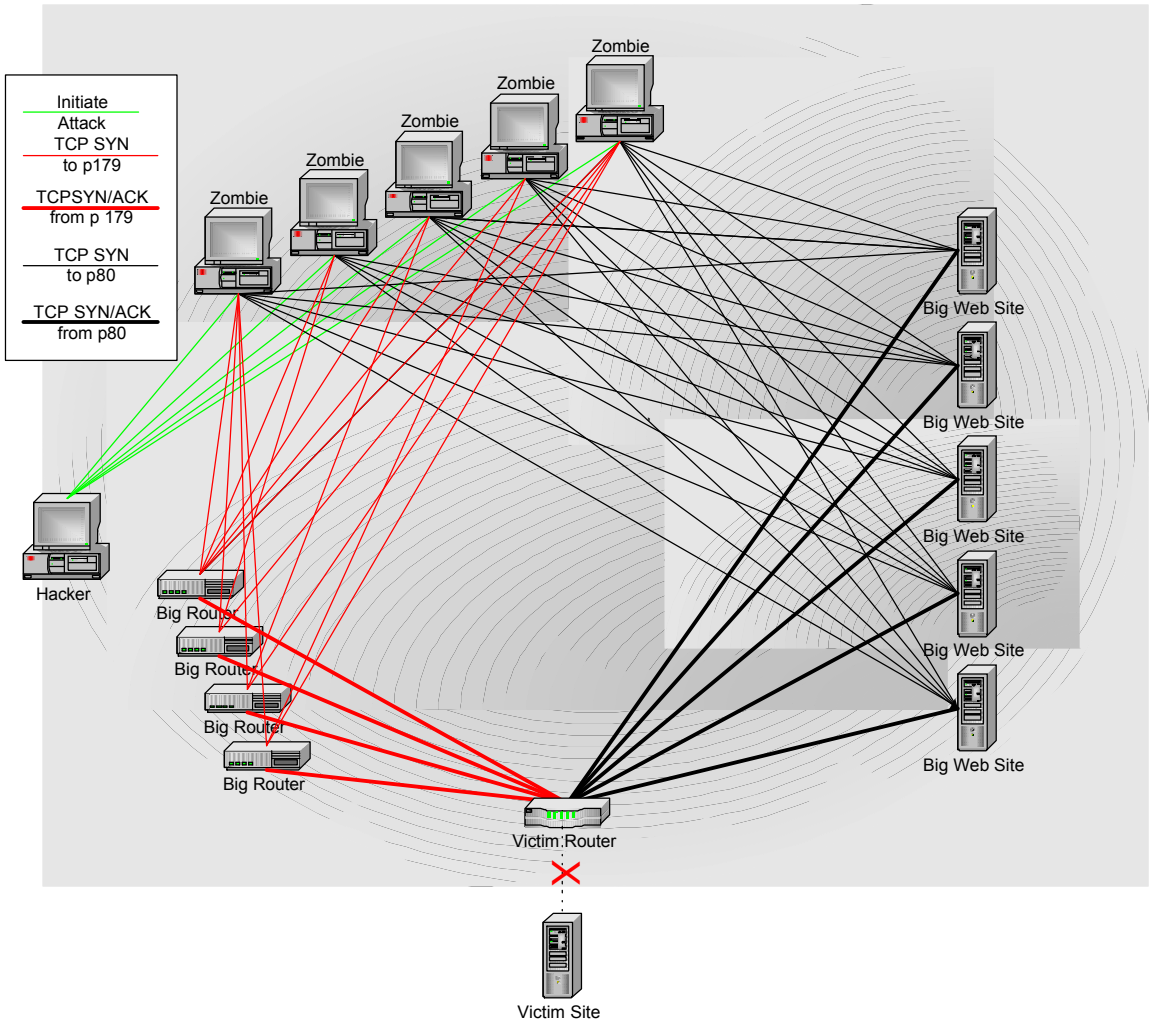
Denial of Service Attack

Background

Using the 50 Linux boxes under my control I will attack Henry Guzman's network with a new form denial of service attack that has no adequate defense. What was that? No defense? That's right, there is no real defense against this type of an attack. It is the newest variation of the Distributed Denial of Service attack (DDoS) attack called Distributed Reflected Denial of Service (DRDoS) coined by Steve Gibson (www.grc.com) who to my knowledge first described it (his paper on the subject is worth reading http://grc.com/dos/drdos.htm).

The attack is quite simple and consists of a basic TCP SYN scan with a spoofed source IP address (that of the victim) that is directed at an open port on a public internet server such as port 80 on web sites or port 179 (BGP) on internet backbone routers. The Internet servers interpret this as the first stage in the three-way TCP handshake and respond with a SYN/ACK packet to the spoofed IP address. The victim however, is not expecting the SYN/ACK it receives from the Internet server (or router) and it attempts to respond with a RST to tear down the connection. However, using a large, distributed pool of attacking systems each of which is simultaneously scanning hundreds of public Internet servers that have open ports waiting for connections, the attack becomes a highly effective distributed reflected denial of service attack: The victim network is blasted off the internet by a SYN/ACK flood from a great throng of unsuspecting, legitimate Internet servers. Moreover, as the border routers and ISP aggregate routers are flooded with packets and begin to drop these, the unsuspecting reflecting servers resend the SYN/ACK. They resend the SYN/ACK because the final ACK in the three-way handshake they were expecting never arrives and to top it off they may try to resend the SYN/ACK up to four times thereby essentially amplifying the attack almost fourfold. In fact, 50 compromised systems on broadband networks used in an attack is more than ample power to bring down a 100-megabit fat Internet pipe.

To the reflection servers which are naturally used to handling a high volume of traffic, the attack will most likely go completely unnoticed because as long as the attack is sprayed over a large group of Internet servers and also spread in time, each individual system will really only see a trickle of what appears to be legitimate traffic from the victim ("source"). Reviews of the firewall and IDS logs on the reflection server networks will only reveal subtle clues to the most attentive security administrator but even that can be thwarted by a very well coordinated attack; if the list of reflection servers is randomized and the attack is spread temporally, it can be almost impossible to detect this attack.



- *Attack Preparation*
-
- The first task is to assemble a list of the thousand or so most commonly visited web sites on the Internet (http://www.referencedesk.org/topsites.html, http://www.placedirectory.com/toplinks.htm, and http://www.web100.com/listings/all.html). Next, using some simple perl scripts I recompile the list of the thousand web sites with their IP addresses (many of the web sites have multiple IP addresses that helps increases the list of reflection servers). Using the web site IP address file and another simple script that runs traceroute on all the IP addresses in the list, I compile a second list of the large backbone routers on the Internet. I load these two IP address files onto my zombie hosts (buried along with my t00ls somewhere where the unsuspecting owner is unlikely to find them).
-
- *The Attack*
-
    - Invoking my 50 strong zombie army of broadband hosts over my secret IRC channel I order them to run nmap TCP SYN scan in stealth mode, using the list of IP addresses as the target hosts and the target system's

spoofed IP address:

```
Nmap -sS -iL <router_ip_list> -p 179 -S <victim_IP> -e eth0 -P0
Reading target specification from FILE: <router_ip_list>

Starting nmap v. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on (first_ip_in_list)
Port       State      Service
179.tcp    filtered   open

Interesting ports on (second_ip_in_list)
Port       State      Service
179.tcp    filtered   open

.
.
.

Interesting ports on (last_ip_in_list)
Port       State      Service
179.tcp    filtered   open


Nmap run complete -- 1000 IP addresses (1000 hosts up) scanned
in 3567 seconds
```

- After about an hour I repeat the procedure but this time I use a slightly different attack:

```
Nmap -sS -iL <website_ip_list> -p 80 -S <victim_IP> -e eth0 -P0
```

- Ideally the second attack is initiated some time before any action can be taken against the attack and depending on the quality of the relationship between the GIAC Enterprises' IT/Security group and their ISP as well as their general preparedness for such an attack, enabling such a filter may take unto several hours. So when a filter to drop all packets with a source port of 179 is implemented on the aggregate routers to stop SYN/ACKs originating from a host of internet routers and the IT/Security team believe they have the problem solved, the network immediately comes under attack from the most frequently visited web sites on the World Wide Web (Microsoft, AOL, Google etc.). This second attack has essentially been in progress all along but it has been out competed by the attack form the routers that sit right on the backbone of the Internet. At this point there is little they can do but wait for the attack to end. And to make it worse, I can order my 'bots to use many of the other common ports such as ftp, SMTP, Telnet, POP, SSH that many Internet servers are listening on just to make it interesting.

If I want to jerk those firewall and network administrators around real good, I could alter my attack so that rather than a flood of the network causing a DoS I spread the attack over even more Internet Servers and I limit the scan to a few blips here and there. Instead of being blasted off the network, GIAC

Enterprises experiences a period where the Internet seems to be really slow – a degradation of service attack. The subtlety of the attack may not alert the firewall administrators to go and massage the logs and see that there is a substantial inflow of "unsolicited" SYN/ACKs from legitimate web sites. Most likely plenty of time would be wasted calling the ISP to see if there are network outages, looking for statistics that indicate a high degree of congestion on the Internet and generally wasting the administrators time. Needless to say, the helpdesk would be flooded with complaints from users that had problems accessing websites.

Mitigation

There are few ways to stop this type of attack. Any attempt to filter packets from one port or another low- or high-numbered, would inevitably deny legitimate clients on your trusted network access to the services on the Internet.

The only way to stop this type of an attack is to implement egress filtering on the ISP's networks. This type of attack will not be possible if egress filters that pass only packets with legitimate source IP addresses (those that belong to the ISP), to leave the ISP's network are implemented on the ISP's routers. Any other attack that depends on spoofed IP source address would also fail. Furthermore, should someone attempt a DoS attack it would be possible to trace the packets to their source or at least to the ISP.

So why do the ISPs not do egress filtering? Primarily it is a matter of cost to manage the many routers and an unwillingness to change it seems. However, it is highly likely that in the coming years we will begin to see that ISPs implement egress filtering on their networks because as the cost of attacks burdens corporate America eventually ISPs will be held liable for the problem in court.

Compromise of an internal system through the firewall.

Preparation for Attack

❑

This attack will be based primarily on the fine art of social engineering, a bit of URL trickery and good, old, insecure ActiveX. The first phase in the attack is to do some reconnaissance that will aid me in zeroing in on suitable targets. I am looking for non-technical people with large egos so that they are likely to fall for my attack (read VPs and CEOs).

I plug the website URL into www.samspade.org to get as much info about the company as I can. The type of information I am looking for is the technical, administrative and billing contacts for the site so that I can accurately guess how the company uses email addresses, but I also get information about the MX record, IP range, DNS servers that I could use for other attacks if I so desired.

```
Registrant:
GIAC Enterprises, Inc. (GIAC-DOM)
   2601 N. Fortune Street
   Fortunately,FL 33604
   US

   Domain Name: GIACENTERPRISES.COM

   Administrative Contact:
      Thompson, Roy  (RT34352)  rthompson@giacent.com
      GIAC Enterprise, Inc.
      2601 N. Fortune Street
      Fortunately,FL 33604
      555-838-8755 (FAX) 555-836-2522

   Technical Contact:
      Hammer, Tina  (TH36752)  thammer@giacent.com
      GIAC Enterprise, Inc.
      2601 N. Fortune Street
      Fortunately,FL 33604
      555-838-8755 (FAX) 555-836-2542
   Billing Contact:
      Brett, Marge  (MIB5634)  mibrett@giacent.com
      GIAC Enterprise, Inc.
      2601 N. Fortune Street
      Fortunately,FL 33604
      555-838-8755 (FAX) 555-836-2522

      Record last updated on 17-May-2000.
      Record expires on 24-Jun-2009.
      Record created on 23-Jun-1997.
      Database last updated on 19-Mar-2002 04:31:00 EST.
```

Next step is a visit to the company web site. Here I find all sorts of useful information. The bios of the President, CEO, CFO, CIO and other executive staff are found on the "Company" web page: John Hutchins, CEO; Tom Castor, CFO; Meredith D. Baker, President, Don Wallace, CIO; Alec Messner, VP Sales; Ben Strong, VP Marketing. There is also link to a page with the board of directors if I want to expand my attack. Furthermore, there are enough phone numbers on

the website to help guess the phone number range if decide to use my wardialer for a modem attack to attempt a breach via a modem. I also get an additional confirmation that I'm on the right track with email because I find a few more email addresses such as contact@giacent.com , sales@giacent.comand customers@giacent.com.

The last information gathering step is to do a Google search (web and groups) using the company name and the names of the people I found on the web sites to see what other information I can learn about the company and it's employees. Through this I might learn more about the company to use in my attack or get confirmed email addresses.

The Message and Payload

The crafted message will appear to come from Microsoft and offer urgent information about a serious vulnerability in Outlook. Hopefully the recipient will click on the link to get updated. In reality it is a URL to a completely different rogue site running on one of the systems I control on the @home network, and instead of a security bulletin the rogue web server returns a web page with an ActiveX control that resets the security settings of the persons browser (but it could just as well have injected the computer with a Trojan program instead).

To lull the target recipient into thinking they are clicking on a harmless URL (Microsoft.com) the exploit uses some URL trickery. You can see an example of how it works here (the original site where it first appeared has been moved because of copyright infringement): http://www.microsoft.com&item%3Dq209354@212.254.206.213/original.html. The "@" sign in the URL causes the browser to ignore everything that comes before it treating it instead as a username. The real server and web page is what comes immediately after the "@" sign as in this example where it is "212.254.206.213/original.html." Most likely however, the non-technical recipients that are the target of the attack are unable to distinguish a malicious URL from one that is benign and I have great confidence that my attack will have a high rate of success.

To see a demonstration of this functionality you can visit the following site and at your own risk see the power of ActiveX objects (there are three different demos on this site): http://www.thur.de/~steffen/activex/index_e.html

The Attack

My first attempt will be to send a crafted email message to the email addresses of the most likely candidates on the executive staff. I used the names I found on the website and the information from the company's domain name contacts to deduce the email addresses and put them in the bcc field of the message: bstrong@giacent.com; mdbaker@giacent.com; amessner@giacent.com; tcastor@giacent.com; jhutchins@giacent.com.

After some time has gone by I send the crafted message to the remainder of the executive staff, the domain name contacts I found through

www.samspade.org's whois-lookup and I throw in a few more email addresses I picked off the website: dwallace@giacent.com; rthompson@giacent.com; thammer@giacent.com; mibrett@giacent.com; contact@giacent.com; sales@giacent.com; customers@giacent.com and jobs@giacent.com. In order to avoid being traced I will be using an SMTP relay server that I grabbed from some Spam messages I have received recently.

With the 9 to 13 or more recipients that got this message (depending on how many receive the generic email correspondence) I had very a high success rate with this attack. Viewing the web logs on my zombie web server I could see that there were seven connection attempts from the giacent.com domain of which four successfully retrieved ActiveX controls. However, there were more than 40 hits from other sites presumably because the email had been forwarded to friends and coworkers from those GIAC Enterprises employees that originally received the email. Either they were trying to be helpful or they were confused because they didn't get the Microsoft Security Bulletin page and sought help from someone else. So we see that even without a virus payload that exploits Outlook contacts to redistribute itself across the Internet, a message with malicious payload in the form of a simple URL can spread by way of social engineering across the Internet. The many virus hoaxes that continue to circulate the Internet years after they were reported as such are a sad testament to the power of social engineering.

Mitigating the Risk of This Sort of Attack

The few recipients that were not able to run the ActiveX control had most likely disabled ActiveX in their browsers or were running a browser that do not support this feature (Netscape or Opera). Those recipients that did not connect to the rogue web site were either well educated in regards to Information Security or ignored the message altogether.

The best way to stop this type of an attack from occurring is to deny ActiveX through your firewall. That however, requires that you have a proxy-based firewall that supports this feature (e.g. Gauntlet). The other alternative is to configure your workstation security policy to not allow ActiveX in the browser assuming you have a network with Windows systems and use Internet Explorer. Often however, it may be undesirable from a business perspective to stop ActiveX because it is commonly used throughout the World Wide Web including many financial institutions. Under those circumstances your best defense is to vigilantly educate your users about exercising caution when receiving unsolicited email. Make sure your users know that the IT and Security staff are responsible for updates to software on all computer systems and that users should never perform their own updates. Furthermore, any virus threats or other suspicious mail that users learn of or receive should only be forwarded to a designated email address where it can be evaluated before dissemination to the general employee public.

The message.

```
From: M1crosoft Product Security [mailto:notify@MICROSOFT.COM]
Sent: Thursday, February 16, 2002 10:35 AM
To: MICROSOFT-SECURITY@ANNOUNCES.MICROSOFT.COM
Subject: Microsoft Security Bulletin MSO2-O12 (version 2.0)

The following is a Security Bulletin from the M1crosoft Product
Security Notification Service.

Please do not reply to this message, as it was sent from an
unattended mailbox.
                     ********************************

-----BEGIN PGP SIGNED MESSAGE-----

---------------------------------------------------------------------------
Title:      Specially Formed URL in HMTL Mail can Execute in
            0utlook when viewing mail
Date:       02 January  2001
Revision:   28 February 2002 (version 2.0)
Software:   M1crosoft 0utlook 97, 2000, XP
Impact:     Instantly Run Code of Attacker's Choice
Max Risk:   Critical
Bulletin:   MSO2-O12

M1crosoft encourages customers to immediately review the Security
Bulletin at:
http://www.microsoft.com-security-bulletin-MSO2-
O12@209.19.248.88/bulletin.html
---------------------------------------------------------------------------

Issue:
======
0utlook runs script contained in email when previewing messages so
that a connection is made to a website where code of the attacker's
choice is run.

A flaw exists in the way Outlook handles script characters in
messages inconjunction with Internet Explorer (IE). If an HTML
message that
contains specially formatted URL is previewed 0utlook, the URL is
launched and the script is excecuted.

Risk Rating:
============
 - Client systems: Critical

Patch Availability
==================
A patch is available to fix this vulnerability. Please read the
Security Bulletin at:
http://www.microsoft.com-security-bulletin-MSO2-
O12@209.19.248.88/bulletin.html
```