Global Information Assurance Certification Paper

Firewalls, Perimeter Protection, and VPNs
(GCFW Practical Assignment, Version 1.6e)

February 2001

David B. Koconis

Assignment 1 – Security Architecture

Overview

In spite of all the Wall Street analysts' dour predictions for online fortune cookie market segment and in the presence of an increasingly slowing economy, GIAC Enterprises has enjoyed considerable success. They have leveraged the advantage gained by their first-on-the-web presence into a thriving e-business. Customers flock to their web site to purchase fortune cookie sayings. Along the way, they have established business relationships with cookie saying suppliers who work offsite and international partners who translate and resell sayings overseas. This section outlines the general network architecture and the access requirements and restrictions for each of these groups, as well as internal GIAC Enterprises employees.

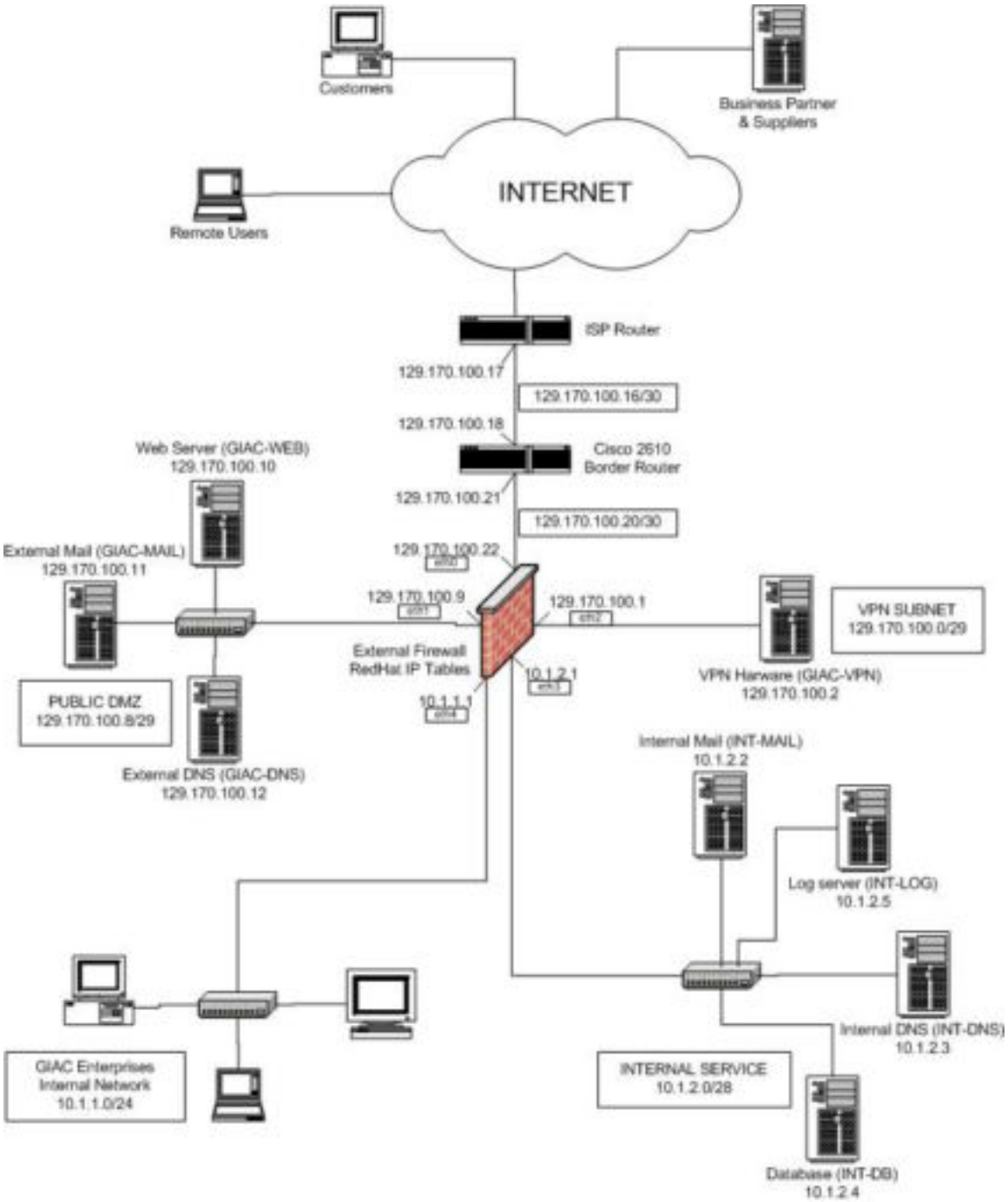
When deciding whether to implement a particular network security defense, the business needs of GIAC Enterprises are always considered. Is the cost of losing access to a given service or exposing some sensitive data less than the cost of implementing the defense to protect the service or data? Keeping this in mind, GIAC Enterprises makes every effort to adhere to the following network security, best-practice policies:

- Defense-in-depth
  Where feasible, multiple layers of defense against unauthorized intrusion will be used. For example, the same filtering rule may exist on the border router and external firewall, so that the offending network traffic would be blocked in the case where either device failed to work properly or became compromised.
- Principle of Least Privilege
  Members of each group are allowed to access only the services and systems they need to successfully perform functions required of them and restricted from accessing any other services
- Isolate systems providing network services
  Systems offering services to external or internal users will be on dedicated service networks and will not be dual-use. For example, the system hosting the publicly available web server will be on a screened subnet and will only host the web server. It will not host DNS, Mail, or any other service.

General Network Architecture

The network architecture for GIAC enterprises is shown in Figure 1. To support their Internet presence, they have leased small subnet of sixteen publicly routable IP addresses from an ISP (129.170.100.0/28) for assignment to publicly accessible servers. These addresses have been assigned as follows: the lower half of the subnet (129.170.100.0/29) is used for a VPN server subnet and the upper half of the subnet (129.170.100.8/29) is used for servers on the publicly accessible service network (referred to as the DMZ from now on). Eight additional IP addresses are masked into two small subnets for the connections between the border router and ISP (129.170.100.16/30) and the border router and GIAC firewall (129.170.100.20/30).

Figure 1 - GIAC Enterprises Network Architecture



At the external boundary between GIAC Enterprises and the ISP, passing all network traffic to and from the ISP router, is the GIAC border router (Cisco 2610 with a T1 CSU/DSU module for the PPP connection to the ISP and an Ethernet interface to the GIAC Enterprises network). Directly behind the router on the GIAC Enterprises, side is a firewall running NetFilter iptables v1.2.4-2 installed via RPM (GIAC-FW). The firewall has five network interfaces: one connected to the router (129.170.100.22), one

connected to the DMZ (129.170.100.9), one connected to an isolated VPN service subnet (129.170.100.1), one connected to an internal service subnet (10.1.2.1), and one connected to the internal employees subnet (10.1.1.1). The firewall performs network address translation (NAT) so that systems with private IP addresses on the GIAC internal network can have limited access to the Internet. By default, hiding NAT is used so that none of the systems are accessible from the Internet, however static NAT can be enabled to provide access on an as needed basis. Each of these subnets is discussed in further detail below.

Public DMZ (129.170.100.8/29)

The public DMZ subnet hosts all the systems that provide services to the general public user on the Internet. It includes the public web server (GIAC-WEB), external mail server (GIAC-MAIL), and external DNS server (GIAC-DNS). All systems are running RedHat Linux 7.2 patched with the latest RPM updates from http://ftp.updates.redhat.com as of 12-FEB-02 and each has OpenSSH 2.9p2-12 client and server installed so that administrators can remotely access and configure them. All are configured to send log data to the GIAC log server (see INT-LOG below) and none offers any network services that are not required by its primary function.

GIAC-WEB has Apache 1.3.22-2 and mod_ssl 2.8.5-1 installed and accepts HTTP and HTTPS connections. During normal operations, the server occasionally needs to complete connections to the fortune cookie saying database on the internal service network (see Internal Service Network and Business Operation Overview for more details). Different areas of the GIAC website are made available to different groups of users depending on their business needs. For example, the pages with general company information and marketing material are made available via unauthenticated, unencrypted HTTP to anyone on the Internet while the pages used by the business partners to deliver translations of sayings are only available via authenticated, encrypted HTTPS. Finally, in order to accept credit card payments, GIAC Enterprises has contracted with an online credit card processing service. To complete the processing, GIAC-WEB initiates HTTPS connections to the credit card processing server.

GIAC-DNS has BIND 9.1.3-4 installed and accepts UDP and TCP connections to port 53 from anywhere. GIAC Enterprises employs split-horizon DNS [1], therefore GIAC-DNS hosts DNS records only for GIAC systems directly accessible from the Internet. Any IP resolution requests for systems with private addresses are forwarded to the internal DNS server.

GIAC-MAIL has sendmail-8.11.6-3 installed, accepts SMTP connections from anywhere, and acts solely as a mail relay. Email from the Internet destined for any GIAC Enterprise employee mailbox is relayed to the internal mail server. Email from the internal mail server is relayed to the Internet. Relaying of all other email is denied.

VPN Subnet (129.170.100.0/29)

The GIAC-VPN server resides on its own subnet off the main firewall, isolated from the systems on the public service network. The primary reason for this to reduce the risk of

exposing encrypted data should one of the other systems in the DMZ be compromised [2].

Internal Service Network (10.1.2.0/28)

The internal service network includes the internal mail server (INT-MAIL), the internal DNS server (INT-DNS), the database server (INT-DB), and the log server (INT-LOG). As with the system on the public DMZ subnet, all systems are running RedHat Linux 7.2 patched with the latest RPM updates from http://ftp.updates.redhat.com as of 12-FEB-02 and each has OpenSSH 2.9p2-12 client and server installed so that administrators can remotely access and configure them. All are configured to send remote log data to the GIAC log server (with the exception of the INT-LOG server itself, which logs locally) and none offers any network services that are not required by its primary function.

INT- MAIL has sendmail-8.11.6-3 installed, accepts SMTP connections from internal users and GIAC-MAIL, and supports POP3 and IMAP so internal users can use popular email clients to retrieve their mail.

INT-DNS has BIND 9.1.3-4 installed and accepts UDP and TCP connections to port 53 from GIAC-DNS as well as lookup requests for users on the internal GIAC network.

INT-DB has MySQL server 3.23.41-1 installed and hosts the GIAC Enterprises fortune cookie sayings database as well as databases with account data for customers, partners and suppliers. INT-DB accepts connections on port 3306 that originate from the GIAC Internal Network only. Since MySQL sends authentication information clear text over the network, connection requests from the web server on the public DMZ are tunneled through the external firewall using an SSH connection that forwards port 3306 on GIAC-WEB to port 3306 on INT-DB as described in [3].

INT-LOG has sysklogd 1.4-7 installed and configured to accept remote log messages on UDP port 514. It serves as the central collection point for log data generated by all systems on the internal and external service networks. To automate log event monitoring, logcheck is installed and configured to review logs hourly for suspicious events. INT-LOG also has ntp 4.0.1-4 installed and is the time NTP server used to synchronize system clocks at GIAC Enterprises so log data from different machines can be correlated.

GIAC Internal Network (10.1.1.0/24)

The GIAC internal network includes the systems of all GIAC employees. The policy of GIAC Enterprises is to disallow all connection attempts originating from outside the internal network destined for inside the internal network.

Business Operations Overview (access requirement and restrictions)

During the course of normal business operations, each of the following groups requires access to the resources on the GIAC network: the public Internet, customers, suppliers, partners, and GIAC employees. In accordance with the principle of least privilege with

regards to accessing network resources, members of each group are allowed to access only the services and systems they need to successfully perform functions required of them and restricted from accessing any other services. A description of the needs and restrictions for each group follows.

The public Internet

Everyone on the Internet is granted limited access into the GIAC Enterprises network. They can initiate HTTP connections to GIAC-WEB to view selected pages with general company information and marketing material. They can initiate SMTP connections to GIAC-MAIL to send mail to employees in the giac.com domain. They can initiate DNS queries to GIAC-DNS to resolve the IP addresses for GIAC systems with routable addresses. Since no other direct connections originating from the public Internet are necessary for normal business operations, none are allowed.

Customers

In addition to the access granted to the public Internet, customers or companies purchasing bulk fortunes online require direct connection to GIAC-WEB via HTTPS protocol (port 443). GIAC proprietary CGI applications accept a customer's account name and password, validate the information against account data stored on INT-DB, and generate a time-sensitive session ID that is passed back to the customer's browser and used for the remainder of the session. During the session, GIAC-WEB initiates at most two different connections with data provided by the customer: 1) MySQL queries (tunneled through SSH) to INT-DB to validate authentication credentials and display cookie sayings options and prices and 2) HTTPS connections to the credit card processing server for obtaining payment approval. Since no other direct connections originating from customers' systems are necessary for online purchasing, none are allowed.

Suppliers and Partners

In addition to the access granted to the public Internet, fortune cookie sayings suppliers and international partners that translate and resell fortunes require direct connection to GIAC-WEB via HTTPS protocol (port 443). As with customers, GIAC proprietary CGI applications accept a supplier's or partner's account ID and password, validate the information against account data stored on INT-DB, and generate a time-sensitive session ID that is passed back to the supplier's browser and used for the remainder of the session. Because suppliers and partner's require, and are granted, permission to upload sensitive data into the cookie sayings database, GIAC Enterprises has implemented additional security to access their area of the web site. Authentication via client certificate is enforced by the Apache configuration file for these web pages [4]. Since no other direct connections originating from suppliers' or partners' systems are necessary for providing fortunes, none are allowed.

GIAC Employees

While on-site, GIAC employees on the internal network utilize the services on systems in the internal service network for day-to-day business operations. The require access via

IMAP, POP3, and SMTP protocols to INT-MAIL to retrieve and send email. They require access via DNS protocol to INT-DNS to resolve IP addresses for any hostname they wish to contact. Certain employees require access to the MySQL database on INT-DB to review and make updates to customer, supplier and partner account records.

In addition limited access to the public Internet is granted to GIAC employees. GIAC Enterprises understands the importance of the public Internet as a valuable reference tool and encourages employees to take advantage of this vast repository of freely available knowledge. They are permitted to surf the Internet from the internal network (i.e. initiate HTTP and HTTPS connections), initiate anonymous FTP sessions (typically required to download software updates), and initiate SSH shell sessions to remote systems.

Shell access, via SSHv2 only, from the internal network is required by administrators to all the systems on the internal service network, the public DMZ, and the VPN subnet.

While off-site, select GIAC employees are granted enough access to the internal network to enable sending and receiving of email. This is done by creation of an IPSec connection using tunnel mode ESP to the VPN server. The firewall allows ports 25 (SMTP), 110 (POP3), and 143(IMAP) to INT-MAIL from the VPN subnet interface.

Assignment 2 – Security Policy

External Border Router (GIAC-BR) Policy

As mentioned earlier, GIAC-BR is a CISCO 2610 router. GIAC employs extended access lists on their border router to enable filtering based on port numbers. The border router is used exclusively for static packet filtering. ACL 100 is applied inbound to the Ethernet interface coming from GIAC-FW and ACL 101 is applied inbound to the T1 CSU/DSU interface going to the ISP (and Internet). Each rule is followed by an brief explanation preceded by a !, the comment character used by the CISCO IOS config file format.

```
!
! Access Control List 100 (coming in from internal network)
!
no access-list 100
access-list 100 deny ip 129.170.100.16 0.0.0.3 any log
! deny and log traffic with source ip of network connecting to ISP
access-list 100 permit icmp host 129.170.100.22 any
! allow ICMP from source IP of eth0 on firewall
access-list 100 permit icmp 129.170.100.0 0.0.0.15 any
! allow ICMP from source IP of public DMZ or VPN
access-list 100 permit tcp host 129.170.100.11 25 any 25
! allow SMTP from source IP of GIAC-MAIL
access-list 100 permit tcp host 129.170.100.12 any 53
! allow DNS (tcp) from source IP of GIAC-DNS
access-list 100 permit udp host 129.170.100.12 any 53
! allow DNS (udp) from source IP of GIAC-DNS
access-list 100 permit tcp host 129.170.100.22 any 80
! allow HTTP from source IP of eth0 on firewall
```

```
access-list 100 permit tcp host 129.170.100.22 any 443
! allow HTTPS from source IP of eth0 on firewall
access-list 100 permit tcp host 129.170.100.22 any range 20 22
! allow FTP/FTP data/SSH from source IP of eth0 on firewall
access-list 100 permit tcp host 129.170.100.10 host 216.136.147.22 443
access-list 100 permit tcp host 129.170.100.10 host 216.136.147.18 443
! allow HTTPS from source IP of GIAC-WEB to credit card servers
access-list 100 permit tcp any any established
! allow established connections (with ACK and/or RST set)
access-list 100 deny ip any any log
! deny and log anything else

!
! Access Control List 101 (coming in from ISP)
!
no access-list 101
access-list 101 deny ip 129.170.100.0 0.0.0.15 any log
access-list 101 deny ip 129.170.100.20 0.0.0.3 any log
! deny and log anything with source IP inside router
access-list 101 permit icmp any host 129.170.100.22
! allow ICMP with destination IP of eth0 on firewall
access-list 101 permit icmp any 129.170.100.0 0.0.0.15
! allow ICMP with destination IP of public DMZ or VPN
access-list 101 permit tcp any host 129.170.100.11 eq 25
! allow SMTP to destination IP of GIAC-MAIL
access-list 101 permit tcp any host 129.170.100.11 eq 113
! allow IDENT to destination IP of GIAC-MAIL
access-list 101 permit tcp any host 129.170.100.10 eq 80
! allow HTTP to destination IP of GIAC-WEB
access-list 101 permit tcp any host 129.170.100.10 eq 443
! allow HTTPS to destination IP of GIAC-WEB
access-list 101 permit udp any host 129.170.100.12 eq 53
! allow DNS (udp) to destination IP of GIAC-DNS
access-list 101 permit tcp any host 129.170.100.12 eq 53
! allow DNS (tcp) to destination IP of GIAC-DNS
access-list 101 permit 50 any host 129.170.100.2
! allow IPSec ESP protocol to destination IP of GIAC-VPN
access-list 101 permit udp any host 129.170.100.2 eq 500
! allow IKE to destination IP of GIAC-VPN
access-list 101 permit tcp any any established
! allow established connections (with ACK and/or RST set)
```

Notes:

1. Router ACLs have an implicit deny as the last rule, so one is not needed at the end of 101 in order to deny unwanted traffic. The explicit one at the end of list 100 causes unexpected traffic out of the GIAC network to be logged.
2. FTP, HTTP, and HTTPS traffic is allowed to the eth0 interface on the firewall to enable internal GIAC employees to access the Internet because the IP address of that interface is the source IP used by the hiding NAT (discussed later).
3. IDENT traffic is allowed through so the firewall can send back an immediate reset and avoid a possible email delivery delay for SMTP connections [5].

In addition to the ACLs filtering inbound and outbound traffic several other configuration options are present in the config file to harden the router and be a good network neighbor. These recommendations are taken directly from the GCFW course notes [6].

```
enable secret
service password encryption
! store password as MD5 has in config file (default is plain text)
no snmp
! Disable SNMP (especially important in light of recent vulnerabilities)
no ip source-route
! Disallow source routed packets
no service tcp-small-servers
no service udp-small-servers
! Disable small services (echo, discard, chargen, and daytime)
no service finger
! Disable finger service
no ip http
no ip bootp
! Do not start HTTP or BOOTP servers on the router
no cdp
! Disable CISCO discovery protocol
no ip direct-broadcast
! help prevent SMURF attacks by disabling broadcast packets to remote networks
no ip unreachables
banner / WARNING: Authorized Access Only /
! Add a warning banner
logging 129.170.100.22
! Log messages to INT-LOG (As discussed later, the firewall will NAT these)
```

Note:

- The syslog server does not reside on the firewall itself. Instead, traffic to UDP port 514 destined for the IP address of the firewall interface is statically NAT'ed by the firewall to have the private destination IP address of INT-LOG (see end of script in Appendix B)

External Firewall (GIAC-FW) Policy and Tutorial

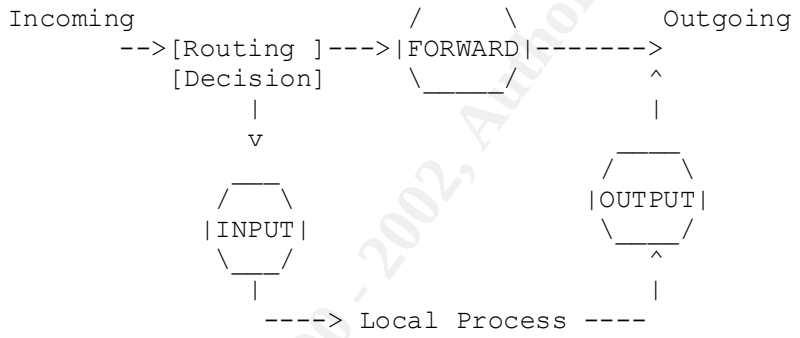
Background Information

As mentioned earlier, GIAC-FW is a RedHat Linux system running netfilter/iptables (from now on referred to simply as iptables). A very good description of the package is found at the project home page (http://netfilter.samba.org/) and reproduced here

> The netfilter/iptables project is the Linux 2.4.x / 2.5.x firewalling subsystem. It delivers you the functionality of packet filtering (stateless or stateful), all different kinds of NAT (Network Address Translation) and packet mangling

Two things to highlight from the above description are that iptables is capable of performing stateful inspection of packets and network address translation. While iptables does not provide a fancy graphical user interface, it does include more than enough capability to enable GIAC Enterprises to enforce the access policies outlined for each of the various groups discussed in the Security Architecture section. In addition, the functionality is built into the Linux kernel, so its performance is very good.

Fundamental to the understanding of implementing security policies using iptables is the concept of a rule chain, usually referred to simply as a chain. A rule chain is simply an access control list (ACL) containing rules to which a packet is compared. Upon matching a rule, a defined action is taken (e.g. DROP the packet, ACCEPT the packet, jump to another chain for additional inspection, etc.). If the packet reaches the end of the chain and no rules have been matched, the default action (also called policy) for the chain is performed. As described in the figure and text below (extracted from [7]), there are three built-in chains: INPUT, FORWARD, and OUTPUT.



1. When a packet comes in (say, through the Ethernet card) the kernel first looks at the destination of the packet: this is called `routing'.
2. If it's destined for this box, the packet passes downwards in the diagram, to the INPUT chain. If it passes this, any processes waiting for that packet will receive it.
3. Otherwise, if the kernel does not have forwarding enabled, or it doesn't know how to forward the packet, the packet is dropped. If forwarding is enabled, and the packet is destined for another network interface (if you have another one), then the packet goes rightwards on our diagram to the FORWARD chain. If it is ACCEPTed, it will be sent out.
4. Finally, a program running on the box can send network packets. These packets pass through the OUTPUT chain immediately: if it says ACCEPT, then the packet continues out to whatever interface it is destined for.

The user is free to create additional user-defined chains, which are primarily used to make the overall rule-set compartmentalized and more readable. Also, there are two special chains for implementing Network Address Translation named PREROUTING and POSTROUTING. PREROUTING rules are processed before the routing decision is made and POSTROUTING rules are processed just before the outgoing packets leave the interface.

Finally, a brief description of the syntax used for iptables is included. For a detailed explanation of the syntax of the iptables command refer to [7,8,9]. The majority of the

time, a command is written to append a rule to the end of the rule-set and it follows the format:

iptables -A *CHAIN* -i *IF* -p *PROTO --sport PORT1 --dport PORT2* -s *SOURCE* -d *DEST*
-j *ACTION* -m state --state *LIST*

where

-A *CHAIN*	Add the rule to the end of the built-in or user-defined *CHAIN* (e.g. FORWARD)
-i *IF*	arriving interface of the packet is *IF* (e.g. eth0)
-p *PROTO*	packet protocol is PROTO (e.g. tcp)
--sport *PORT1*	(valid for tcp & udp) packet source port is *PORT1*
--dport *PORT2*	(valid for tcp & udp) packet destination port is *PORT2*
-s *SOURCE*	host or network where packet originated is *SOURCE* (e.g. *GIAC-WEB or10.1.1.0/24*)
-d *DEST*	host or network for packets final destination is *DEST* (e.g. *INT-DB or10.1.2.0/28*)
-m state	keep track of this connection in the state table
--state *LIST*	types of connection states that should match (e.g. *NEW*, *ESTABLISHED*)
-j *ACTION*	what should be done with packets that match (e.g. *ACCEPT*, *REJECT*, *CHAIN*)
	If *CHAIN* is specified, packet jumps to beginning of *CHAIN* for further processing.

If any option is omitted from the command line, then all valid values are matched. For example, if "-d *DEST*" is left out of a rule, then packets to any destination will match the rule. There are of course other flags to insert (-I), delete (-D), and replace (-R) rules in a chain, but due to the implementation procedure described in the next section, they seldom need to be used.

Rule-Set and Discussion

The complete policy rule-set for the firewall that enforces the access restrictions discussed in assignment one is shown in Appendix A.

Since GIAC Enterprises only allows shell access to the firewall from the system's console, the INPUT and OUTPUT chains are uninteresting. The first rule in both chains allows traffic where the sourece and destination are both the loopback interface. The INPUT chain includes a rule allowing NTP traffic in to the firewall from the GIAC log server INT-LOG. The OUTPUT chain includes a rule allowing syslog traffic out to INT-LOG. Finally, both chains then have the same rules to allow ICMP traffic so the firewall can ping other systems (see HANDLE_ICMP chain), weed out traffic GIAC is not interested in logging (see WEED_OUT chain), log then reject everything else. Note that a catch-all REJECT rule is included for insurance, even though the policy for both chains is to DROP packets that reach their ends without matching any rule. Of course, order is important, since all packets will match the REJECT rule, any rules that ACCEPT packets after it will never be matched.

The firewall configuration used by GIAC Enterprises primarily uses the built-in FORWARD chain and user-defined chains to perform it's role as network traffic cop. The logic goes as follows, after handling ICMP packets, rules in the FORWARD chain determine which user-defined chain to use to further process the packet solely based on its arriving interface, source IP address and destination IP address. As with the other

built-in chains, the FORWARD chain ends with rules to weed out, log and reject any packet that has not matched a previous rule.

User-defined chains implement the rules enforcing the security policy discussed in section 1and have self-explanatory names:

TO_BR - traffic destined for the border router (i.e. Internet)
TO_DMZ - traffic destined for the public DMZ
TO_VPN - traffic destined for the VPN gateway
TO_INTDMZ - traffic destined for the internal service network

Note that there is no chain for traffic destined to the internal GIAC network because GIAC policy does not allow connections into this subnet that have been initiated from outside the subnet. Any reply packets for established or related connections initiated by systems in the GIAC internal network will be accepted based on their state table entries. For further explanation of any rule in th FORWARD chain or any of the user-defined chains, please see the comments in the script implementing the policy (Appendix B).

Implementation Details

Since the rule-set is resident in kernel memory, the firewall server must be configured so that the rule-set is loaded automatically during the boot process. Furthermore, to ensure there is no period of time in which network connectivity is enabled without filtering rules in place, loading of the rule-set occurs prior to bringing up any of the network interfaces. The RedHat iptables start-up script (*/etc/rc.d/init.d/iptables*) runs just before the network start-up script (*/etc/rc.d/init.d/network*) and attempts to load rule-sets found in a file named */etc/sysconfig/iptables*, if it exists. The format of this file is not designed for human readability and editing this file to add or change firewall policy is prone to errors. Therefore, the customary procedure (and the one used in this paper) is to create a heavily commented script to implement the rule-set, load it into a running kernel, then generate the */etc/sysconfig/iptables* file automatically with the "*/etc/rc.s/init.d/iptables save*" command. Future additions or modifications to the firewall policy should be made to the script. To install the new rules in kernel memory, simply run the script again. If the new rules perform as desired, save them using the *save* command above which will overwrite the old rule-set in the */etc/sysconfig/iptables* file. The complete, uninterrupted script is given in Appendix B, and sections of it are repeated throughout this tutorial, as needed.

To implement the policy follow these steps:

1. Verify that the kernel is compiled with support for iptables
   Modular support for iptables is compiled into the kernel RPM installed by the RedHat distribution. To ensure that a custom built kernel contains the correct config options, see [8].
2. Ensure that iptables is installed and configured to start
   The iptables RPM installed by the RedHat distribution should take care of the details of putting the necessary files in the right places. Make sure iptables is configured to start by issuing the command:

```
[root@GIAC-FW /root]# /sbin/chkconfig iptables on
```

Note that the modules for ipchains, the older Linux kernel firewall, and iptables are not compatible and cannot be loaded simultaneously. If ipchains is installed, be sure that ipchains is not configured to start by issuing the command:

```
[root@GIAC-FW /root]# /sbin/chkconfig --level 0123456 ipchains off
```

3. Ensure that the firewall is configured to send kernel log messages from iptables to INT-LOG by including the following line in the */etc/syslog.conf* file

   kern.* @10.1.2.5

4. Copy the script in Appendix B to the firewall server and execute it to load the rule-set
5. Review the rule-set for errors/omissions with the command

   ```
   [root@GIAC-FW /root]# /sbin/iptables -L -n -v
   ```

   (list rules verbosely and do not resolve IP or ports to names)
6. Test that rules are working properly. Here are three examples

   RULE Description:
   FTP traffic from internal network is allowed to Internet.
   iptables command syntax:

   ```
   iptables -A TO_BR -p tcp --dport 21 -s 10.1.1.0/24 -m state \
               --state NEW,ESTABLISHED,RELATED -j ACCEPT
   ```

   Test procedure:
   From a system on the GIAC internal network, ftp to an Internet server (like ftp.redhat.com), connect and download a file.
   Expected result:
   FTP connection should be allowed and file download should succeed since the FTP data connection is related to the initial port 21 connection.

   RULE Description:
   Attempt to open a connection to port 113 (IDENT) on GIAC-MAIL from the Internet.
   iptables command syntax (matches the end of the TO_DMZ chain):

   ```
   $IPTABLES -A TO_DMZ -p tcp --dport 113 -j REJECT --reject-with
   tcp-reset
   ```

   Test:
   From a system with a non-GIAC IP address on the Internet, use telnet command to attempt to connect to port 113 on GIAC-MAIL.
   Expected result:
   Connection should be blocked (i.e. no packet arrives at GIAC-MAIL interface). Originating system should receive a TCP reset, immediately breaking down the connection.

   RULE Description:
   SMTP traffic from the GIAC internal network is not allowed to the DMZ mail server GIAC-MAIL (matches end of TO_DMZ chain).
   iptables command syntax:

   ```
   $IPTABLES -A TO_DMZ -j $LOGFLAG --log-prefix "END TO_DMZ CHAIN:"
   $IPTABLES -A TO_DMZ -j REJECT
   ```

   Test procedure:
   From an internal GIAC system, atempt to telnet to port 25 on GIAC-MAIL.
   Expected result:

Should receive an ICMP port unreachable message. Log message prefixed by 'END TO_DMZ CHAIN' should be sent to the log file on INT-LOG designated for kernel facility messages.

7. Store the rule-set with the command
   ```
   [root@GIAC-FW /root]#/etc/rc.d/int.d/iptables -save
   ```
8. Reboot the server and verify that rules get loaded automatically when system boots by listing them again.

VPN Server (GIAC-VPN) Policy

The VPN Server is a RedHat 7.1 Linux system running Free/SWAN [10]. RPM packages containing necessary kernel modifications and userland utilities can be downloaded from http://rpms.steamballoon.com/freeswan. Client systems are traveling users' laptops running primarily Windows 2000. They have Sentinel Internet Pilot v1.2.3 software from SSH Communications Security (http://www.ipsec.com) installed to enable tunnel creation.

Secure remote access for GIAC Enterprises employees connecting from the Internet is accomplished as follows. After getting appropriate managerial level approval, a shared secret is generated for the employee. The following line is added to the */etc/ipsec.secrets* file on the VPN server to allow connections from any IP address on the Internet:

```
129.170.100.2 0.0.0.0: PSK "really random shared-secret string"
```

The shared secret is loaded onto the employees laptop, along with a stern warning about the importance of keeping it secure. The procedure to load the secret on the laptop is covered in detail in the Sentinel documentation [11] and by Napier in [12,13].

The VPN connection description for remote users is located in /etc/ipsec.conf and reads:

```
conn giac-remote
      type=tunnel
      left=129.170.100.2
      leftnexthop=129.170.100.1
      leftsubnet=10.1.2.0/28
      right=0.0.0.0
      keylife=30m
      authby=secret
      auto=add
```

The configuration defines the IP address, next hop and protected subnet for the left participant (the VPN server). The right participant (the remote GIAC employee) can come from any network. The only method of key exchange supported by Free/SWAN is IKE and the key life is set to 30 minutes. Renegotiation of the connection keys will occur every 30 minutes with perfect forward secrecy enabled (the Free/SWAN default). Therefore, compromise of any given connection key will only compromise the data protected by that key, not future connections. Tunnel mode must be used because the VPN server acts as a gateway to the private internal service network, and, therefore the

ultimate destination of a packet (INT-MAIL) has a different destination IP address than the IPsec gateway itself. Finally, in order to avoid having the remote GIAC user's email account password and username, as well as, the contents of his email traveling across the Internet in clear text for anyone to see, encapsulating security payload (ESP) is used. ESP provides encryption of the entire packet payload.

The firewall and VPN server are configured to accept packets with any source IP address using IKE (udp port 500) and ESP (ip protocol 50). However, the VPN server will create connections only for packets from systems with a pre-shared secret listed in the */etc/ipsec.secrets* file. These packets will be decrypted and sent back through the firewall on their way to their final destination. In accordance with firewall policy, only packets destined for the internal mail server ports 25 (SMTP), 110 (POP3), and 113 (IMAP) will be alowed out of the VPN subnet. Remote users attempting to access other services on other systems or subnets will be denied.

Assignment 3 - Audit the Security Architecture[1]

GIAC Enterprises requires an audit of their primary firewall in order to be eligible for discounted insurance rates to limit losses caused by unauthorized network intrusions. A description of the audit process follows.

Plan the audit

The audit to verify that the primary firewall is properly implementing GIAC Enterprises' security policy must include steps to test for the following general items:

1. Packets that should get dropped are dropped
2. Any activity that should get logged does get logged

There are two phases to the audit: passive network observation and active scanning. In the passive network observation phase, samples of network traffic will be collected separately from each subnet. The captured samples will be reviewed for packets that should have been dropped by the firewall, but were not, indicating that the firewall is not enforcing policy correctly. In the active scanning phase, each service system will be scanned from the Internet, public DMZ, VPN, internal DMZ, and GIAC internal subnets. During the scanning, packets on the destination subnet with a source IP address of the scanning system will be captured. Output from the scans, log entries recorded during scanning, and packets capture during scanning will be reviewed to determine if the firewall is enforcing policy and logging properly.

The passive observation phase should be during normal business hours when network traffic is usually at its peak. This phase should not interfere with business operations since it will not generate any new packets. At least three, one-hour snapshots, taken at different times of day over the span of a week, should be captured for each subnet.

[1] Not having access to enough hardware to actually conduct the audit limited my ability to complete the requirements of this assignment as written. As a next best thing, I explain the steps, commands, and expected results in enough detail that such and audit could be performed on the actual hardware.

The active scanning should not disrupt GIAC Enterprises normal business operations either since unusual or unexpected packets should be dropped by the firewall before reaching any of the systems. However, it should be carried out at a time when it will cause minimal disruption should any problems occur. The best times are third shift or pre-arranged hours on a weekend or vacation day. Administrators responsible for the systems should be notified in advance and necessary personnel should be present to recover from any unexpected down-time event.

The total time required by the audit is approximately 40 hours (@ 250$/hour estimated cost will be $10,000). The effort breaks down as follows:

```
TCPDUMP filter rule creation            -     3 hours
Scanning script planning and creation   -     2 hours
Equipment setup                         -     5 hours
Running tests and collecting data       -    20 hours
Post processing and evaluating results  -    10 hours
```

The equipment required for the audit is two hubs and two laptops. The laptops should have network scanning (e.g. nmap) and packet capturing (e.g. tcpdump) software.

Conduct the audit - Passive Phase

Start by connecting a listener laptop to a hub placed on the public DMZ subnet between the firewall interface and the switch. Use the following list as a guide to develop the tcpdump filter file that screens out packets know to be allowed on this subnet by the GIAC firewall policy (named *publicDMZ.filter*).

```
Allowed packets on public DMZ:
  tcp from NET_DMZ to anywhere with SYN bit set
  any protocol from NET_DMZ to NET_DMZ
  udp from NET_DMZ to anywhere
  icmp from NET_DMZ to anywhere
  tcp from GIAC Internal ports 1024:65535 to NET_DMZ port 22 (and returning)
  tcp from GIAC Internal ports 1024:65535 to GIAC_WEB port 80, 443 (and returning)
  tcp from INT-MAIL ports 1024:65535 to GIAC-MAIL port 25 (and returning)
  tcp from INT-DNS ports 1024:65535 to GIAC-DNS port 53 (and returning)
  udp from INT-DNS ports 1024:65535 to GIAC-DNS port 53 (and returning)
  tcp from any external ports 1024:65535 to GIAC-MAIL port 25 (and returning)
  tcp from any external ports 1024:65535 to GIAC-DNS port 53 (and returning)
  udp from any external ports 1024:65535 to GIAC-DNS port 53 (and returning)
  tcp from any external ports 1024:65535 to GIAC-WEB port 80, 443 (and returning)
```

Capture packets using the following tcpdump command:

```
tcpdump -n -i eth0 -F publicDMZ.filter -w publicDMZ.capture

      -i    laptop interface connected to target subnet
      -n    do not resolve IP addresses
      -F    use named file to filter capture
      -w    write packets to file (not stdout)
```

Next move the listener laptop to a hub placed on the VPN subnet between the firewall interface and the switch. Use the following list as a guide to develop the tcpdump filter

file that screens out packets know to be allowed on this subnet by the GIAC firewall policy (named *VPN.filter*).

```
Allowed packets on VPN subnet:
  tcp from GIAC_VPN to anywhere with SYN bit set
  udp from GIAC_VPN to anywhere
  icmp from GIAC_VPN to anywhere
  tcp from GIAC Internal ports 1024:65535 to GIAC_VPN port 22 (and returning)
  ip protocol 50 from any external to GIAC_VPN (and returning)
  udp from any external port 500 to GIAC_VPN port 500 (and returning)
```

Capture packets using the tcpdump command given for the public DMZ, substituting VPN subnet filenames, as appropriate.

Next move the listener laptop to a hub placed on the internal service subnet between the firewall interface and the switch. Use the following list as a guide to develop the tcpdump filter file that screens out packets know to be allowed on this subnet by the GIAC firewall policy (named *INTDMZ.filter*).

```
Allowed packets on internal service network:
  tcp from NET_INTDMZ to anywhere with SYN bit set
  any from NET_INTDMZ to NET_INTDMZ
  udp from NET_INTDMZ to anywhere
  icmp from NET_INTDMZ to anywhere
  tcp from GIAC Internal ports 1024:65535 to NET_INTDMZ port 22 (and returning)
  tcp from GIAC Internal ports 1024:65535 to INT_MAIL port 25 (and returning)
  tcp from GIAC Internal ports 1024:65535 to INT-DNS 53 (and returning)
  udp from GIAC Internal ports 1024:65535 to INT-DNS 53 (and returning)
  tcp from GIAC Internal ports 1024:65535 to INT-DB port 3306 (and returning)
  tcp from GIAC-VPN ports 1024:65535 to INT-MAIL port 25 (and returning)
  tcp from GIAC-VPN ports 1024:65535 to INT-MAIL port 110,143 (and returning)
  udp from GIAC-VPN ports 1024:65535 to INT-LOG port 123,514 (and returning)
  tcp from GIAC_WEB ports 1024:65535 to INT-DB port 22 (and returning)
  tcp from GIAC-MAIL ports 1024:65535 to INT-MAIL port 25 (and returning)
  tcp from GIAC-DNS ports 1024:65535 to INT-DNS port 53 (and returning)
  udp from GIAC-DNS ports 1024:65535 to INT-DNS port 53 (and returning)
  udp from NET_DMZ  ports 1024:65535 to INT-LOG port 123,514 (and returning)
  udp from GIAC-BR ports 1024:65535 to INT-LOG port 514 (and returning)
  udp from GIAC-FW ports 1024:65535 to INT-LOG port 123 (and returning)
```

Capture packets using the tcpdump command given for the public DMZ, substituting internal service subnet filenames, as appropriate.

Next move the listener laptop to a hub placed on the GIAC internal subnet between the firewall interface and the switch. Use the following list as a guide to develop the tcpdump filter file that screens out packets know to be allowed on this subnet by the GIAC firewall policy (named *GIACINT.filter*).

```
Allowed packets on the GIAC internal network
  tcp from NET_INTNET to anywhere with SYN bit set
  any from NET_INTNET to NET_INTNET
  udp from NET_INTNET to anywhere
  icmp from NET_INTNET to anywhere
  tcp from NET_INTNET ports 1024:65535 to !GIAC ports 20,21,22,80,443 (and returning)
  tcp from NET_INTNET ports 1024:65535 to GIAC-WEB ports 80,443 (and returning)
  tcp from NET_INTNET ports 1024:65535 to INT-MAIL ports 25,110,143 (and returning)
  tcp from NET_INTNET ports 1024:65535 to INT-DB port 3306 (and returning)
  tcp from NET_INTNET ports 1024:65535 to NET_INTDMZ port 22 (and returning)
```

Capture packets using the tcpdump command given for the public DMZ, substituting GIAC internal subnet filenames, as appropriate

Finally move the listener laptop to a hub placed on the subnet between the firewall interface and the border router. Use the following list as a guide to develop the tcpdump filter file that screens out packets know to be allowed on this subnet by the GIAC firewall and border router policy (named *GIACBR.filter*).

```
Allowed packets on the GIAC Border Router subnet:
  icmp from anywhere to 129.170.100.22 (SNAT firewall interface IP)
  icmp from anywhere to 129.170.100.0/28 (public DMZ and VPN subnets)
  tcp from anywhere ports 1024:65535 to GIAC-MAIL port 25,113 (and returning)
  tcp from anywhere ports 1024:65535 to GIAC-WEB port 80,443 (and returning)
  tcp from anywhere ports 1024:65535 to GIAC-DNS port 53 (and returning)
  udp from anywhere ports 1024:65535 to GIAC-DNS port 53 (and returning)
  udp from anywhere ports 1024:65535 to GIAC-VPN port 500 (and returning)
  protocol 50 from anywhere to GIAC-VPN (and returning)
  tcp from 129.170.100.22 ports 1024:65535 to anywhere port 20, 21, 22, 80, 443 (and
  returning)
  tcp from GIAC-WEB ports 1024:65535 to credit server port 443 (and returning)
  tcp from GIAC-MAIL ports 1024:65535 to anywhere port 25 (and returning)
  tcp from GIAC-DNS ports 1024:65535 to anywhere port 53 (and returning)
  udp from GIAC-DNS ports 1024:65535 to anywhere port 53 (and returning)
```

Capture packets using the tcpdump command given for the public DMZ, substituting GIAC border router subnet filenames, as appropriate

Any packets that are captured by any of the above tests indicate potential problems with the firewall enforcement of GIAC Enterprises security policy. If a packet is captured does conform to policy, review the tcpdump filter files for errors or omissions.

Conduct the audit - Scanning Phase

The scanning phase is further broken down into internal and external portions. The goal of the internal portion is to verify that the firewall is segmenting internal traffic correctly. The goal of the external portion is to verify that the firewall handles traffic inbound from an IP address outside of the GIAC network correctly.

The internal portion utilizes two laptops during each scan, one performing the scans and one listening on the target host's subnet for any packets that are passed by the firewall. Four scans (two tcp, one udp, and one protocol 50) should be run against each target service host using the following *nmap* commands:

```
nmap -sT -p 1-65335 -oN test1.out -iL DMZhosts.list
nmap -sA -p 1-65335 -oN test1.out -iL DMZhosts.list
nmap -sU -p 1-65335 -oN test1.out -iL DMZhosts.list
nmap -sO -p 50 -oN test1.out -iL DMZhosts.list

Flags:  -sT     generic TCP scan
        -sA     TCP ACK scan
        -sU     UDP scan
        -sO     protocol scan
        -p      port or range of ports. exception: Protocol # when sO is used
        -oN     send human readable output to named file
        -iL     read targets from named file
```

To reduce the time required for the audit and insure repeatability of results, scripts containing the appropriate set of nmap commands should be created in advance. Scripts

with the appropriate tcpdump command to capture packets generated by each scan should also be developed in advance. In addition, three files to be used as input should be created, each containing a space-separated list of IP addresses for a subnet with service hosts (public DMZ, VPN, and internal service network). All scripts and input files should be placed on both laptops.

To insure that all combinations are tested, place laptops according to the following Table:

Test #	Scanning Laptop (Source Subnet)	Listening Laptop (Dest. Subnet)
1	GIAC Internal	Internal Service
2	GIAC Internal	VPN Subnet
3	GIAC Internal	public DMZ
4	public DMZ	VPN Subnet
5	public DMZ	Internal Service
6	public DMZ	GIAC Internal
7	VPN Subnet	public DMZ
8	VPN Subnet	Internal Service
9	VPN Subnet	GIAC Internal

The external portion of the audit includes a scan initiated from an IP address outside the GIAC Enterprises network against each the four systems with a publicly routable IP address, GIAC-WEB, GIAC-MAIL, GIAC-DNS, and GIAC-VPN. A listening laptop is place on the target subnet to capture any traffic that is passed by the firewall. Connect the scanning laptop to the GIAC subnet between the border router and firewall (129.170.100.20/30) and assign it a non-GIAC IP address. Run the same set of four nmap scans against each server, with two modifications. Add the -P0 flag so the scan will run without ping'ing the host first and change -sT to -sS to run a SYN scan. Note that to avoid return packets being sent back out to the Internet, temporarily add a deny statement for the scanning laptop IP address to ACL 100 for the router (the one applied inbound to Ethernet interface looking towards the internal network). Use a tcpdump filter file on the listening laptop design to only capture packets with source host matching the IP address of the scanning laptop.

Review the nmap output, the syslog files collected during the time of the scans, and the packet captures to assess firewall policy enforcement performance.

Evaluate the audit

Recommendations for improvements or alternate architecture

1. Add a reverse-web proxy to allow for content based filtering of traffic to GIAC-WEB. This would add an extra layer of protection against HTML based attacks targeting the web server. This step is strongly recommended since GIAC Enterprises uses the web server as the access point for suppliers and partners, so downtime or compromise of the system would be costly.
2. Consider using RSA public/private key pairs for the VPN authentication instead of a simple pre-shared secret. If a laptop configured for VPN access is stolen, the thief will not be able to access network resources without also having the pass phrase that is protecting the RSA private key on it.

3. Tests should indicate that logging to the log server is working correctly and email alerts should be generated by swatch. However, security of the GIAC network could be enhanced by the implementation of an intrusion detection system (IDS). Such a system could alert administrators of pre-attack activities originating from source IP addresses or networks that could then be blocked from accessing the GIAC network. Some possible options for IDS configuration would be to install snort (http://snort.sourcefire.com/downloads.html) on the firewall or implement a network-based system like SHADOW (http://www.nswc.navy.mil/ISSEC/CID/) with sensors placed in the public DMZ and VPN subnets reporting to an analyst console in the internal service network.

Assignment 4 – Design Under Fire

The network design chosen for attack is taken from the practical submitted by James Manion and shown in Appendix C. In this section, the first type of attack will discuss an attack against the primary external firewall itself. For the primary firewall, the network design employs a Sparc20 running Checkpoint Firewall-1, version 4.1, Service Pack 1. Given that the date on the practical submission is October 2001, 15 months after release of CheckPoint Firewall-1, Service Pack 2 for version 4.1 [14], it will be assumed that the Firewall is still running Service Pack 1 (in early 2002). For the second type of attack, a plan to compromise information on the web server on the customer network running Microsoft IIS 4 on a Windows NT system with Service Pack 6a.

Attack against the CheckPoint Firewall-1 primary firewall

There are ten vendor-confirmed vulnerabilities for CheckPoint 2000 [15](VPN-1/Firewall-1 version 4.1). Some, but not all, of these will not be present since Service Pack 1 is applied on the target firewall. Security Focus lists six vulnerabilities that remain with SP1 applied [16]:

2001-09-12	GUI Log Viewer Vulnerability
2001-09-08	GUI Client Log Viewer Symbolic Link Vulnerability
2001-09-08	Policyname Temporary File Creation Vulnerability
2001-07-18	SecureRemote Network Information Leak Vulnerability
2001-07-11	Management Station Format String Vulnerability
2001-01-17	Denial of Service Vulnerability

All but the last one are privilege elevation exploits and require local access to the system. The last one (Denial of Service Vulnerability) [17] can be initiated remotely, but packets with spoofed internal IP addresses must arrive at the internal interface. These packets should be blocked at the border router, so this vulnerability will not be used for the planned attack:

> FW-1 Limited License DOS (CVE-2001-0182)
> This vulnerability is made possible by a problem with the way license usage is calculated by the firewall. According to the report, the number of available licenses on a Check Point Firewall can be consumed when a user

with malicious intent sends a large number of source-routed packets with fictitious or valid IP addresses to the internal interface of the firewall. When the number of licenses is exceeded, an error message is sent to the console for each offending packet. It is theoretically possible that the internal user can send enough packets to result in the firewall system becoming inaccessible from the console due to the CPU load required to generate all the error messages.

There are two additional vulnerabilities, with exploit code available, that potentially allow attackers to establish connections through the firewall to blocked TCP services: Fast Mode Vulnerability [18] and RDP Communication Vulnerability [19,20].

Fast Mode Vulnerability (CAN-2001-0082)
This vulnerability is made possible if Fast Mode has been enabled for any TCP services allowed through by the firewall. Fast Mode is intended to improve performance by only processing the initial SYN packet for a connection request to a protected service. Any non-SYN packet with a source or destination port equal to a Fast Mode enabled service will be passed unchecked. The exploit allows connections to be established to any service on a system that has at least one service allowed by the firewall rules. It accomplished by sending a carefully crafted pair of fragmented packets with the same IP id to the target system. Each packet matches the fast mode criteria, so it is passed by the firewall. However, the target system drops some of the fragments from each packet and re-assembles the remaining fragments into a different, third packet that contains a TCP SYN flag. Thus, a connection request is initiated to a service not allowed by the firewall.

Since Fast Mode is not enabled by default and the practical makes no mention of it, this exploit will likely not be successful.

RDP Communication Vulnerability
Check Point uses a proprietary RDP protocol over UDP to establish encrypted sessions. As a shortcut, only the destination port (259) and RDP command are checked before the packet is passed through the firewall. According to [20], by attaching "a faked RDP header to normal UDP traffic any content can be passed to port 259 on any remote host on either side of the firewall". Check Point confirms this vulnerability, reports that it's fixed with service pack 5 and a hot fix installed, and offers an intermediate work around that prevents the exploit by disabling functionality [19].

The required proof-of-concept exploit code can be downloaded from the Inside Security web site [21]. Note that in order to complete the attack and establish a tunnel, an internal system must be listening on UDP port 259 and configured appropriately to respond. This could be accomplished by exploiting an internal system or by sending an email

attachment for an unsuspecting GIAC employee to open. It is unclear whether such an attack would be detected by the network designed in the practical. IDS systems are deployed, but the details provided about how they are configured are scarce.

Attack against the IIS 4 Web Server internal system

The internal web servers run Internet Information Service Version 4 (IIS4) on Windows NT ServicePack6a. The list of Microsoft IIS vulnerabilities is long and growing. The attack used to attempt to compromise information on the internal system will be CVE-2001-0004 [22], BUGTRAQ ID 20010108 [23], "IIS 5.0 allows viewing files using %3F+.htr". Although the short title does not include IIS 4.0, the CVE report adds that it is also vulnerable. For this attack, the reconnaissance required simply involves using a browser to access the GIAC web site. Look for a page that is an executable script (e.g. with a .pl, .asp, or .cgi extension) that offers customer and supplier login, or credit card purchase authorization. It is likely that these scripts contain database access information (i.e. username and password) in their source code since authentication of a user requires a database connection. Once a script is located, append the string "%3F+.htr" to the URL and submit the request. If the server is vulnerable, view the source code and see what it reveals. While this exploit by itself will not result in denial of service or harm to the web server. It may reveal database access information that could be used later to gain access to customer data or GIAC proprietary sayings information.

The reconnaissance for the exploit attempt will not be blocked since it is just normal HTTP traffic allowed to the web server. The web server log files will show the actual URL used by the exploit that contains the appended characters. However, depending on the normal volume of web traffic, they may be lost in the noise of the logs.

In addition to applying the patch from Microsoft [24], one countermeasure that GIAC could employ is to deploy a reverse web proxy [25] in front of their web server that only passed URLs for the specific pages offered by the web server.

References

1. Pomeranz, Hal and Brotzman, Lee, "Running UNIX Applications Securely", *SANS GCUX Course Book6.4*, p103-137, February 2001.

2. "VPNs and Remote Access", *SANS GCFW Course Book 2.4*, p. 6 - 30, December 2001.

3. Koconis, David, "Step-By-Step Guide to Configuring an SSL Enabled Web Server that Accesses a Backend Database using RedHat 7.0", practical submitted for GCUX certification. URL: http://www.giac.org/practical/David_Koconis_GCUX.doc

4. "Client Authentication with SSL", FreeBSD Diary, July 12, 2001. URL: http://www.freebsddiary.org/openssl-client-authentication.php

5. "Firewalls 101: Perimeter Protection with Firewalls", *SANS GCFW Course Book 2.2*, p. 207, December 2001.

6. "Firewalls 102: Perimeter Protection and Defense In-Depth", *SANS GCFW Course Book 2.3*, p. 57, December 2001.

7. Russel, Rusty, "Linux 2.4 Packet Filtering HOWTO". Revsion 1.1, 07-JAN-2002. URL: http://netfilter.samba.org/documentation/HOWTO/packet-filtering-HOWTO.txt

8. Andreasson, Oskar, "iptables Tutorial 1.1.7". 2001 URL: http://www.boingworld.com/workshops/linux/iptables-tutorial/

9. Linux iptables man page. For online version see URL: http://www.linuxguruz.org/iptables/howto/maniptables.html

10. Linux Free S/WAN Project, *FreeS/WAN documentation*, URL: http://www.freeswan.org/freeswan_trees/freeswan-1.95/doc/index.html

11. SSH Communications Security Corporation, *SSH Sentinel 1.2 User Manual*, October 2001. URL: http://www.ipsec.com

12. Napier, Duncan, "Setting up a VPN Gateway". *Linux Journal*, Issue 93, January 2002. URL: http://www.linuxjournal.com/article.php?sid=4772

13. Napier, Duncan, "Administering Linux IPSec Virtual Private Networks". *Sys Admin* Volume 11, Number 3, March 2002. URL: http://www.samag.com/documents/s=4072/sam0203c/sam0203c.htm

14. securitywatch.com, “Check Point Firewall-1: Patching-time everyone” URL: http://www.securitywatch.com/newsforward/default.asp?AID=3462

15. Check Point, “Potential Security Issues in VPN-1/FireWall-1”, July 26 2000. URL: http://www.checkpoint.com/techsupport/alerts/list_vun.html

16. Security Focus Vulnerabilities by vendor, Vendor: Check Point Software, Title: Firewall-1, Version: 4.1SP1. URL: http://www.securityfocus.com/cgi-bin/vulns.pl

17. “FireWall-1 limited-IP license denial of service”, Internet Security Systems, January 17, 2001. URL: http://www.iss.net/security_center/static/5966.php

18. Lopatic, Thomas, “FireWall-1 Fastmode Vulnerability”, Posted to bugtraq 2000-12-18 06:04:04 CST, archive URL: http://archives.neohapsis.com/archives/bugtraq/2000-12/0271.html

19. Check Point, “RDP Communication Vulnerability”, July 12, 2001. URL: http://www.checkpoint.com/techsupport/alerts/rdp.html

20. Inside Security, “Check Point FireWall-1 RDP Bypass Vulnerability”, July 14, 2001. URL: http://www.inside-security.de/fw1_rdp.html

21. Inside Security, “Check Point FireWall-1 RDP Bypass Vulnerability Proof of Concept Code”, July 14, 2001. URL: http://www.inside-security.de/fw1_rdp_poc.html

22. ICAT database entry. URL: http://icat.nist.gov/icat.cfm?cvename=CVE-2001-0004

23. Guninski, Georgi, “IIS 5.0 allows viewing files using %3F+.htr”, Posted to bugtraq 2001-01-08 15:00:30, archive URL: http://marc.theaimsgroup.com/?l=bugtraq&m=97897954625305&w=2

24. “Malformed .HTR Request Allows Reading of File Fragments”, *Microsoft Security Bulletin MS01-004*, January 29, 2001. URL: http://www.microsoft.com/technet/security/bulletin/MS01-004.asp

25. “Jeanne: Reverse Proxy Server”, Institute for Security Technology Studies, 2001. URL: http://www.ists.dartmouth.edu/IRIA/projects/jeanne.htm

Appendix A - Complete rule-set for external firewall

```
[root@GIAC-FW /root] /sbin/iptables -L -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
  pkts bytes  target     prot  opt   in    out       source         destination
    0     0    ACCEPT     all   --   lo     *       127.0.0.1        127.0.0.1
    0     0    ACCEPT     udp   --  eth3    *        10.1.2.5         10.1.2.1       udp dpt:123
    0     0  HANDLE_ICMP  icmp  --    *     *       0.0.0.0/0        0.0.0.0/0
    0     0   WEED_OUT    all   --    *     *       0.0.0.0/0        0.0.0.0/0
    0     0      LOG      all   --    *     *       0.0.0.0/0        0.0.0.0/0 LOG flags 0 level 5 prefix `END INPUT CHAIN:'
    0     0    REJECT     all   --    *     *       0.0.0.0/0        0.0.0.0/0 reject-with icmp-port-unreachable

Chain FORWARD (policy DROP 0 packets, 0 bytes)
  pkts bytes  target     prot  opt   in    out   source           destination
    0     0  HANDLE_ICMP  icmp  --    *     *    0.0.0.0/0         0.0.0.0/0
    0     0    TO_DMZ     all   --  eth0    *    !10.1.0.0/23   129.170.100.8/29
    0     0    TO_DMZ     all   --  eth4    *     10.1.1.0/24   129.170.100.8/29
    0     0    TO_DMZ     all   --  eth3    *     10.1.2.0/28   129.170.100.8/29
    0     0    TO_DMZ     all   --  eth2    *   129.170.100.0/29 129.170.100.8/29
    0     0    TO_VPN     all   --  eth0    *     !10.1.0.0/23   129.170.100.0/29
    0     0    TO_VPN     all   --  eth4    *     10.1.1.0/24    129.170.100.0/29
    0     0   TO_INTDMZ   all   --  eth0    *     !10.1.0.0/23     10.1.2.0/28
    0     0   TO_INTDMZ   all   --  eth1    *   129.170.100.8/29   10.1.2.0/28
    0     0   TO_INTDMZ   all   --  eth4    *     10.1.1.0/24      10.1.2.0/28
    0     0   TO_INTDMZ   all   --  eth2    *      0.0.0.0/0       10.1.2.0/28
    0     0     TO_BR     all   --  eth1  eth0 129.170.100.8/29     0.0.0.0/0
    0     0     TO_BR     all   --  eth2  eth0 129.170.100.0/29     0.0.0.0/0
    0     0     TO_BR     all   --  eth4  eth0   10.1.1.0/24        0.0.0.0/0
    0     0   WEED_OUT    all   --    *     *      0.0.0.0/0        0.0.0.0/0
    0     0      LOG      all   --    *     *      0.0.0.0/0        0.0.0.0/0 LOG flags 0 level 5 prefix `END FORWARD CHAIN:'
    0     0    REJECT     all   --    *     *      0.0.0.0/0        0.0.0.0/0 reject-with icmp-port-unreachable

Chain OUTPUT (policy DROP 0 packets, 0 bytes)
  pkts bytes  target     prot  opt   in    out       source         destination
    0     0    ACCEPT     all   --    *     *       127.0.0.1        127.0.0.1
    0     0    ACCEPT     udp   --    *   eth3      127.0.0.1        10.1.2.5  udp dpt:514
    0     0  HANDLE_ICMP  icmp  --    *     *       0.0.0.0/0        0.0.0.0/0
    0     0   WEED_OUT    all   --    *     *       0.0.0.0/0        0.0.0.0/0
    0     0      LOG      all   --    *     *       0.0.0.0/0        0.0.0.0/0 LOG flags 0 level 5 prefix `END OUTPUT CHAIN:'
    0     0    REJECT     all   --    *     *       0.0.0.0/0        0.0.0.0/0 reject-with icmp-port-unreachable
```

```
Chain HANDLE_ICMP (3 references)
 pkts bytes  target     prot  opt   in    out    source          destination
   0     0    ACCEPT     icmp  --    *     *       127.0.0.1       0.0.0.0/0 icmp type 8 state NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     icmp  --    *     *      10.1.1.0/24      0.0.0.0/0 icmp type 8 state NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     icmp  --    *     *   129.170.100.8/29    0.0.0.0/0 icmp type 8 state NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     icmp  --    *     *   129.170.100.0/29    0.0.0.0/0 icmp type 8 state NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     icmp  --    *   !eth0   10.1.2.0/28       0.0.0.0/0 icmp type 8 state NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     icmp  --  eth0  *       0.0.0.0/0     129.170.100.8/29   icmp type 8 state NEW,RELATED,ESTABLISHED
   0     0      LOG      all   --    *     *       0.0.0.0/0       0.0.0.0/0 LOG flags 0 level 5 prefix `END HANDLE_ICMP:'
   0     0    REJECT     all   --    *     *       0.0.0.0/0       0.0.0.0/0 reject-with icmp-port-unreachable

Chain TO_BR (3 references)
 pkts bytes  target     prot  opt   in    out    source          destination
   0     0    ACCEPT     tcp   --    *     *   129.170.100.10   216.136.147.22  tcp dpt:443 state NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     tcp   --    *     *   129.170.100.10   216.136.147.18  tcp dpt:443 state NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     tcp   --    *     *   129.170.100.11      0.0.0.0/0 tcp dpt:25 state NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     udp   --    *     *   129.170.100.12      0.0.0.0/0 udp dpt:53 state NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     tcp   --    *     *   129.170.100.12      0.0.0.0/0 tcp dpt:53 state NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     tcp   --    *     *     10.1.1.0/24       0.0.0.0/0 tcp dpt:22 state NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     tcp   --    *     *     10.1.1.0/24       0.0.0.0/0 tcp dpt:80 state NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     tcp   --    *     *     10.1.1.0/24       0.0.0.0/0 tcp dpt:443 state NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     tcp   --    *     *     10.1.1.0/24       0.0.0.0/0 tcp dpt:21 state NEW,RELATED,ESTABLISHED
   0     0   WEED_OUT    all   --    *     *       0.0.0.0/0       0.0.0.0/0
   0     0      LOG      all   --    *     *       0.0.0.0/0       0.0.0.0/0 LOG flags 0 level 5   prefix `END TO_BR CHAIN:'
   0     0    REJECT     all   --    *     *       0.0.0.0/0       0.0.0.0/0 reject-with icmp-port-unreachable

Chain TO_DMZ   (4   references)
 pkts bytes  target     prot  opt   in    out      source         destination
   0     0    ACCEPT     tcp   --    *     *       0.0.0.0/0     129.170.100.10  tcp dpt:80       state   NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     tcp   --    *     *       0.0.0.0/0     129.170.100.10  tcp dpt:443      state   NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     tcp   --    *     *       0.0.0.0/0     129.170.100.11  tcp dpt:25       state   NEW,RELATED,ESTABLISHED
   0     0    REJECT     tcp   --    *     *       0.0.0.0/0       0.0.0.0/0     tcp dpt:113      reject-with    tcp-reset
   0     0    ACCEPT     udp   --    *     *       0.0.0.0/0     129.170.100.12  udp dpt:53       state   NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     tcp   --    *     *       0.0.0.0/0     129.170.100.12  tcp dpt:53       state   NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     udp   --    *     *        10.1.1.5       0.0.0.0/0     udp dpt:123      state   NEW,RELATED,ESTABLISHED
   0     0    ACCEPT     tcp   --    *     *      10.1.1.0/24      0.0.0.0/0     tcp dpt:22       state   NEW,RELATED,ESTABLISHED
   0     0   WEED_OUT    all   --    *     *       0.0.0.0/0       0.0.0.0/0
   0     0      LOG      all   --    *     *       0.0.0.0/0       0.0.0.0/0     LOG flags 0 level 5      prefix `END TO_DMZ     CHAIN:'
   0     0    REJECT     all   --    *     *       0.0.0.0/0       0.0.0.0/0     reject-with icmp-port-unreachable
```

```
ChainTO_INTDMZ  (4   references)
  pkts bytes  target     prot  opt   in    out       source         destination
    0     0    ACCEPT     tcp   --    *     *    129.170.100.11     10.1.2.2  tcp dpt:25 state     NEW,RELATED,ESTABLISHED
    0     0    ACCEPT     tcp   --    *     *     10.1.1.0/24       10.1.2.2  tcp dpt:25 state     NEW,RELATED,ESTABLISHED
    0     0    ACCEPT     tcp   --    *     *     10.1.1.0/24       10.1.2.2  tcp dpt:110 state    NEW,RELATED,ESTABLISHED
    0     0    ACCEPT     tcp   --    *     *     10.1.1.0/24       10.1.2.2  tcp dpt:143 state    NEW,RELATED,ESTABLISHED
    0     0    ACCEPT     tcp   --   eth2   *      0.0.0.0/0        10.1.2.2  tcp dpt:25 state     NEW,RELATED,ESTABLISHED
    0     0    ACCEPT     tcp   --    *     *      0.0.0.0/0        10.1.2.2  tcp dpt:110 state    NEW,RELATED,ESTABLISHED
    0     0    ACCEPT     tcp   --   eth2   *      0.0.0.0/0        10.1.2.2  tcp dpt:143 state    NEW,RELATED,ESTABLISHED
    0     0    ACCEPT     tcp   --    *     *     10.1.1.0/24       10.1.2.4  tcp dpt:3306 state   NEW,RELATED,ESTABLISHED
    0     0    ACCEPT     udp   --    *     *      0.0.0.0/0        10.1.2.5  udp dpt:514 state    NEW,RELATED,ESTABLISHED
    0     0    ACCEPT     tcp   --    *     *     10.1.1.0/24       0.0.0.0/0 tcp dpt:22 state     NEW,RELATED,ESTABLISHED
    0     0    ACCEPT     tcp   --    *     *    129.170.100.10     10.1.2.4  tcp dpt:22 state     NEW,RELATED,ESTABLISHED
    0     0   WEED_OUT    all   --    *     *      0.0.0.0/0        0.0.0.0/0
    0     0      LOG      all   --    *     *      0.0.0.0/0        0.0.0.0/0 LOG flags 0 level    5      prefix  `END    TO_INTDMZ      CHAIN:'
    0     0    REJECT     all   --    *     *      0.0.0.0/0        0.0.0.0/0 reject-with icmp-port-unreachable

Chain  TO_VPN    (2   references)
  pkts bytes  target     prot  opt   in    out       source         destination
    0     0    ACCEPT  ipv6-crypt  --   *     *      0.0.0.0/0        0.0.0.0/0 state NEW,RELATED,ESTABLISHED
    0     0    ACCEPT     udp   --    *     *      0.0.0.0/0        0.0.0.0/0 udp dpt:500 state    NEW,RELATED,ESTABLISHED
    0     0    ACCEPT     udp   --    *     *       10.1.1.5        0.0.0.0/0 udp dpt:123  state   NEW,RELATED,ESTABLISHED
    0     0    ACCEPT     tcp   --    *     *     10.1.1.0/24       0.0.0.0/0 tcp dpt:22 state     NEW,RELATED,ESTABLISHED
    0     0   WEED_OUT    all   --    *     *      0.0.0.0/0        0.0.0.0/0
    0     0      LOG      all   --    *     *      0.0.0.0/0        0.0.0.0/0 LOG flags 0 level    5      prefix  `END    TO_VPN  CHAIN:'
    0     0    REJECT     all   --    *     *      0.0.0.0/0        0.0.0.0/0 reject-with icmp-port-unreachable

Chain WEED_OUT   (7   references)
  pkts bytes  target     prot  opt   in    out       source         destination
    0     0     DROP      tcp   --   eth0   *      0.0.0.0/0        0.0.0.0/0     tcp dpts:137:139
    0     0     DROP      tcp   --   eth0   *      0.0.0.0/0      !129.170.100.10  tcp dpt:80
    0     0    RETURN     all   --    *     *      0.0.0.0/0        0.0.0.0/0

[root@GIAC-FW /root] /sbin/iptables -L -n -v -t nat
Chain PREROUTING (policy ACCEPT 16591 packets, 2993K bytes)
  pkts bytes  target     prot  opt   in    out       source         destination
    0     0     DNAT      udp   --   eth0   *      0.0.0.0/0      129.170.100.22    udp dpt:514 to:10.1.2.5:514

Chain POSTROUTING (policy ACCEPT 3306 packets, 246K bytes)
  pkts bytes  target     prot  opt   in    out       source         destination
    0     0     SNAT      all   --    *    eth0    10.1.1.0/24       0.0.0.0/0 to:129.170.100.22
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Appendix B - IPTABLES firewall script

Script to load iptables firewall rule-set for GIAC-FW, complete with copious comments embedded.

```
#! /bin/bash
#
# Description: Loads iptables firewall rule-set for GIAC-FW.giac
#
# Uncomment the next line and all commands will be
# repeated to the shell (very useful for debugging).
# set -x
#
################################################
# In this first section, define variables
################################################
# The full path to the iptables executable.
IPTABLES="/sbin/iptables"

##### INTERFACES
# The loopback interface and address
IF_LO="lo"
IP_LO="127.0.0.1"
# The interface and IP between the FW and router
IF_BR="eth0"
IP_BR="129.170.100.22"
# The interface and IP between the FW and public DMZ
IF_DMZ="eth1"
IP_DMZ="129.170.100.9"
# The interface and IP between the FW and VPN gateway
IF_VPN="eth2"
IP_VPN="129.170.100.1"
# The interface and IP between the FW and Internal service net
IF_INTDMZ="eth3"
IP_INTDMZ="10.1.2.1"
# The interface and IP between the FW and Internal service net
IF_INTNET="eth4"
IP_INTNET="10.1.1.1"

##### NETWORKS
# public DMZ
NET_DMZ="129.170.100.8/29"
# VPN network
NET_VPN="129.170.100.0/29"
# Internal Service Network
NET_INTDMZ="10.1.2.0/28"
# Internal Employees Network
NET_INTNET="10.1.1.0/24"
# GIAC Enterprises Network
NET_GIAC="10.1.1.0/23"

##### GIAC SYSTEMS
GIAC_WEB="129.170.100.10/32"
GIAC_MAIL="129.170.100.11/32"
GIAC_DNS="129.170.100.12/32"
GIAC_VPN="129.170.100.2/32"
```

```
INT_DB="10.1.2.4/32"
INT_MAIL="10.1.2.2/32"
INT_DNS="10.1.2.3/32"
INT_LOG="10.1.2.5/32"
IP_INT_LOG="10.1.2.5"

##### EXTERNAL SYSTEMS
IP_CREDITCARD1="216.136.147.22/32"  # xml.test.surepay.com
IP_CREDITCARD2="216.136.147.18/32"  # xml.surepay.com

#################################################
# End variable definition
#################################################

#################################################
# Get the environment ready.  We want to start with a known baseline:
# no rules defined, no user-defined chains, and default policies
# that DENY everything.
#################################################

# First flush all the rules.  This should be done prior to
# attempting to delete user-defined chains because a user-defined
# chain will not be deleted if a rule exists that references it.
$IPTABLES -F
$IPTABLES -F -t nat

# Delete all user-defined chains
$IPTABLES -X

# Set default policy for built-in chains
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP

# Create user-defined chains
$IPTABLES -N TO_BR
$IPTABLES -N TO_DMZ
$IPTABLES -N TO_VPN
$IPTABLES -N TO_INTDMZ
$IPTABLES -N HANDLE_ICMP
$IPTABLES -N WEED_OUT

# Set log level
LOGFLAG="LOG --log-level notice"

#################################################
# End environment set-up
#################################################
#################################################
# THE WEED_OUT CHAIN
# Packets entering this chain have not matched
# any rule yet and would get logged if not dropped.
# This is used to get rid of messages that clutter
# up logs.
#################################################

# NetBIOS traffic from the Internet
```

```
$IPTABLES -A WEED_OUT -i $IF_BR -p tcp --dport 137:139 -j DROP

# HTTP requests from border router not destined for GIAC_WEB
$IPTABLES -A WEED_OUT -i $IF_BR -p tcp --dport 80 -d ! $GIAC_WEB -j
DROP

# RETURN to the chain that sent the packet here
$IPTABLES -A WEED_OUT -j RETURN

################################################
# THE HANDLE_ICMP CHAIN
# Packets entering this chain match ICMP protocol
################################################

# Ping allowed from loopback to anywhere
$IPTABLES -A HANDLE_ICMP -p icmp --icmp-type echo-request -s $IP_LO -m
state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# Ping allowed from internal systems to anywhere
$IPTABLES -A HANDLE_ICMP -p icmp --icmp-type echo-request -s
$NET_INTNET -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# Ping allowed from public DMZ to anywhere
$IPTABLES -A HANDLE_ICMP -p icmp --icmp-type echo-request -s $NET_DMZ -
m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# Ping allowed from VPN gateway to anywhere
$IPTABLES -A HANDLE_ICMP -p icmp --icmp-type echo-request -s $NET_VPN -
m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# Ping allowed from internal DMZ to anywhere except Border Router IF
$IPTABLES -A HANDLE_ICMP -o ! $IF_BR -p icmp --icmp-type echo-request -
s $NET_INTDMZ -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# Ping allowed from Border Router IF to DMZ servers so people on
# the Internet can check if our public servers are up.
$IPTABLES -A HANDLE_ICMP -i $IF_BR -p icmp --icmp-type echo-request -d
$NET_DMZ -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# LOG and REJECT everything else
$IPTABLES -A HANDLE_ICMP -j $LOGFLAG --log-prefix "END HANDLE_ICMP:"
$IPTABLES -A HANDLE_ICMP -j REJECT

################################################
# THE TO_DMZ CHAIN
# Packets entering this chain have destination IP
# of $NET_DMZ and could come from anywhere.
################################################

# HTTP/HTTPS traffic is allowed to the web server
$IPTABLES -A TO_DMZ -p tcp --dport 80 -d $GIAC_WEB -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A TO_DMZ -p tcp --dport 443 -d $GIAC_WEB -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# SMTP traffic is allowed to the mail server
```

```
$IPTABLES -A TO_DMZ -p tcp --dport 25 -d $GIAC_MAIL -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# Send connection resets to ident requests to avoid delays
$IPTABLES -A TO_DMZ -p tcp --dport 113 -j REJECT --reject-with tcp-
reset

# DNS traffic is allowed to dns server (UDP and TCP inbound)
$IPTABLES -A TO_DMZ -p udp --dport 53 -d $GIAC_DNS -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A TO_DMZ -p tcp --dport 53 -d $GIAC_DNS -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# NTP from INT-LOG
$IPTABLES -A TO_DMZ -p udp --dport 123 -s $INT_LOG -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# SSH from GIAC internal network to all systems
$IPTABLES -A TO_DMZ -p tcp --dport 22 -s $NET_INTNET -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# WEED OUT, LOG and REJECT everything else
$IPTABLES -A TO_DMZ -j WEED_OUT
$IPTABLES -A TO_DMZ -j $LOGFLAG --log-prefix "END TO_DMZ CHAIN:"
$IPTABLES -A TO_DMZ -j REJECT

################################################
# THE TO_VPN CHAIN
# Packets entering this chain have destination IP
# of $GIAC_VPN and could come from anywhere.
################################################

# IPSEC traffic
$IPTABLES -A TO_VPN -p 50 -m state --state NEW,ESTABLISHED,RELATED -j
ACCEPT
$IPTABLES -A TO_VPN -p udp --dport 500 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# NTP from INT-LOG
$IPTABLES -A TO_VPN -p udp --dport 123 -s $INT_LOG -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# SSH from GIAC internal network for administration
$IPTABLES -A TO_VPN -p tcp --dport 22 -s $NET_INTNET -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# WEED OUT, LOG and REJECT everything else
$IPTABLES -A TO_VPN -j WEED_OUT
$IPTABLES -A TO_VPN -j $LOGFLAG --log-prefix "END TO_VPN CHAIN:"
$IPTABLES -A TO_VPN -j REJECT

################################################
# THE TO_INTDMZ CHAIN
# Packets entering this chain have destination IP of
# $NET_INTDMZ and could be from anywhere except $IF_BR.
################################################
```

```
# SMTP traffic to INT-MAIL from GIAC-MAIL
$IPTABLES -A TO_INTDMZ -p tcp --dport 25 -s $GIAC_MAIL -d $INT_MAIL -m
state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# SMTP, POP3 and IMAP traffic to INT-MAIL from internal network
$IPTABLES -A TO_INTDMZ -p tcp --dport 25 -s $NET_INTNET -d $INT_MAIL -m
state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A TO_INTDMZ -p tcp --dport 110 -s $NET_INTNET -d $INT_MAIL -
m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A TO_INTDMZ -p tcp --dport 143 -s $NET_INTNET -d $INT_MAIL -
m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# SMTP, POP3 and IMAP traffic to INT-MAIL from VPN gateway interface
$IPTABLES -A TO_INTDMZ -i $IF_VPN -p tcp --dport 25 -d $INT_MAIL -m
state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A TO_INTDMZ -p tcp --dport 110 -d $INT_MAIL -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A TO_INTDMZ -i $IF_VPN -p tcp --dport 143 -d $INT_MAIL -m
state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# MySQL traffic to INT-DB from internal network
$IPTABLES -A TO_INTDMZ -p tcp --dport 3306 -s $NET_INTNET -d $INT_DB -m
state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# syslog traffic from anywhere to INT_LOG
$IPTABLES -A TO_INTDMZ -p udp --dport 514 -d $INT_LOG -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# SSH from GIAC internal network for administration
$IPTABLES -A TO_INTDMZ -p tcp --dport 22 -s $NET_INTNET -m state --
state NEW,ESTABLISHED,RELATED -j ACCEPT

# SSH initiated from GIAC_WEB to INT-DB for port forwarding
$IPTABLES -A TO_INTDMZ -p tcp --dport 22 -s $GIAC_WEB -d $INT_DB -m
state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# WEED OUT, LOG and REJECT everything else
$IPTABLES -A TO_INTDMZ -j WEED_OUT
$IPTABLES -A TO_INTDMZ -j $LOGFLAG --log-prefix "END TO_INTDMZ CHAIN:"
$IPTABLES -A TO_INTDMZ -j REJECT

################################################
# THE TO_BR CHAIN
# Packets entering this chain have destination outside
# GIAC and could be from DMZ, VPN or INTNET
################################################

# HTTPS traffic can be initiated from web server to credit servers
$IPTABLES -A TO_BR -p tcp --dport 443 -s $GIAC_WEB -d $IP_CREDITCARD1 -
m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A TO_BR -p tcp --dport 443 -s $GIAC_WEB -d $IP_CREDITCARD2 -
m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# SMTP traffic to third party mail servers from GIAC_MAIL
$IPTABLES -A TO_BR -p tcp --dport 25 -s $GIAC_MAIL -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
```

```
# DNS traffic can be initiated to third party DNS servers (UDP and TCP)
$IPTABLES -A TO_BR -p udp --dport 53 -s $GIAC_DNS -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A TO_BR -p tcp --dport 53 -s $GIAC_DNS -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# SSH from GIAC internal network
$IPTABLES -A TO_BR -p tcp --dport 22 -s $NET_INTNET -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# HTTP/HTTPS from GIAC internal network
$IPTABLES -A TO_BR -p tcp --dport 80 -s $NET_INTNET -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A TO_BR -p tcp --dport 443 -s $NET_INTNET -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# FTP from GIAC internal network
$IPTABLES -A TO_BR -p tcp --dport 21 -s $NET_INTNET -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# WEED OUT, LOG and REJECT everything else
$IPTABLES -A TO_BR -j WEED_OUT
$IPTABLES -A TO_BR -j $LOGFLAG --log-prefix "END TO_BR CHAIN:"
$IPTABLES -A TO_BR -j REJECT

################################################
# THE INPUT CHAIN
# Only packets destined for the firewall box
# itself should enter this chain.
################################################

# Allow traffic to self
$IPTABLES -A INPUT -i $IF_LO -s $IP_LO -d $IP_LO -j ACCEPT

# Allow NTP traffic from INT-LOG
$IPTABLES -A INPUT -i $IF_INTDMZ -p udp --dport 123 -s $INT_LOG -d
$IP_INTDMZ -j ACCEPT

# Handle ICMP traffic
$IPTABLES -A INPUT -p icmp -j HANDLE_ICMP

# Before logging, weed out packets we know we can drop
# and don't care to log.  Log and reject everything else
$IPTABLES -A INPUT -j WEED_OUT
$IPTABLES -A INPUT -j $LOGFLAG --log-prefix "END INPUT CHAIN:"
$IPTABLES -A INPUT -j REJECT

################################################
# THE OUTPUT CHAIN
# Only packets originating on the firewall box
# itself should enter this chain.
################################################

# Allow traffic to self
$IPTABLES -A OUTPUT -s $IP_LO -d $IP_LO -j ACCEPT

# Allow syslog messages to INT-LOG
```

```
$IPTABLES -A OUTPUT -o $IF_INTDMZ -p udp --dport 514 -s $IP_LO -d
$INT_LOG -j ACCEPT

# Handle ICMP traffic
$IPTABLES -A OUTPUT -p icmp -j HANDLE_ICMP

# Weed out, log, and reject everything else
$IPTABLES -A OUTPUT -j WEED_OUT
$IPTABLES -A OUTPUT -j $LOGFLAG --log-prefix "END OUTPUT CHAIN:"
$IPTABLES -A OUTPUT -j REJECT

################################################
# THE FORWARD CHAIN
# This is where the action is since most traffic
# is only passing through the firewall.  To make
# management of rules simpler, we send packets to
# user-defined chains for further processing.
################################################

#### REJECT non-routable IP addresses from the border router
for NET in $IP_PRIVATE; do
  $IPTABLES -A FORWARD -i $IF_BR -s $NET -j DROP
done

#### Handle ICMP traffic (since we allow related most ICMP
# that is appropriate should be handled by state table)
$IPTABLES -A FORWARD -p icmp -j HANDLE_ICMP

#### Traffic to Public DMZ goes to the TO_DMZ chain
# Coming from border router accept any source IP not GIAC private
$IPTABLES -A FORWARD -I $IF_BR -s ! $NET_GIAC -d $NET_DMZ -j TO_DMZ
# Coming from internal network with source IP internal network
$IPTABLES -A FORWARD -I $IF_INTNET -s $NET_INTNET -d $NET_DMZ -j TO_DMZ
# Coming from int service network with source IP int service network
$IPTABLES -A FORWARD -i $IF_INTDMZ -s $NET_INTDMZ -d $NET_DMZ -j TO_DMZ
# Coming from VPN with source IP VPN
$IPTABLES -A FORWARD -i $IF_VPN -s $NET_VPN -d $NET_DMZ -j TO_DMZ

#### Traffic to VPN goes to the TO_VPN chain
# Coming from border router accept any source IP not GIAC private
$IPTABLES -A FORWARD -i $IF_BR -s ! $NET_GIAC -d $NET_VPN -j TO_VPN
# Coming from internal network with source IP internal network
$IPTABLES -A FORWARD -i $IF_INTNET -s $NET_INTNET -d $NET_VPN -j TO_VPN

#### Traffic to Internal Service Net goes to the TO_INTDMZ chain
# Coming from border router accept any source IP not GIAC private
$IPTABLES -A FORWARD -i $IF_BR -s ! $NET_GIAC -d $NET_INTDMZ -j
TO_INTDMZ
# Coming from DMZ accept DMZ source IP
$IPTABLES -A FORWARD -i $IF_DMZ -s $NET_DMZ -d $NET_INTDMZ -j TO_INTDMZ
# Coming from internal network with source IP internal network
$IPTABLES -A FORWARD -i $IF_INTNET -s $NET_INTNET -d $NET_INTDMZ -j
TO_INTDMZ
# Coming from VPN with any source IP
$IPTABLES -A FORWARD -i $IF_VPN -d $NET_INTDMZ -j TO_INTDMZ

#### Traffic to border router goes to the TO_BR chain
```

```
# Coming from DMZ interface with source IP of DMZ network
$IPTABLES -A FORWARD -i $IF_DMZ -o $IF_BR -s $NET_DMZ -j TO_BR
# Coming from DMZ interface with source IP of VPN
$IPTABLES -A FORWARD -i $IF_VPN -o $IF_BR -s $NET_VPN -j TO_BR
# Coming from internal network interface, source IP of internal network
$IPTABLES -A FORWARD -i $IF_INTNET -o $IF_BR -s $NET_INTNET -j TO_BR

# Weed out, log, and reject everything else
$IPTABLES -A FORWARD -j WEED_OUT
$IPTABLES -A FORWARD -j $LOGFLAG --log-prefix "END FORWARD CHAIN:"
$IPTABLES -A FORWARD -j REJECT

################################################
# NETWORK ADDRESS TRANSLATION
# Use hiding NAT for packets connecting to the
# Internet from the GIAC internal network
################################################

# Flush the NAT table
$IPTABLES -F -t nat

# Use hiding NAT for traffic from internal to outside world
$IPTABLES -t nat -A POSTROUTING -o $IF_BR -s $NET_INTNET -j SNAT --to-
source $IP_BR

# Use static NAT so router can log to syslog
$IPTABLES -t nat -A PREROUTING -i $IF_BR -p udp --dport 514 -d $IP_BR -
j DNAT --to-destination $IP_INT_LOG:514
```

Appendix C - Network Design by James Manion
(http:/www.giac.org/practical/James_Manion_GCFW.doc)