GIAC Certified Firewall Analyst Practical Research Paper

By

Michael Reiter

GCFW Practical Version 1.6

Section 1: GIAC Enterprises Security Architecture

The core of the architecture is the T. Rex application proxy firewall. The specific configuration is the hardware appliance model Trex-ES x34-1 with 3 NICs, a rack-mount model running a hardened version of Linux and T. Rex version 2 licensed for unlimited concurrent sessions. An application proxy was chosen not for its speed but for its additional security derived from the capability to drill down into the application level data. Speed in this case is secondary to security; the B2B business model is slightly more tolerant of a minor delay and the number of users is expected to be small in comparison to a successful B2C site. Inbound connections initiated from the internet are directed to the screened subnet; only proxies for DNS, secure web traffic and inbound mail (Lotus Notes) are permitted. No connections are permitted from the screened subnet to the interior network.. Users must configure their mail clients to poll the mail server to get mail. All connections from the users to the internet are permitted through the firewall.

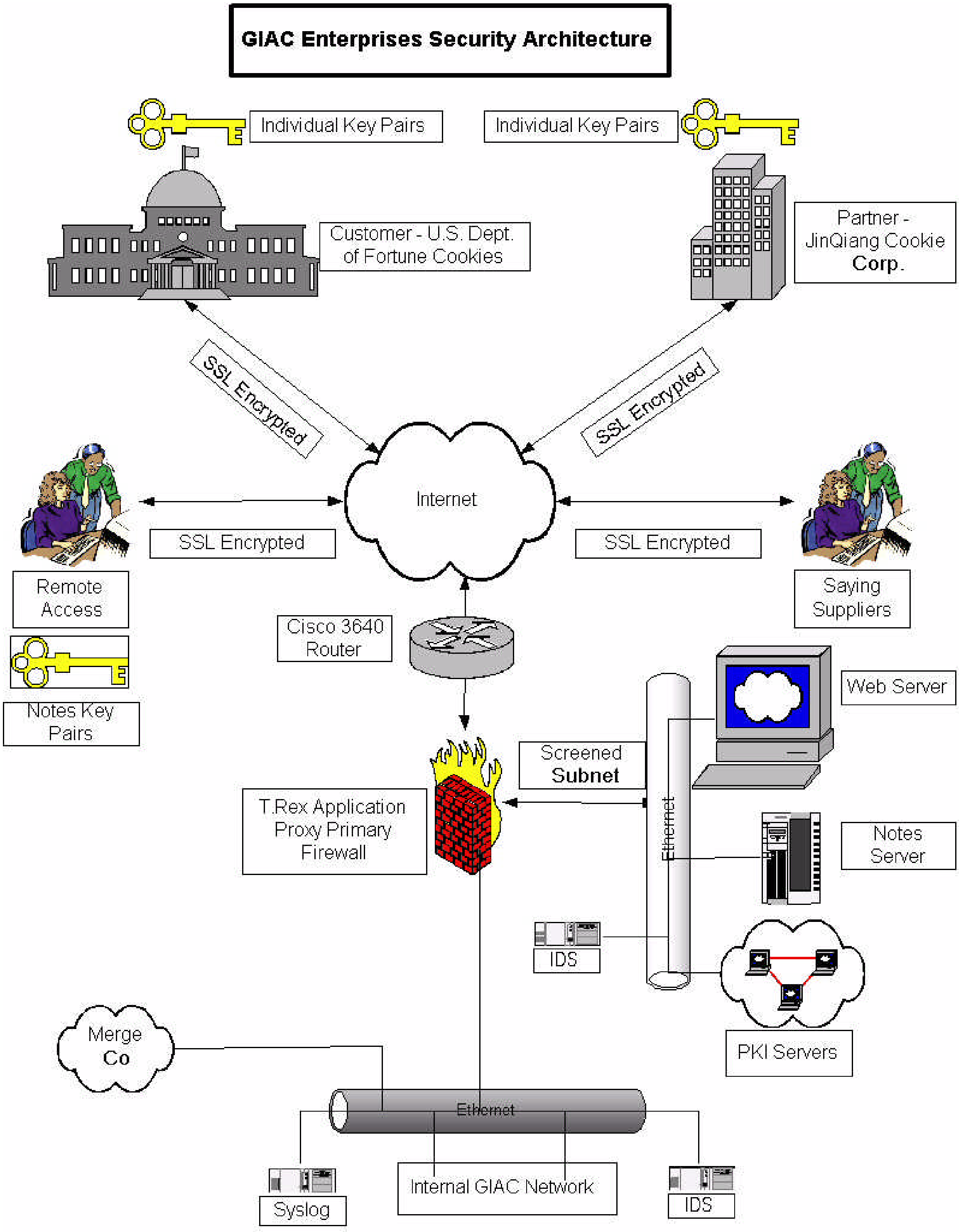
The router is a Cisco 3640 running IOS 12.1. It has a generic ACL designed to prevent spoofing and various other common network based attacks, but otherwise merely moves packets between the GIAC network and the internet.

The interior network contains the original GIAC net plus that of the acquired firm, MergeCo. MergeCo's web site has been made into subdomain of the GIAC site, and is hosted by an ISP. MergeCo users were assigned new IP addresses and have been moved in to the GIAC offices. Users control trojans sending from their machines through the use of the commercial version of Zone Alarm, which applies a cryptographic checksum to executables on the hard drive to prevent trojans from masquerading as legitimate executables, and also alerts users to traffic originating from their machines. A Snort network Intrusion Detection Sensor (IDS) sits on a SPAN port on the network switch.

The screened subnet contains the web server, the Notes server, and servers supporting the Public Key Infrastructure. The web server listens only on port 443, SSL. Unencrypted port 80 traffic is blocked at the firewall and the http daemon does not listen for it. The web server – a Microsoft Windows 2000 Server running IIS 5.0 with all patches – is running separate but linked virtual sites. One site is the general public site, www.giacsays.com; this is purely for advertising. Another site is the suppliers' site where new sayings are uploaded, www.selltogiac.com. This site has a simple form CGI that allows anyone to upload a saying, along with their contact information. If GIAC likes the saying, a check is cut on a per-saying basis; there are no contractual relationships with the suppliers and all inputs to the CGI are filtered to allow in only legal characters and prevent system or database commands from going through. Customers come in through the www.giacsales.com site. This is a straightforward web-based catalogue of bulk sayings packages; new B2B customers must be pre-approved and verified by the sales department. GIAC is not doing B2C, as the marginal profit from sales of individual sayings is miniscule. Verification is via personal contact and a credit check. Once verification of the customer's company and designated buyer is complete, they are given instructions on how to generate a public/private key pair and obtain a GIAC-signed digital certificate from the site. Applications for digital certificates are approved manually after a phone confirmation with the buyer's supervisor to enhance security and prevent spoofing. Partners also come in to www.giacsales.com; however, their digital certificates are issued to identify them as partners instead of customers, and they are given discounted prices. A Snort network Intrusion Detection Sensor (IDS) sits on a SPAN port on the network switch.

Remote users create digital certificates before leaving the office. The Lotus Notes Server enables authorized users from the internet to access mail and selected databases while on the road. Traffic between the remote users and the Notes server is SSL encrypted and PKI authenticated.

One may legitimately pose the question, "Why not put partners and customers on the Notes VPN"? This could certainly be done; however, this would entail building a Notes network at each site. Above and beyond the cost, there is also the issue of trust. Extending the network to another site means extending trust to that site. We don't trust organizations outside our own; we prefer to confer limited trust on a designated representative of that organization.


GIAC Enterprises Security Architecture
Individual Key Pairs
Individual Key Pairs
Customer - U.S. Dept. of Fortune Cookies
Partner - JinQiang Cookie Corp.
SSL Encrypted
SSL Encrypted
Internet
SSL Encrypted
SSL Encrypted
Remote Access
Saying Suppliers
Notes Key Pairs
Cisco 3640 Router
Web Server
Screened Subnet
Ethernet
T.Rex Application Proxy Primary Firewall
Notes Server
IDS
PKI Servers
Merge Co
Ethernet
Syslog
Internal GIAC Network
IDS

Section 2: Security Policy

Border Router

Border Router Exterior Interface: 98.76.54.32
Border Router Interior Interface: 12.34.56.79

The GIAC border router is intended only to screen incoming/outgoing traffic for obvious bad actors; it is a first line of defense, but we do not expect it to protect us from a sophisticated attack. Because we are not Cisco certified engineers, we have used a modified version of the generic secure ACL found at http://www.pasadena.net/cisco/secure.html as a base for our ACL. This was written by Frank Keeney of Pasadena Networks, based in turn on security suggestions found at http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_c/scoverv.htm. Comments within the ACL explain the purpose of each rule.

```
! Beginning of access-list 101
!
! Deny rfc 1918 addresses (address allocation for private internets):
!Any packet with these source addresses are bound to be bad actors
access-list 101 deny ip 192.168.0.0 0.0.255.255 any log
access-list 101 deny ip 172.16.0.0 0.15.255.255 any log
access-list 101 deny ip 10.0.0.0 0.255.255.255 any log
!
! Deny packets with localhost, broadcast and multicast addresses:
!These would be spoofed packets or junk
access-list 101 deny ip 127.0.0.0 0.255.255.255 any log
access-list 101 deny ip 255.0.0.0 0.255.255.255 any log
access-list 101 deny ip 224.0.0.0 7.255.255.255 any log
!
! Deny packets without ip address.
!
access-list 101 deny ip host 0.0.0.0 any log
!
! Prevent spoofing. Deny incoming packets that have
! our internal address:
!
access-list 101 deny ip 12.34.56.0 0.255.255.255 any log
!
! More spoofing prevention. Insert ip address of external
! router interface ip address:
!
access-list 101 deny ip host 98.76.54.32 any log
!
! Allow only ACKed tcp packets to our network.  Note that all inbound
!packets will have the IP address of the external address of the
!firewall as their destination:
!
access-list 101 permit tcp any 12.34.56.78 gt 1023 established
!
!Enable the tcp intercept service with the following command:
!
```

```
ip tcp intercept list 101
!This will prevent SYN flood attacks
!
! Allow only specific ICMP:
! http://www.iana.org/assignments/icmp-parameters
! http://www.worldgate.com/~marcs/mtu/
!Simply denying all ICMP will cause us more problems than it would
!solve
access-list 101 permit icmp any 12.34.56.78 3 0 ! net-unreachable
access-list 101 permit icmp any 12.34.56.78 3 1 ! host-unreachable
access-list 101 permit icmp any 12.34.56.78 3 3 ! port-unreachable
access-list 101 permit icmp any 12.34.56.78 3 4 ! packet-too-big
access-list 101 permit icmp any 12.34.56.78 3 13 ! administratively-
prohibited
access-list 101 permit icmp any 12.34.56.78 4 ! source-quench
access-list 101 permit icmp any 12.34.56.78 11 0 ! ttl-exceeded
!
! Allow smtp traffic to notes servers only:
!
access-list 101 permit tcp any host 12.34.56.78 eq 1352
!
! Allow incoming dns traffic to name servers only:
! Note: Probably best to limit tcp domain traffic to specific
servers.
!
access-list 101 permit tcp any host 12.34.56.78 eq domain log
access-list 101 permit udp any host 12.34.56.78 eq domain
!
! Allow incoming news traffic to nntp server only:
!
access-list 101 permit tcp any host 169.254.92.103 eq nntp
!
! We deny ident. We're not sure if it's secure. Entry is here
! to keep log files from filling up:
!
access-list 101 deny tcp any any eq 113
!
! Log everything that does not meet the above rules.
!
access-list 101 deny ip any any log
!
! End of access-list 101
!
! Add this to external interface of screening router:
!
no ip directed-broadcast
no ip proxy-arp
no ip unreachables ! Don't send icmp for denied items in access-list.
ntp disable
!
! Apply access list to external interface:
!
ip access-group 101 in
!
! Use this command if you want to see denied hosts while
! logged into the router. Use command:
! "show ip accounting access-violations"
!
! ip accounting access-violations
!
!----------------
!Outbound filter:
!----------------
!
! Beginning of access-list 102
!
access-list 102 deny ip 192.168.0.0 0.0.255.255 any log
access-list 102 deny ip 172.16.0.0 0.15.255.255 any log
access-list 102 deny ip 10.0.0.0 0.255.255.255 any log
access-list 102 deny ip any 192.168.0.0 0.0.255.255 log
access-list 102 deny ip any 172.16.0.0 0.15.255.255 log
access-list 102 deny ip any 10.0.0.0 0.255.255.255 log
!
! Don't allow internal hosts to send icmp.
!
access-list 102 deny icmp any any log
!
! Only allow packets from our network.
!
access-list 102 permit ip 12.34.56.78 any
!
! Log everything else:
!
access-list 102 deny ip any any log
!
! End of access-list 102
!
! Apply access list 102 to outbound external interface
! or inbound on internal interface.
!
! Miscellaneous:
!
service password-encryption
service linenumber
no cdp run
no service finger
no service udp-small-servers
no service tcp-small-servers
no ip source-route
no ip bootp server
no ip http server
no ntp master
no ip domain-lookup ! If you don't have a name server.
no logging console ! Save cpu cycles.
logging buffered
!
! Cisco NTP information:
!
!http://www.cisco.com/univercd/cc/td/doc/product/software/ios11/sbook
/ssysmgmt.htm
! http://www.cisco.com/warp/customer/105/30.html
!
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
clock timezone PST -8 ! My timezone.
clock summer-time zone recurring
!No NTP server here, but keep if we need it later
!ntp source e0
!ntp update-calendar
!ntp server 196.254.92.38
!
! VERY VERY IMPORTANT! Log everything to syslog!  The FW syslog
daemon
!logs to an internal syslog server
!
logging 12.34.56.78
!
! Secure snmp with a community name other than public or private.
! Add access-list security.
!
snmp-server community secret RO 21
snmp-server trap-authentication
!
!
! --------------------------------
!Secure vty (Telnet) and aux port:
!--------------------------------
line aux 0
access-class 2 in
transport input all
line vty 0 4
access-class 1 in
password 7 Very3ecurePsswd!?
login

!
! Add access-lists:
!
! Allow only specific hosts to telnet into router:
!
access-list 1 permit 12.34.56.80
!
! Block access to aux.
!
access-list 2 deny 0.0.0.0 255.255.255.255
```

Perimeter Firewall

The perimeter firewall is the T. Rex application proxy firewall. This open source firewall can be downloaded for free, should the user wish to compile the code and configure the hardware and OS for maximum security. For most others, there are binaries and installers loaded onto CD-ROMs and also hardware appliances. The firewall is the central embodiment of the GIAC security policy, and its configuration reflects that. For the purposes of this paper, the following web addresses and ranges will be used:

Firewall exterior interface:	12.34.56.78	FW.GIAC.COM

Firewall interior interface:	172.16.1.1 (private address; NAT will be in use)	
Firewall screened subnet interface:	87.65.43.1	
Interior Network Range:	172.16.0.0 (Class B private range)	
Screened subnet range:	87.65.43.0	
Web Server:	87.65.43.2	WEBGIAC
Notes Server:	87.65.43.3	MAILGIAC
PKI Server:	87.65.43.4	PUBKEYGIAC
Interior DNS Server:	172.16.1.2	INTDNSGIAC
Exterior DNS Server:	56.78.90.12	ISPDNS

While these addresses may exist in the real world, any resemblance between this paper and the real owners is entirely coincidental. The names given to the servers signal their use for clarity in the paper; in a real world application, they would be given more random (and less descriptive) names to provide a little security through obscurity.

T. Rex uses configuration files for its proxies rather than a consolidated rule base like packet filters. However, within each proxy configuration file there may be a set of permit and deny rules similar in nature to a packet filter. There are many minor maintenance tasks that need to be done; this paper will not go into every configuration step for the sake of brevity, but will cover crucial proxy configurations. We will use the convention that file names or commands are in bold; the file contents as revealed by a **cat** command are italicized.

The first file is the Dual DNS config.

Internal DNS is contained in the file

/etc/firewall/resolv.inside.conf

```
domain        giac1.xyz       #non-routable extension .xyz used on interior network
nameserver    172.16.1.2
```

External DNS configuration is contained in the file

/etc/resolv.conf

```
domain                56.78.90.12    #IP of the ISP DNS Server
```

Now, tell the firewall which networks are protected:

/etc/firewall/securenets

```
172.16.       #secure net
87.65.43.     #secure net
```

Define the secure ports on the firewall (so the firewall will know which NICs are secured and so prevent spoofing based on IP). This requires the full IP address:

/etc/firewall/secureports

```
12.34.56.78    #secure interface
172.16.1.1     #secure interface
```

Set default group permissions (similar to NT policies).

/etc/gwuser.conf

```
# file: /etc/firewall/gwuser.conf system: T.Rex gateway
#Modified from Default conf file
# function: The /etc/firewall/gwuser.conf file is used by the gwuser
# database administration utility.
#
# (C) Freemont Avenue Software, Inc. 1995-2000.
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
# The following entries are mandatory, but the values can be changed
# Default group for new user
default_group = staff
# Default token type for new group (CRYPTO, SNK, SDI (AIX only), NONE)
default_token = NONE
# Default telnet access for new group (NO_TN, UNPRO_TN, PRO_TN, PRO_TNSA,
# ADM_TN, or combination; PRO_TN and PRO_TNSA are exclusive; ADM_TN also
# requires PRO_TNSA, UNPRO_TN, or both)
default_telnet = NO_TN
# Default FTP access for new group (NO_FTP, UNPRO_FTP, PRO_FTP, PRO_FTPSA,
# GET_FTP, PUT_FTP, DEL_FTP, or combination; GET_FTP, PUT_FTP, and DEL_FTP
# also require PRO_FTP, PRO_FTPSA, or UNPRO_FTP; PRO_FTP and PRO_FTPSA are
# exclusive)
default_ftp = NO_FTP
# Default time access for new group (ALL, NONE, MTuWThFSaSu, 1-12, A, P; a
# hyphen indicates a range, otherwise just the indicated day or hour applies;
# hours must be followed by A or P and modify preceding days; if hours are
# not specified, 12A-12P applies; minutes are not allowed)
```

```
default_times = ALL
# Password min length (4 - 16)
password_min_length = 10
# Password min alpha characters (0 - 16; password_min_alpha
# plus password_min_other must not exceed password_min_length)
password_min_alpha = 3
# Password min non-alpha characters (0 - 16; password_min_alpha
# plus password_min_other must not exceed password_min_length)
password_min_other = 2
# Password max change interval in days (UNLIMITED, 1 - 365)
password_max_age = 90
# Number of unique passwords before reuse (NONE, 1 - 12)
no_unique_passwords = 4
# Number of failed logins before locked (UNLIMITED, 1 - 12)
no_login_attempts = 3
```

Add an administrators group:

Format - gwgroup –a group –f FTPACCESS –t TELNETACCESS –times

gwgroup *–a admin –times ALL*

This rule prevents remote access to the firewall. All configuration to the firewall has to be done at the console. This can be a burden on administrators, but denying remote configuration adds another layer of security.

Add the administrative user to the gwusers database:

gwuser –a user –g group

gwuser *–a root –g admin*

Proxies

When setting up proxies, we are telling a stripped down (and presumably secure) version of the service in question what to let through, and to whom. The proxy is intended to ensure that the traffic going through is what it claims to be. The configuration file also allows for a limited amount of packet filter – style rules. Just as with a packet filter, the order of the rules count. The rule tells the proxy what traffic to let through, which interface to direct it to, and to which interior server the interface should redirect the traffic.

Aproxy

The aproxy is a generic proxy for connection oriented TCP/IP applications. It has access control and logging, giving it a finer grain control than the generic proxy that also comes with T. Rex. We want to allow connections to Lotus Notes (Notes does its own authentication); deny all other connections from the outside and allow all connections from the inside. The format is as follows:

permit from *src_addr* to *loc_addr loc_port* redirect (*dest_addr1, dest_addr2, ...)*
***dest_port* [using loc_addr2]**
[ignorerst [sleep_secs]] [userexit *exitname]*

deny from src_addr to loc_addr loc_port

```
# file: /etc/firewall/aproxy.conf system: T.Rex gateway
# function: The /etc/firewall/aproxy.conf file is used to control Aproxy.
# Aproxy permits or denies access to the requested service
# based on "permit" and "deny" rules found in this file.
#Modified from the default
#
# (C) Freemont Avenue Software, Inc. 1995 - 2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE
EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# The first permit statement allows all internal systems to access the exterior on any port.
# Any other host attempting to establish a connection through the
# aproxy  will be denied access.
#
gproxpath = /home/firewall
timeout = 300
permit from * to 12.34.56.78 1352 redirect 87.65.43.3 1352 #Exterior to Notes Server
permit from 172.16.*  to  172.16.1.1 1352 redirect 87.65.43.3 1352 #Interior to Notes Server
deny from * to 172.16.* to * *
```

FTP

The ftproxy is used to allow ftp transactions through the firewall. However, the company policy is not to let ftp through the firewall in either direction. The risks outweigh the convenience; if a file is truly needed a user will have to come to the IT department with a special request. In the file **/etc/firewall/ftproxy.conf**, the following line is all that's required:

*Deny * **

HTTP

Interior users have to be able to get out to the internet for web access. Outside users do not need to have HTTP access through the firewall and can all be denied. Access to the web server is via the webgate proxy, discussed later. According to the T. Rex documentation, the HTTP proxy has the following functions:

- The basic services for Web Browsing,
- Enforcement of management policies regarding use of the web,
- URL content filtering,
- Selective blocking of Java, JavaScripts, ActiveX and cookies,
- Caching of frequently used documents provides faster Web response,
- Conserves network bandwidth,
- pre-forking to eliminate up to 90% of system overhead,
- non-disruptive upgrades with on-the-fly scalability without downtime,
- fault tolerant mission critical design,
- automatic fail-over when configured as a redundant system,
- logs WWW access in Common Log Format,
- Report generation

While the HTTP proxy allows for blocking cookies, Java, and javascript, GIAC does not plan to do this. Neither will GIAC use the content filtering, for now. Because the proxy only allows users on the inside to access web sites outside, there is little security configuration. We simply have to tell it which networks are allowed to use the proxy.

Allow from 172.16.0.0/255.255.0.0

RPC

There is no reason to allow RPC across the firewall. We deny all traffic in the /etc/firewall/rpcproxy.conf file.

*Deny from * to **

Mail

Our users do need mail. Both internal and remote users access the Notes server through the firewall; their traffic goes through the aproxy, defined above. Outsiders sending mail do it via the domain name; the DNS lookup for our mail server gives the Notes server.

Our interest is in doing this without compromising the mail server or the rest of the network. T. Rex uses a proxy called smwrap. It is a small program running in a non-privileged state in a chroot'ed directory. This makes it harder to abuse. Smwrap has a number of security features such as anti-spoofing, spam control, header scrubbing, reporting, mail blocking, and multiple domain support. The smwrapd configuration file is actually quite small; only the deny list provides any security configuration:

```
# file: /etc/firewall/smwrap.conf
# function: The smwrap.conf file is used to control the execution of the
# smwrap and smwrapd programs.
#
# (C) Freemont Avenue, Inc. 1995 - 2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE
EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
#Block mail - deny from src_addr to rcpt_addr [block|forward|sequester]
deny from @evil.org to * block

timeout = 600
userid = hermes
groupid = staff
spoolpath = /home/hermes
maxbytes = 1500000
maxreceipts = 100
maxchildren = 10
smwpdpath = /usr/local/etc
smtppath = /usr/lib/sendmail
undelivpath = /home/hermes/sequestered
wakeup= 60
```

The alias file defines the mail server and users:

```
# file: /etc/firewall/aliases system: gw.your.domain
# function:
# created:by rjl@lsli.com 4/03/94
#Modifed from default
#
# (C) COPYRIGHT Freemont Avenue Software, Inc. 1994 - 2000
# All Rights Reserved
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OF MERCHANTABILITY AND FITNESS, FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR ANY AGENT OF FAS) ASSUME THE ENTIRE COST OF ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#mail server        @domain.name            @mail.server
                    @giac1.com              @mailgiac.giac1.com
#
```

```
# NOTE:
# /usr/local/etc/adam -b must be run after updating the
# aliases file. There will be a delay while the command updates
# the aliases data bases. When the command will print a
# summary of what it did.
#
# Replace root@system2 with the internal e-mail address of the
# systems administrator who should receive mail for
# MAILER-DAEMON, postmaster, and root. In this example
# the entry for root should appear before MAILER-DAEMON and
# postmaster.
#
# Aliases for root, mailer-daemon, and postmaster
root: root@giac1.com
MAILER-DAEMON: root@ giac1.com
postmaster: root@ giac1.com
# Aliases to handle mail to msgs and news
nobody: /dev/null
# Replace the following examples with your own external and internal names
# external_addr: internal_addr

# Aliases to handle mail to msgs and news
nobody: /dev/null
# Aliases in the form of first initial & last name
bing: bing@crosby
bud: bud@abbot
buster: buster@keaton
charlie: charlie@chaplin
```

Telnet

We do not want to let telnet through the firewall; external users have no legitimate need to access internal servers, and internal users can live without it.

```
# file: /etc/firewall/tnproxy.conf
# function: The tnproxy.conf file is used to control the execution of the
# tnproxy program. Connections between external and internal
# hosts are permitted or denied based on the permit and deny rules.
#
# (C) Freemont Avenue Software, Inc. 1995 - 2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE
#IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
#THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE
#EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# The format of the commands are as follows:
#
# timeout nnn the number of seconds genproxy will wait for a message
# before it exits.
```

```
#auth=yes requires a password to access the proxy
# deny src_host src_port
#
# permit src_host src_port to dest_host dest_port
auth = yes
groupid = staff
userid = hermes
tnproxpath = /home/hermes
timeout = 300
deny groups = * from * to *
```

The Web Server Proxy

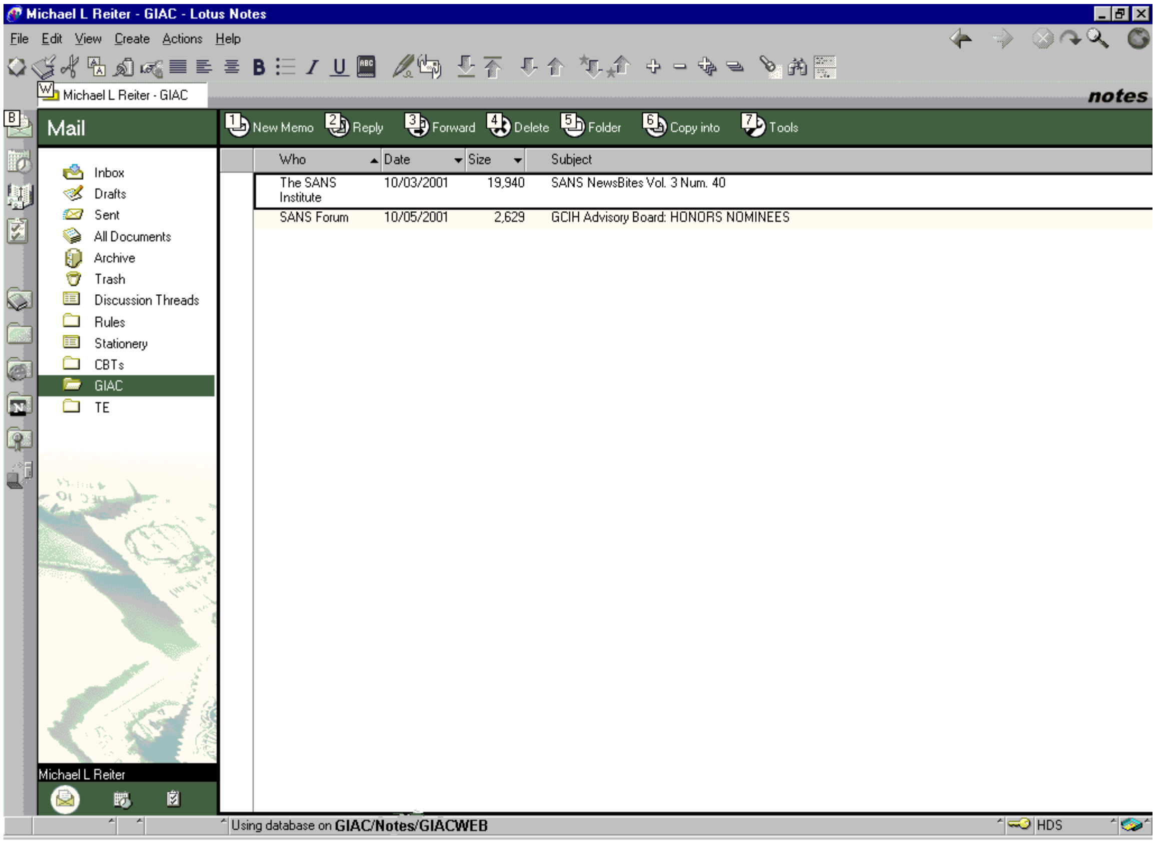
This is the most important proxy on the system, from the business standpoint. This is how we do business. Basically, we want to let internet users in to the screened subnet on port 443 (SSL), but not into the interior network. There is an implicit deny going inbound from the internet to any interior network.

```
# file: /etc/firewall/webgate.conf system: T.Rex gateway
# function: The /etc/firewall/webgate.conf file is used to control the web
# gateway program.
# The webgate program will permit or deny access to the web server(s) based
# on the use of the "permit" and "deny" rules found in this file.
#
# (C) Freemont Avenue Software, Inc. 1995 - 2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE
EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# The permit from statements allow all systems to access the secured web servers using
# HTTP with encryption (SSL)..
#
#
gproxpath = /home/hermes
httplog = http.access.log
timeout = 300
nprocs = 80 20 40
maxprocs = 200
maxuse = 1000
checkprocs = 10
# Allow traffic from the outside through the firewall to the screened subnet to the web server over
#SSL only; verbose logging
permit from * to 12.34.56.78 443 redirect 87.65.43.2 443 vlog
```

```
#block traffic originating from the web server to the interior network (anti-hacking measure)
deny from 87.65.43.2 * to 87.65.43.1 * redirect 172.16.* *
```

VPN

The VPN is actually a two part affair. New employees are provided with laptops/desktops pre-loaded with the Notes client and their Notes ID. For remote employees, the network is accessed via Lotus Notes. A remote user simply access the internet via dialup or LAN connection, and the Notes client installed on the laptop connects to the Notes server on demand over the internet. The client is password protected locally; for the internet connection, Notes uses its built-in PKI to verify that the connecting client has the appropriate key to authenticate with. Notes can serve both as a mail server and a database server. Certain databases are created and made available to selected users based on their Notes ID and key. The critical factor is ensuring that only authorized employees receive the Notes client and ID, and that access lists to the databases are kept up to date.



Who	Date	Size	Subject
The SANS Institute	10/03/2001	19,940	SANS NewsBites Vol. 3 Num. 40
SANS Forum	10/05/2001	2,629	GCIH Advisory Board: HONORS NOMINEES

Using database on GIAC/Notes/GIACWEB

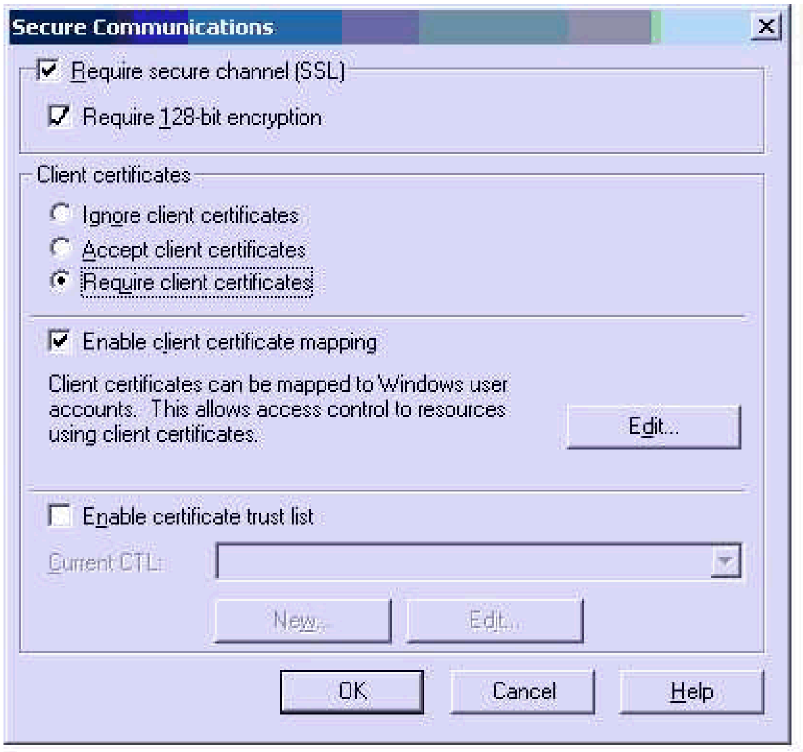
Figure 1: Lotus Notes VPN Screenshot

The partners and customers access the www.giacsales.com website. This is a web-based catalog; prices are accessed from lists customized for each customer or partner. The site is SSL and certificate enabled, giving each user an encrypted, authenticated connection. The Windows 2000 web server accesses the Windows 2000 Certificate Server. The Certificate Server and Web Server have been set up for secure access in accordance with the procedure previously outlined in detail by this author[1]. We will not go over the process of creating this in detail for this paper; however, certain key factors will be pointed out to illustrate the VPN policy.

Again, the important factor in the partner/customer PKI is not as much the precise configuration of the certificate-enabled web site, but in the authentication process prior to approving the digital certificate issuance. We will configure the site so that anyone may enroll for a digital certificate; however, we will only issue a digital certificate after we have personally contacted the applicant and run a credit check with a third party agency. All orders will be via line of credit; in this way, if someone should somehow gain unauthorized access to the site and illicitly order fortune cookie sayings, there will be no

[1] Creating a Certificate-Enabled Public Web Site With Windows 2000; Michael Reiter, May 2001; http://www.sans.org/y2k/practical/Michael_Reiter_GCNT.doc

line of credit and no sale or loss to GIAC.



Secure Communications

☑ Require secure channel (SSL)

☑ Require 128-bit encryption

Client certificates

○ Ignore client certificates

○ Accept client certificates

● Require client certificates

☑ Enable client certificate mapping

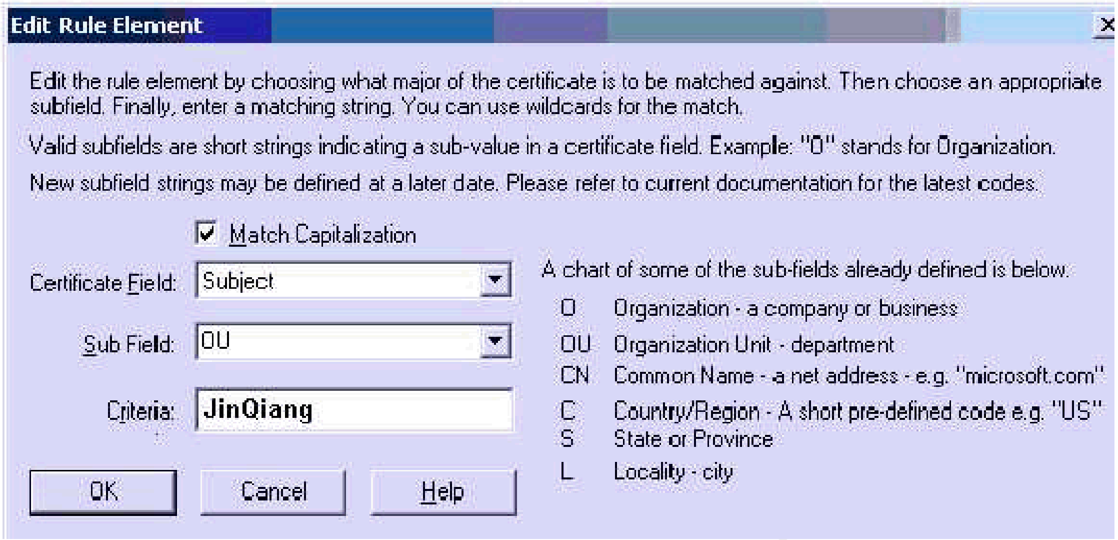
Client certificates can be mapped to Windows user accounts. This allows access control to resources using client certificates.

☐ Enable certificate trust list

Current CTL:

Figure 2: Configuring Certificate Access

In this configuration window from the IIS Server, we are requiring 128 bit SSL encryption along with a client certificate to access our web site. We will create user accounts and group accounts for customers and partners to provide fine-grained access control. Additionally, different partners and customers will have different Organizational Units (OU) in their digital certificates; this will be used as another distinguishing characteristic. This is illustrated in Figure 3, below, using the partner JinQiang Cookie Corporation..



Edit the rule element by choosing what major of the certificate is to be matched against. Then choose an appropriate subfield. Finally, enter a matching string. You can use wildcards for the match.

Valid subfields are short strings indicating a sub-value in a certificate field. Example: "O" stands for Organization.

New subfield strings may be defined at a later date. Please refer to current documentation for the latest codes.

☑ Match Capitalization

Certificate Field: Subject

Sub Field: OU

Criteria: JinQiang

A chart of some of the sub-fields already defined is below.

O	Organization - a company or business
OU	Organization Unit - department
CN	Common Name - a net address - e.g. "microsoft.com"
C	Country/Region - A short pre-defined code e.g. "US"
S	State or Province
L	Locality - city

Figure 3: Selection Criteria for a Partner Certificate

Section 3: Auditing the GIAC Security Architecture

Planning the Assessment

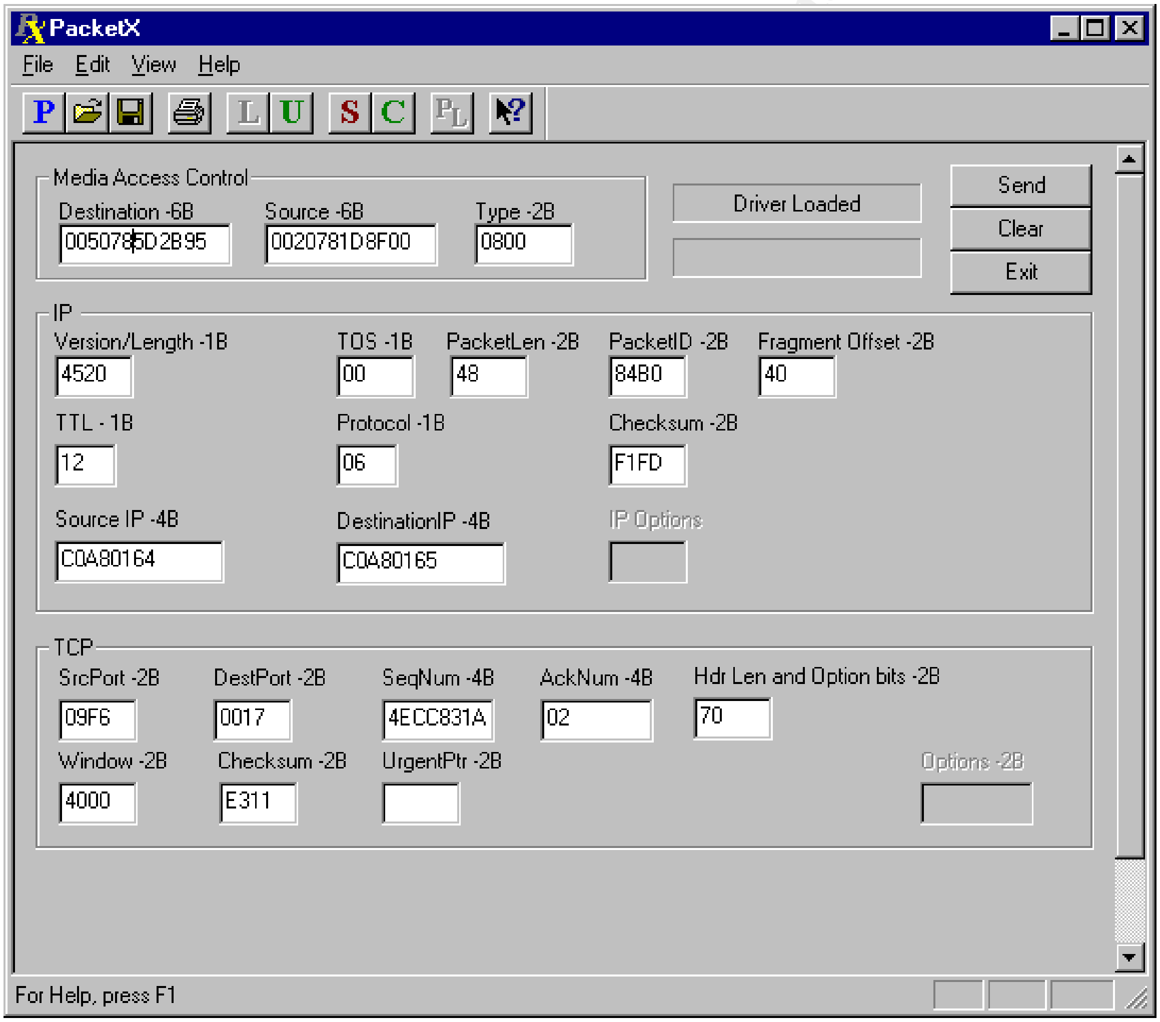
The goal is to assess the perimeter. Therefore, while we are certainly interested in an internal vulnerability assessment and will conduct one, this will not be discussed in this section. We want to find out whether our publicly accessible servers are vulnerable. This means we must conduct an external penetration test and also conduct host-based assessments of the security of the firewall, web server, and Notes server. There are testing tools and protocols available for both the router and the T. Rex firewall. Because we have a working system, we want to avoid conducting our tests during the working day; network loads may slow the site down. Additionally, we will not conduct Denial of Service (DoS) tests based on network loading. We have patched our servers and do not expect to have problems from the Ping of Death or similar exploits; however, a DDoS test that fills up the network pipe will almost certainly deny access to our site. We are a small company and cannot afford the kind of redundant siting and large bandwidth required to defeat these attacks; we accept this risk. Finally, we will analyze the IDS logs to see what penetrated the firewall. We estimate the level of effort as follows:

External Penetration:	40 hours
Host Assessments:	8 hours/server (24 hours total)
Router test:	4 hours
Firewall Test:	4 hours
IDS log analysis:	4 hours
Total:	**84 hours**
Cost ($50/hour)	**$4200**

Since a full Vulnerability Assessment from a professional services company of ethical hackers would cost $30,000 - $100,000, we will conduct this assessment in-house.

Implementing the Assessment

We have implemented the router ACL and wish to test its effectiveness. To do so, we will craft packets that implement (or violate) each of the rules in the router ACL. Software from NTObjectives (now part of Foundstone), called PacketX, will generate raw packets on Windows NT systems. The tool is not particularly user friendly – all packet parameters have to be entered in hex, and none are calculated (for instance, the checksum or length fields). A telnet packet is pictured below (MAC addresses altered for security reasons):



Media Access Control

Destination -6B: 0050785D2B95
Source -6B: 0020781D8F00
Type -2B: 0800

IP

Version/Length -1B: 4520
TOS -1B: 00
PacketLen -2B: 48
PacketID -2B: 84B0
Fragment Offset -2B: 40
TTL - 1B: 12
Protocol -1B: 06
Checksum -2B: F1FD
Source IP -4B: C0A80164
DestinationIP -4B: C0A80165

TCP

SrcPort -2B: 09F6
DestPort -2B: 0017
SeqNum -4B: 4ECC831A
AckNum -4B: 02
Hdr Len and Option bits -2B: 70
Window -2B: 4000
Checksum -2B: E311
UrgentPtr -2B:

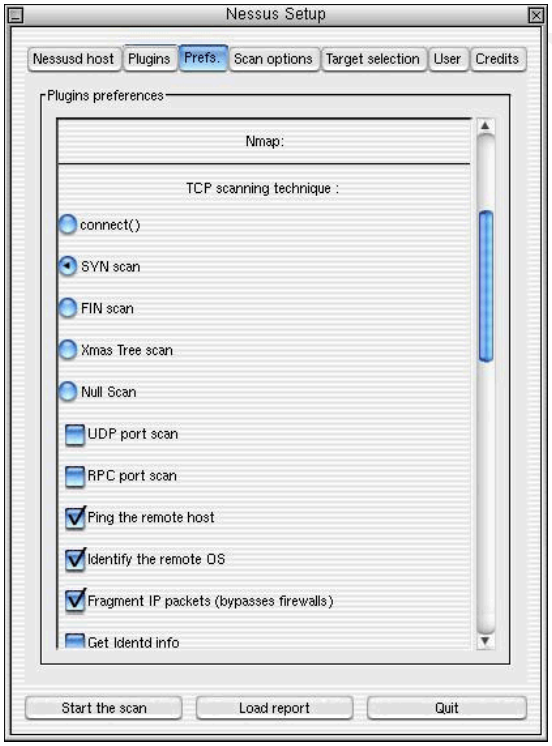
Figure 4: PacketX GUI with sample Telnet packet

Using this utility from the internet side of the router, we will generate packets originating from non-routable IP addresses, from the firewall exterior interface, legal and illegal ICMP packets, broadcast packets, and all others covered by our router ACL. On the other (interior) side of the router we will have a protocol analyzer (sniffer); any packet that makes it through the router will be logged. We will also check the router logs to ensure that we logged traffic as expected.

The firewall will be tested according to the protocol listed in Chapter 27 of the T. Rex Administration Manual, entitled "Testing T. Rex". The chapter appears in this paper as Appendix A.

After running these tests, we will use the tool nessus[2] to scan our system from the outside and see what "footprint" our system leaves for the outside world. Nessus is an open source vulnerability scanner that makes use of another open source tool, nmap[3]. Nmap is a port scanner and OS identifier. Nessus has a wide variety of checks, including many specific to Internet Information Server. Since it is open source and the scripting language for new vulnerabilities is widely available, tests for the latest vulnerability usually show up quickly. A test for the Nimda worm affecting IIS servers was on the nessus site on September 19th, for instance. Configuration of Nessus is shown in the series of figures below.

Figure 5: Nessus Nmap Configuration



Plugins preferences

Nmap:

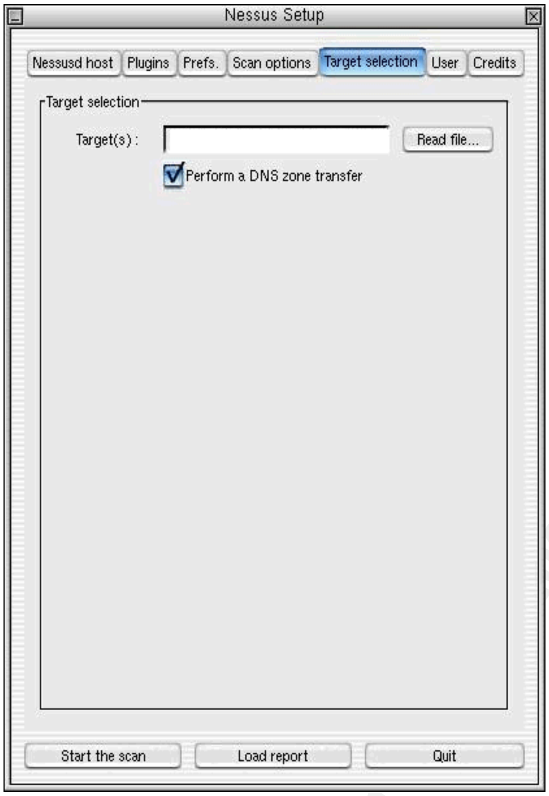
TCP scanning technique :

○ connect()
● SYN scan
○ FIN scan
○ Xmas Tree scan
○ Null Scan
□ UDP port scan
□ RPC port scan
☑ Ping the remote host
☑ Identify the remote OS
☑ Fragment IP packets (bypasses firewalls)
□ Get Identd info

In Figure 5 we see that we will be using a SYN scan format. SYN scans are somewhat stealthier than a connect() scan, since the connection is never completed (and therefore may not be logged). However, most systems today are aware of this and SYN scans are

[2] http://www.nessus.org
[3] http://www.insecure.org

no longer as stealthy as they were. We will first ping the remote host to see if it responds, and will send malformed TCP/IP packets to gauge the OS by checking the responses to the malformed packets. Finally, we will send fragmented packets to interior systems in an attempt to pass through the firewall.

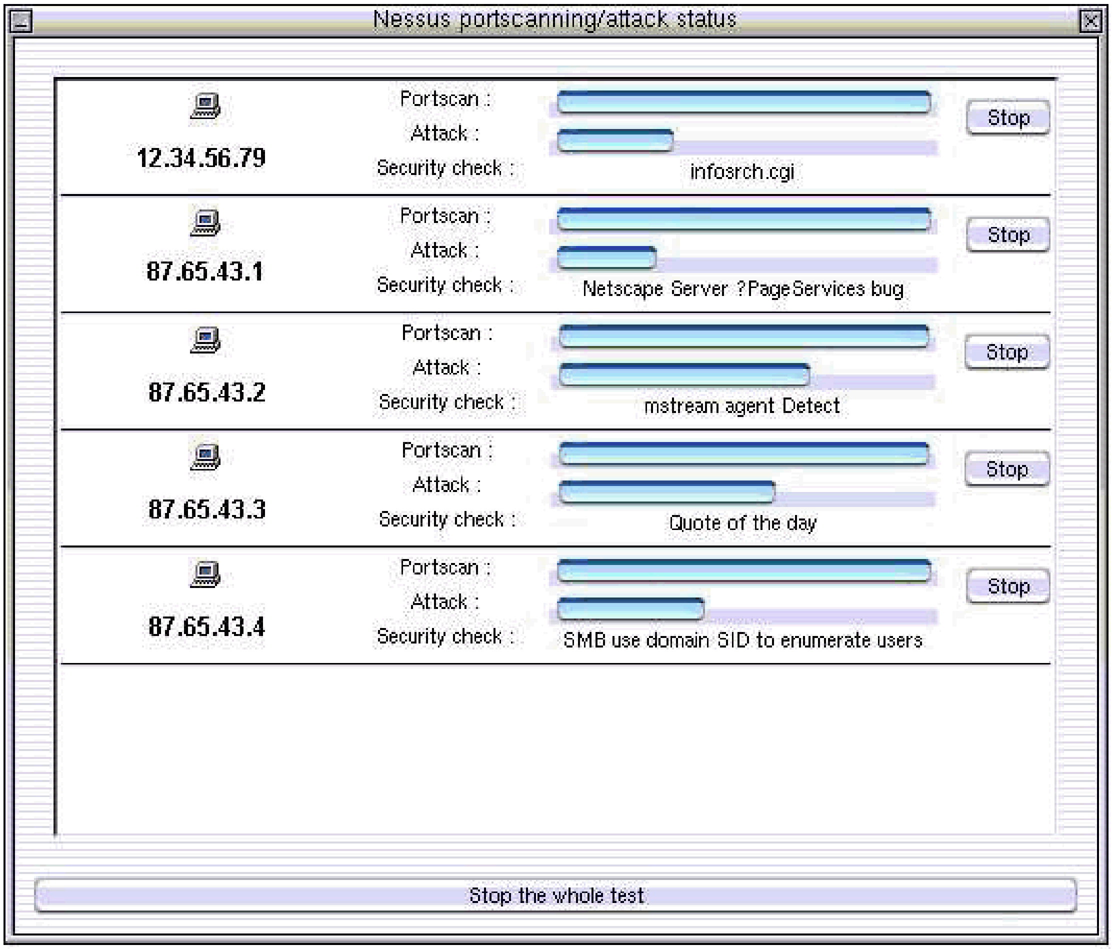


Target(s) :

☑ Perform a DNS zone transfer

Figure 6: Nessus Target Selection

In Figure 6 we see the Target Selection window. We have created a file with the lists of all the IP addresses in the screened subnet as well as the IP range for the firewall exterior / router interior (12.34.56.0/24). We include the whole range because it has happened in the past that people have installed devices without notifying anyone; if an illicit device is there, the scan will find it. We will also attempt a DNS Zone Transfer in an effort to gain data from the T. Rex DNS server.



12.34.56.79
Portscan :
Attack :
Security check : infosrch.cgi

87.65.43.1
Portscan :
Attack :
Security check : Netscape Server ?PageServices bug

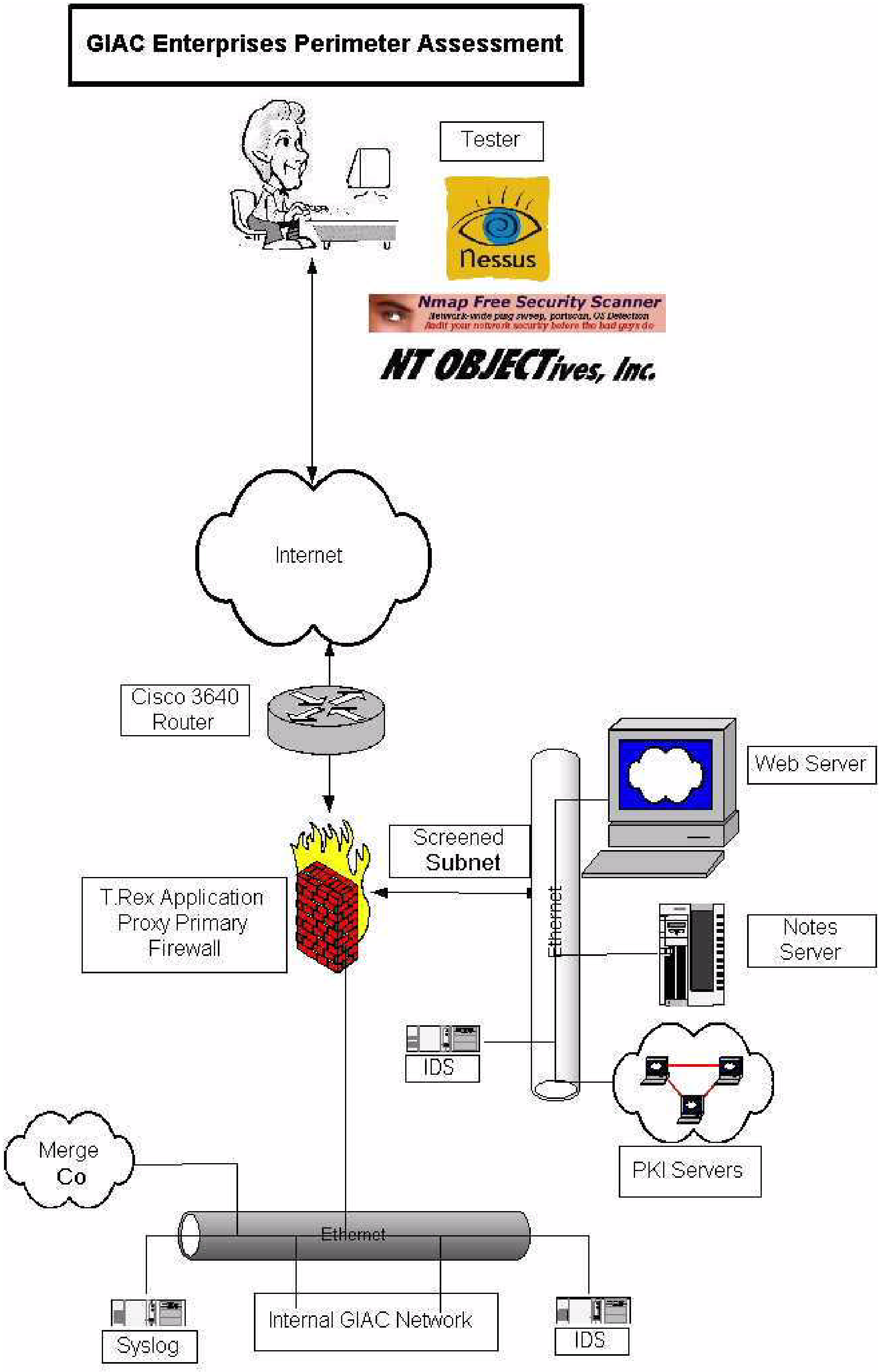
87.65.43.2
Portscan :
Attack :
Security check : mstream agent Detect

87.65.43.3
Portscan :
Attack :
Security check : Quote of the day

87.65.43.4
Portscan :
Attack :
Security check : SMB use domain SID to enumerate users

Figure 7: Example Scan Result

Figure 7 is a mock-up of a nessus scan status window. Since we did not actually build this system outside of the report, we cannot do a genuine scan.


GIAC Enterprises Perimeter Assessment
Tester
Nessus
Nmap Free Security Scanner
Network-wide ping sweep, portscan, OS Detection
Audit your network security before the bad guys do
NT OBJECTives, Inc.
Internet
Cisco 3640 Router
Web Server
Screened Subnet
T.Rex Application Proxy Primary Firewall
Ethernet
Notes Server
IDS
PKI Servers
Merge Co
Ethernet
Syslog
Internal GIAC Network
IDS

Perimeter Analysis

A search of USENET groups via Google did not reveal any vulnerabilities in the T. Rex firewall. Neither did a search of Packetstorm[4]. Of course, this does not mean there are none – simply none that have been discovered or published. Of greater concern is the underlying OS. A T. Rex installation automatically hardens the underlying OS, but there is no guarantee that this is effective. However, we can scan the system as described (simulated) and display the simulated output.

PortScan Output

PortScan is the utility included with T. Rex to do a self-test. We run this according to the instructions to look for any services:

```
$ /usr/T.Rex/portscan -h 65535 FW > portscan.output.txt
```

This command scans all possible ports and responds with their status, written to a file:

```
portscan: trying stream ports between 0 and 65535 on FW
service name port/tcp message
WARNING! DNS (53/tcp) is running. Make certain DNS is running as a caching-only server on
the firewall, and can only resolve external hosts.
domain 25/tcp is alive on FW.GIAC.COM(12.34.56.78)
domain 53/tcp is alive on FW.GIAC.COM(12.34.56.78)
https 443/tcp is alive on FW.GIAC.COM(12.34.56.78)
notes 1352/tcp is alive on FW.GIAC.COM(12.34.56.78)
ports scanned = 6001, number active = 3
```

This is what we intended, and so this test was successful. Running Nessus and nmap, we found that we could get limited access to the screened subnet servers; however, we could not discover any vulnerabilities to exploit. A simulated nmap output for the firewall is shown below:

```
Starting nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

Host  (12.34.56.78) appears to be up ... good.
Initiating SYN half-open stealth scan against  (12.34.56.78)
Adding TCP port 25 (state open).
Adding TCP port 53 (state open).
Adding TCP port 443 (state open).
Adding TCP port 1352 (state open).
The SYN scan took 10 seconds to scan 65535 ports.
Interesting ports on  (192.168.43.254):
```

[4] http://www.packetstormsecurity.com

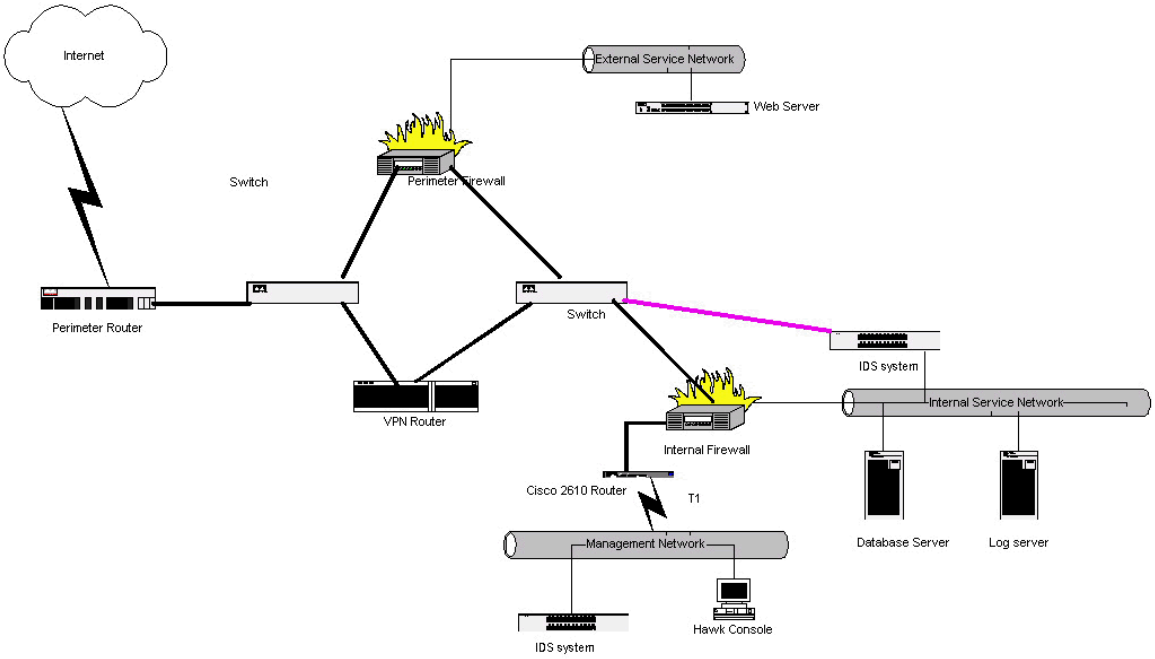
```
(The ports scanned but not shown below are in state: closed)
Port       State       Service
25/tcp     open        mail
53/tcp     open        dns
443/tcp    open        https
1352/tcp   open        notes


TCP Sequence Prediction: Class=random positive increments
                         Difficulty=3627591 (Good luck!)

Sequence numbers: 8C46016D 8CF5E867 8C7C1D87 8C44485F 8C5348A0 8CBD5C04
Remote OS guesses: Solaris 2.7

Nmap run completed -- 1 IP address (1 host up) scanned in 10 seconds
```

Again, this confirms our configuration. A small amount of information may be released to the world if someone else runs this software. However, we found that the firewall did not perform the DNS zone transfer when requested by the nessus scanner; this is good. The https port will accept connections but only allows access to secure functions after a public/private key exchange. Similarly, the Notes server will only allow usage after its own public/private key exchange. Therefore, the information is of little use unless a new exploit for these services becomes available.

How could we improve the perimeter defenses or security architecture? One way to improve the defense against flooding attacks (and to increase the robustness of the network) would be to add redundant servers with failover connections. For instance, if we had two identical firewalls with load balancing, the possibility of flooding would be reduced. Also, if one failed the other could pick up the load. We could make this even more robust with physically separate ISPs, each having redundant servers. Of course, this would significantly increase the cost of maintaining the web site, and GIAC is still a comparatively small company. In addition, all servers need to have integrity checkers such as Tripwire. If vital files or binaries change unexpectedly, there is a good chance that the server has been hacked. The Tripwire output should alert administrators to changes; of course, the administrators have to follow up on alerts.
The use of a second firewall in front of the T. Rex firewall – preferably a stateful inspection packet filter like Checkpoint Firewall 1 – would add additional security if programmed correctly. By filtering out numerous attacks before they ever reach the application proxies, we remove another large portion of the threat.

Design Under Fire

http://www.sans.org/y2k/practical/Tamara_Bowman_GCFW.doc



Equipment Installed	
Perimeter	
Perimeter Router	Cisco 7505 IOS ver 12.0(5)
Perimeter Firewall	Cisco PIX 525 ver 5.3(1)
VPN Router	Cisco 7140-2FE IOS ver 12.0(7)T
IDS Device	Sun Netra, Solaris 2.8, 2 NICS. The NIC on the perimeter switch does not have an IP address
External Service Net	
Web Server	Sun Netra, Solaris 2.8, Netscape Server
Inside Firewall	Sun Netra, Solaris 2.8, Raptor 6.4
Internal Service Net	
Oracle Database	2 Sun 4500, Solaris 2.8 running Sun Cluster and Oracle 8
Management Net	
Log Server	Sun 450, Solaris 2.8
Raptor Management Station	Sun Ultra 5, Solaris 2.8 running Raptor Hawk

The perimeter firewall is a Cisco Pix 525 version 5.3(1). The Cisco Pix is a packet filter whose primary advantage is its fantastic speed and throughput. As such, it is unlikely to be affected by a small DDoS attack.
The Pix configuration as listed allows Secure Shell access. SSH was found to have a weakness and Cisco posted a vulnerability notice at http://www.cisco.com/warp/public/707/SSH-multiple-pub.html on June 27, 2001. According to the notice, the following impacts apply:

> **CRC-32 integrity check vulnerability**
>
> By exploiting this protocol weakness, the attacker can insert arbitrary commands in the session after the session has been established.
>
> **Traffic analysis**
>
> This vulnerability exposes the exact lengths of the passwords used for login authentication. This is only applicable to an interactive session that is being established over the tunnel protected by SSH. This can significantly help an attacker in guessing the password using the brute force attack.
>
> **Key recovery in SSH protocol 1.5**
>
> This vulnerability may lead to the compromise of the session key. Once the session key is determined, the attacker can proceed to decrypt the stored session using any implementation of the crypto algorithm used. This will reveal all information in an unencrypted form.

Since any host on the internet may access port 22 (ssh), this may compromise the perimeter firewall. If we can decrypt traffic streams, we have no need to go any further – the traffic will give us illicit access to purchased fortune cookie sayings. This may be fixed by upgrading to version 5.3(2), available in August 2001.

Since Ms. Bowman has implemented the TCP Intercept feature, her systems are unlikely to be affected by a SYN flood, especially one originating from only 50 home PCs (major servers are unlikely to be accessing the internet via cable/DSL modems). ICMP is denied at the exterior interface of the router; again, there should be no problem for internal systems. On the other hand, a UDP flood such as those caused by Trinoo directed at the internal web server may have an effect. Since Trinoo does not spoof the IP addresses of the zombies sending the flood, they should not be affected by router rules. The best defense a small company can effect against the Trinoo-type attacks is to have the ISP detect and block these types of attacks prior to ingress.

How would we compromise an internal system through the perimeter? We would take advantage of human nature. There is no indication in the paper of software or policies to deny access to mail attachments; therefore, sending a trojan to a vulnerable individual is a likely method of ingress. I would target the non-technical people, because they are more likely to open an untrusted attachment in an email. To identify the targets, I would check sources such as the "Who We Are" page on the web site (executives are famously clueless), the company directory, the automated phone exchange, and any other public source of data. I would also make phone calls asking to be put through to an executive's secretary; by identifying this person (and extracting information such as the email address format), I can get multiple likely targets. I want to avoid the system and network admins, for obvious reasons. Many trojans might be suitable; however, I want one that will connect to me through the firewall. Since packets originating within the network (and their replies) travel freely through the firewall, this is ideal. Examples might be slightly modified versions of rwwwshell[5] or httptunnel[6]. The trojan should be wrapped in another program so that the user does not suspect a trojan. Another method of trojan installation (for the truly dedicated and those willing to risk jail time) would be to gain physical access posing as a janitor, delivery person, or other semi-trusted individual and quickly use a floppy disk with an installer script to load the code. Modifying the code would allow it to be customized for a particular platform (rwwwshell requires Perl, not commonly found on Windows desktops, but httptunnel has a Windows binary). Modifying the code may also help it escape virus detection.

Once a single desktop or server has been compromised, additional machines may be compromised in an island-hopping fashion. Ultimately we will get root or administrator on a machine, or find a critical trust to exploit. Additional trojans may be planted so that multiple backdoors exist. A sniffer should be installed and set to mail logs to an anonymous account. This will reveal passwords, many of which will be for critical accounts. By keeping a low profile (not running exploits gratuitously or generating large amounts or suspicious types of traffic) we may escape detection. Ultimately we should be able to compromise the entire network in this fashion.

In short, the perimeter defense is only that. Mail traffic penetrates the perimeter in both directions, and web traffic originating from the interior is usually not inspected. Physical access, gained via many different ways, also penetrates the perimeter. Defense in depth is the only way to stop this – through the use of virus checkers, host and network based intrusion detection systems, integrity checkers, traffic analysis, and other methods. By making life difficult for the hacker at every step, we can prevent or stop most attacks.

[5] http://thc.pimmel.com/files/thc/fw-backd.htm

[6] http://www.nocrew.org/software/httptunnel.html

References

http://www.pasadena.net/cisco/secure.html by Frank Keeney, Pasadena Networks

Creating a Certificate-Enabled Public Web Site With Windows 2000; Michael Reiter, May 2001; http://www.sans.org/y2k/practical/Michael_Reiter_GCNT.doc

http://www.nessus.org ; Nessus

http://www.insecure.org ; Nmap

http://www.packetstormsecurity.com

http://thc.pimmel.com/files/thc/fw-backd.htm; rwwwshell

http://www.nocrew.org/software/httptunnel.html; httptunnel

Appendix A

From the T. Rex Administration Manual

Chapter 27. Testing T.Rex

This chapter shows you how to use the test programs that ship with the T.Rex firewall system.

27.1 Testing for IP Packet Forwarding

During the T.Rex installation process the netinet module is replaced with a version that has IP Packet forwarding disabled. To ensure that IP Packet forwarding has been properly disabled use the ping command to try and ping hosts on the opposite sides of the firewall. Suppose host 192.168.220.2 is on a protected network and that 198.65.130.21 is on an unprotected network.

Addressing the unprotected network from the protected network

From the protected system enter the ping command. You should not see any response. Wait five or six seconds then interrupt the command with **CNTL c.** You should then see a message indicating that 100% of the packets sent were lost.

```
$ ping 198.65.130.21
PING 198.65.130.21: (198.65.130.21) 56 data bytes.
6 packets transmitted, 0 packets received, 100% packet loss.
```

Addressing the protected network from the unprotected network

Repeat the process from the unprotected side. Again you should see 100% packet loss.

```
$ ping 192.168.220.2
PING 192.168.220.2: (192.168.220.2) 56 data bytes.
6 packets transmitted, 0 packets received, 100% packet loss.
```

27.2 Testing the Domain Name Server

When properly installed the T.Rex firewall will be running a **caching-only** Domain Name Server that can resolve host names on the unprotected network. The DNS should not be able to resolve internal names. To make certain external users can't use DNS to lookup protected hosts use the nslookup command.

From either an unprotected host or the firewall itself issue the nslookup command using the name of an host on a protected network.

```
$ nslookup fido
Server: lsli-port.sccsi.com
Address: 198.65.130.21
*** No address information is available for fido
```

Testing Ports with portscan

The portscan utility can be used to test which ports on the T.Rex firewall are active. Use this program to validate that applications you want to run are listening to their respective ports and

that dangerous applications are not active and listening.
The portscan program is in the /usr/T.Rex directory. You can copy it to another directory if you choose.
The syntax of the portscan command is as follows:

portscan [-l low_port] [-h high_port] [-f services_file_name] host
-l low_port This is used to specify the low port for the scan process. The port number must be a positive integer, less than 64000. The default value is 0.

-h high_port This is used to specify the high port for the scan process. The port number must be a positive integer larger than the low port number. The default value is 32000. If you run with the default ports it will take less than 55 seconds to scan all 32000 ports using an Ethernet LAN.

-f services_file_name This is the fully qualified path name of a file containing the service entries.
This file must have the same format as the /etc/services file. If a file is not specified then portscan will use the hosts **/etc/services** file.

host The host name of the system to be scanned. This must be the name of the host or the IP address of the host in dot-decimal format (eg. gw.lsli.com or 198.65.130.22).

27.4 Portscan Services file

The format of the services file is the same as the /etc/services file. Each service is listed on a separate line. The format of each line is as follows:

ServiceName PortNumber/Protocol Aliases or comments

ServiceName The service name specifies the official Internet service name. This name can be from 1 to 16 characters long.

PortNumber The socket port number used for the service (0 - 64000).

Protocol The transport protocol used for the service ("**tcp**" or "**udp**").
The items on each line can be separated by one or more blanks or tabs. Comments begin with a '**#**'and continue to the end of the line. The PortNumber and the protocol can be separated by a '**/**' or a'**,**'.

27.5 Sample Services file

A sample services file would look like the following:

```
# file: /etc/services.gw used to test portscan
#
# (C) COPYRIGHT Freemont Avenue Software, Inc. 1995
# All Rights Reserved
# Licensed Materials - Property of Freemont Avenue Software
#
#
echo 7/tcp
echo 7/udp
discard 9/tcp sink null
discard 9/udp sink null
systat 11/tcp users
daytime 13/tcp
daytime 13/udp
netstat 15/tcp
qotd 17/tcp quote
```

```
chargen 19/tcp ttytst source
chargen 19/udp ttytst source
ftp-data 20/tcp
ftp 21/tcp
telnet 23/tcp
smtp 25/tcp mail
time 37/tcp timserver
time 37/udp timserver
rlp 39/udp resource # resource location
nameserver 42/udp name # IEN 116
whois 43/tcp nicname
domain 53/tcp nameserver # name-domain server
domain 53/udp nameserver
mtp 57/tcp # deprecated
bootps 67/udp # bootp server port
bootpc 68/udp # bootp client port
tftp 69/udp
rje 77/tcp netrjs
finger 79/tcp
http 80/tcp #WWW server
link 87/tcp ttylink
supdup 95/tcp
hostnames 101/tcp hostname # usually from sri-nic
iso_tsap 102/tcp
x400 103/tcp
x400-snd 104/tcp
csnet-ns 105/tcp
pop 109/tcp postoffice
sunrpc 111/tcp
sunrpc 111/udp
auth 113/tcp authentication
sftp 115/tcp
uucp-path 117/tcp
nntp 119/tcp readnews untp # USENET News Transfer Protocol
ntp 123/tcp
NeWS 144/tcp
snmp 161/udp # snmp request port
snmp-trap 162/udp # snmp monitor trap port
smux 199/tcp # snmpd smux port
src 200/udp # System Resource controller
exec 512/tcp # rexec
biff 512/udp comsat
login 513/tcp # rlogin - dangerous on a firewall
who 513/udp whod
shell 514/tcp # rshd - dangerous on a firewall
syslog 514/udp
printer 515/tcp spooler # line printer spooler
talk 517/udp
ntalk 518/udp
efs 520/tcp # for LucasFilm
route 520/udp router routed
timed 525/udp timeserver
tempo 526/tcp newdate
courier 530/tcp rpc
conference531/tcp chat
netnews 532/tcp readnews
netwall 533/udp # -for emergency broadcasts
uucp 540/tcp uucpd # uucp daemon
new-rwho 550/udp
remotefs 556/tcp rfs_server rfs # Brunhoff remote filesystem
rmonitor 560/udp
monitor 561/udp
securid 755/udp # added by rjl for security dynamics
socks 1080/tcp # socks
instsrv 1234/tcp # network install service
ingreslock 1524/tcp
writesrv 2401/tcp # temporary port number
```

```
securidprop 5510/tcp # added by rjl for security dynamics
```

27.6 Sample portscan output

The following example shows the normal output of the portscan command. The default output will go the users terminal. The output can also be directed to a file if you want to save it, as shown below.

```
$ /usr/T.Rex/portscan -h 6000 gw > portscan.ouput.941210
```

To display the output on you screen enter the command:

```
$ /usr/T.Rex/portscan -h 6000 gw

portscan: trying stream ports between 0 and 6000 on gw
service name port/tcp message
echo 7/tcp is alive on gw.lsli.com (198.65.130.22)
discard 9/tcp is alive on gw.lsli.com (198.65.130.22)
daytime 13/tcp is alive on gw.lsli.com (198.65.130.22)
chargen 19/tcp is alive on gw.lsli.com (198.65.130.22)
ftp 21/tcp is alive on gw.lsli.com (198.65.130.22)
telnet 23/tcp is alive on gw.lsli.com (198.65.130.22)
smtp 25/tcp is alive on gw.lsli.com (198.65.130.22)
time 37/tcp is alive on gw.lsli.com (198.65.130.22)
WARNING! DNS (53/tcp) is running. Make certain DNS is running as a caching-
only server on the firewall, and can only resolve external hosts.
domain 53/tcp is alive on gw.lsli.com (198.65.130.22)
smux 199/tcp is alive on gw.lsli.com (198.65.130.22)
1026/tcp is alive on gw.lsli.com (198.65.130.22)
writesrv 2401/tcp is alive on gw.lsli.com (198.65.130.22)
cppbrowse 4242/tcp is alive on gw.lsli.com (198.65.130.22)
6000/tcp is alive on gw.lsli.com (198.65.130.22)
ports scanned = 6001, number active = 14
```

Sample portscan output (with warnings)

The following output shows the warning messages generated when dangerous applications are running on the target host. The command is run using all the default values. The warnings are for **rexecd, rlogin, rshd** and **uuc**p, none of which should be running on a firewall.

```
$ portscan duo
portscan: trying stream ports between 0 and 32000 on duo
service name port/tcp message
echo 7/tcp is alive on duo.lsli.com (192.168.220.2)
discard 9/tcp is alive on duo.lsli.com (192.168.220.2)
daytime 13/tcp is alive on duo.lsli.com (192.168.220.2)
chargen 19/tcp is alive on duo.lsli.com (192.168.220.2)
ftp 21/tcp is alive on duo.lsli.com (192.168.220.2)
telnet 23/tcp is alive on duo.lsli.com (192.168.220.2)
smtp 25/tcp is alive on duo.lsli.com (192.168.220.2)
time 37/tcp is alive on duo.lsli.com (192.168.220.2)
WARNING! DNS (53/tcp) is running. Make certain DNS is running as a caching-
only server on the firewall, and can only resolve external hosts.
domain 53/tcp is alive on duo.lsli.com (192.168.220.2)
```

```
smux 199/tcp is alive on duo.lsli.com (192.168.220.2)
WARNING! Use of exec on port 512 is considered UNSAFE!
exec 512/tcp is alive on duo.lsli.com (192.168.220.2)
WARNING! Use of login on port 513 is considered UNSAFE!
login 513/tcp is alive on duo.lsli.com (192.168.220.2)
WARNING! Use of shell on port 514 is considered UNSAFE!
shell 514/tcp is alive on duo.lsli.com (192.168.220.2)
WARNING! Use of uucp on port 540 is considered UNSAFE!
uucp 540/tcp is alive on duo.lsli.com (192.168.220.2)
1026/tcp is alive on duo.lsli.com (192.168.220.2)
writesrv 2401/tcp is alive on duo.lsli.com (192.168.220.2)
cppbrowse 4242/tcp is alive on duo.lsli.com (192.168.220.2)
6000/tcp is alive on duo.lsli.com (192.168.220.2)
spc 6111/tcp is alive on duo.lsli.com (192.168.220.2)
7685/tcp is alive on duo.lsli.com (192.168.220.2)
ports scanned = 32001, number active = 20
4 WARNING(S) ISSUED.
```