



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

GIAC Enterprises

An Internet perimeter architecture

Introduction

GIAC Enterprise is a medium-sized business that manufactures printed "fortune cookie sayings" to cookie makers. The management team has decided to use the Internet to support a distributed staff of authors, as well as most business transactions with customers and partners, using VPNs to reduce costs and streamline processes.

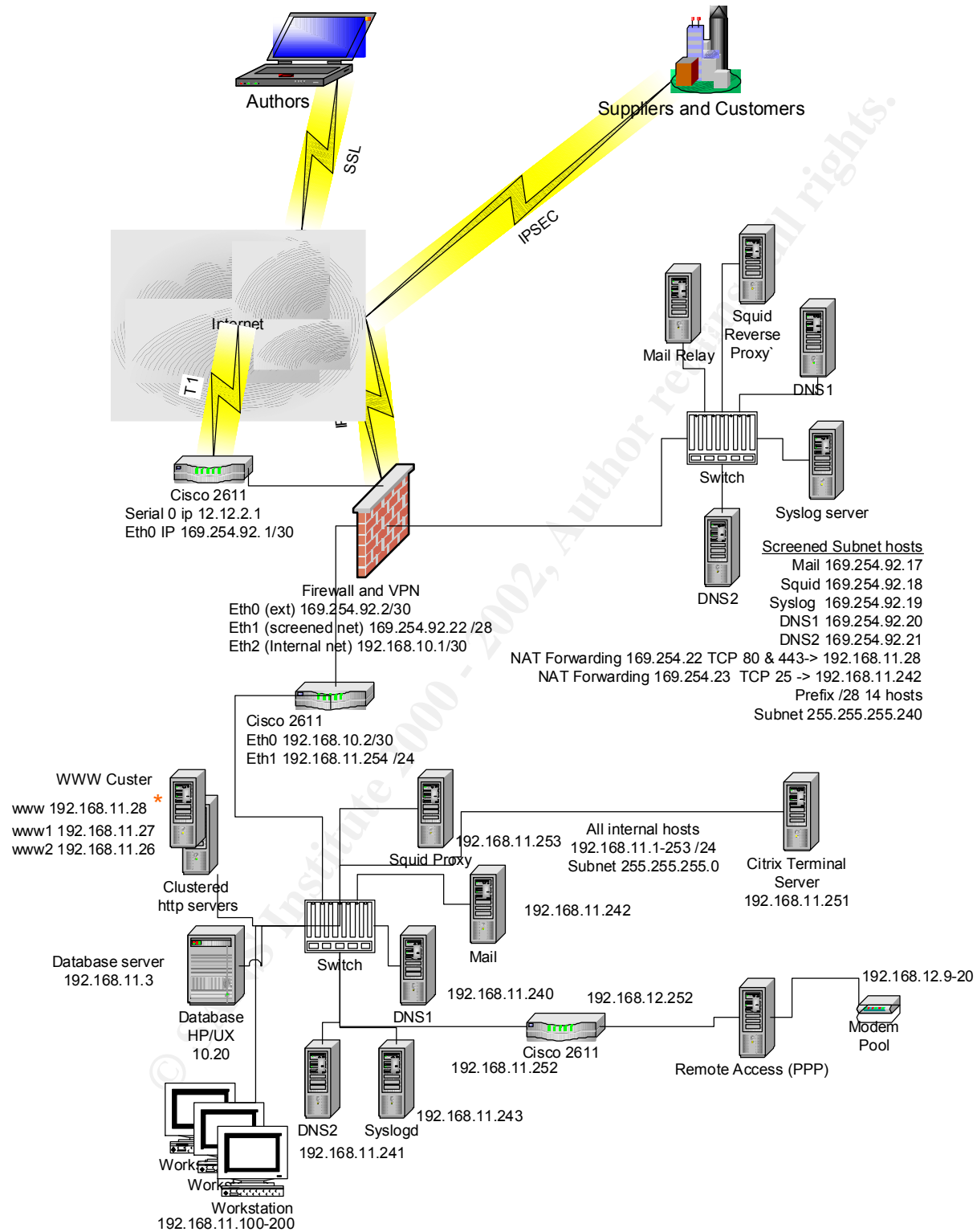
Requirements

The desired result of this endeavor is to provide services using current Internet technologies including the WWW and VPNs wherever they provide the most competitive and strategic advantages. A web site will be used to collect sayings from authors and will maintain the identity of the author for payroll. Provisions for partnering businesses, customers, and suppliers to access data for ordering, billing and data sharing will also be made.

Wherever possible, cost and administrative overhead will be kept to a minimum while maintaining a highly available secure perimeter. These contradictory goals are driven by the desire to succeed in a competitive marketplace while maintaining minimal support staff.

Goals

- The goals of the GIAC network are to provide:
 - Customers with the ability to purchase sayings in bulk using VPNs.
 - Suppliers (authors) with the ability to submit new sayings from anywhere in the world using the WWW.
 - Secure transactions with raw materials vendors (order ink, paper and office supplies).
- Network priorities:
 - System reliability and availability.
 - Data assurance.
 - Compliance with "good netizen" practices.
- Architecture:
 - 2 [Cisco](#) 2611 routers.
 - 1 [Watchguard](#) "Firebox II" firewall appliance and VPN.
 - Switches for all Ethernet ports.
 - ISP is a "tier 1" service provider.
 - Reverse proxy for public web access
 - Cache proxy for employee web access
 - PPTP remote user VPN and PPP based dial up access



Giac Enterprises Network Architecture

Security Policy

Border Routers

The GIAC network edge begins with a Cisco 2611 router connecting GIAC's network with their Internet service provider. The router is the primary connection point for Internet communications. Protecting it is critical to maintaining service. Furthermore, it is the first point of entry to and the last point of egress from GIAC's network. It has been configured for basic packet filtering to limit the types of traffic that can get to the firewall. Access lists are packet filters in Cisco IOS. They increase CPU load on a router and ought to be used sparingly on lower end routers. In this scenario, access lists will be used to protect the router and provide a minimal first layer of defense for the network by blocking the private (RFC 1918) address blocks and test network blocks, as well as detecting and blocking spoofed, loopback, multicast, and Class E addresses.

Rules

Unused services such as HTTP management, small servers (services with port numbers under 10) and Finger will be disabled. Outgoing ICMP and source route attempts will be blocked. Telnet, from all but administrative management stations, will be blocked. While a router can handle a lot, keeping the rule set small for performance purposes is good idea.

Border router configuration

<pre>version 12.0 service timestamps debug uptime service timestamps log uptime service password-encryption ! hostname gw-giac-rem ! enable secret 5 \$1\$eU2F\$5hgV5Y3N/ScE3sx8ErTwC/ ! username giacadmin password 7 0505031B2643 ip subnet-zero no ip source-route no ip finger ip name-server ISP NS1 ip name-server ISP NS2 ! no ip bootp server ! interface Ethernet0/0 ip address 169.254.92.1 255.255.255.224 no ip redirects no ip unreachable no ip directed-broadcast no ip proxy-arp no ip route-cache no ip mroute-cache ! interface Serial0/0 ip address 12.12.2.1 255.255.255.252</pre>	<p>IOS Version Set services including MD5 passwords</p> <p>Set router hostname</p> <p>Set the enable secret password with MD5 hash</p> <p>Create a uid and password for auditing system configs Disable potentially harmful services sourceroute and finger and bootp server.</p> <p>Using interface eth 0/0 command Disable icmp messages proxy-arp and smurf attempts See netscan.org and http://www.powertech.no/smurf/ for more info on Smurf and Fraggle</p> <p>Using interface ser 0/0 command Enable access lists</p>
---	--

<pre> ip access-group ser1-in in ip access-group ser1-out out no ip redirects no ip unreachable no ip directed-broadcast no ip proxy-arp no ip mroute-cache no fair-queue ! ip default-gateway 12.12.2.1 ip classless ip route 0.0.0.0 0.0.0.0 12.12.2.1 ip route 169.254.92.16 255.255.255.240 169.254.92.2 no ip http server ! ip access-list extended ser1-in deny ip 169.254.92.0 0.0.0.31 any log deny ip host 255.255.255.255 any log deny ip 127.0.0.0 0.255.255.255 any log deny ip 224.0.0.0 15.255.255.255 any log deny ip 240.0.0.0 7.255.255.255 any log deny ip 10.0.0.0 0.255.255.255 any log deny ip 172.16.0.0 0.15.255.255 any log deny ip 192.168.0.0 0.0.255.255 any log permit ip any 169.254.0.0 0.0.0.31 log ip access-list extended ser1-out deny icmp any any echo-reply log deny icmp any any mask-reply log deny icmp any any redirect log deny icmp any any time-exceeded log deny icmp any any timestamp-reply log deny icmp any any unreachable log permit ip 169.254.92.0 0.0.0.31 any deny ip any any log ip access-list extended telnet-in permit tcp 169.254.92.2 0.0.0.3/ 169.254.92.1 eq telnet deny ip any any log logging 169.254.92.19 ! banner motd ^CC *****WARNING***** THIS IS A PRIVATE COMPUTING FACILITY. UNLESS YOU HAVE BEEN SPECIFICALLY AUTHORIZED, YOUR CONTINUED ATTEMPTED ACCESS WILL EXPOSE YOU TO CRIMINAL AND/OR CIVIL PROCEEDINGS. ***** WARNING***** ^C ! line con 0 login local logging synchronous transport input none line aux 0 login local transport input none </pre>	<p>Disable ICMP messages proxy-arp and Smurf attacks</p> <p>Set up routing</p> <p>Disable HTTP server</p> <p>Create named access lists: Deny spoofing Deny directed broadcast Deny loopback Deny multicast Deny class E Deny RFC 1918 addresses Permit packets destined for us and log it all!</p> <p>Egress list: Deny ICMP Deny ping reply Deny Subnet mask reply Deny redirects Deny all time exceeded Deny Time stamp replies Deny all unreachable Permit our traffic Deny all else and log it!</p> <p>Restrict telnet to our nat- ted IP</p> <p>Specify the syslog host</p> <p>Display a banner to warn away unauthorized users.</p> <p>Login local requires local password</p> <p>Set an md5 password, limit</p>
---	--

```
line vty 0 4
 password 5 dsf9sdfo3el30
 ip access-class telnet-in in
 logging synchronous
 login local
 transport input telnet
!
end
```

transport to telnet and apply access-list filter to limit telnet sources.

These settings provide a basic level of protection for the router and prevent many of the preliminary information gathering techniques used by hostile persons to start mapping networks. To create and apply a named access-list enter configure mode and perform the following steps:

```
gw-giac-rem#conf t
Enter configuration commands, one per line. End with CNTL/Z.
gw-giac-rem(config)#ip access-list extended serlout
gw-giac-rem(config-ext-nac)# deny icmp any any echo-reply log
gw-giac-rem(config-ext-nac)# deny icmp any any mask-reply log
gw-giac-rem(config-ext-nac)# deny icmp any any redirect log
gw-giac-rem(config-ext-nac)# deny icmp any any time-exceeded log
gw-giac-rem(config-ext-nac)# deny icmp any any timestamp-reply log
gw-giac-rem(config-ext-nac)# deny icmp any any unreachable log
gw-giac-rem(config-ext-nac)# permit ip 169.254.92.00.0.0.31 any
gw-giac-rem(config-ext-nac)# deny ip any any log
gw-giac-rem(config-ext-nac)# exit
```

To apply the access list to the interface:

```
gw-giac-rem(config)# int serial 0/0
gw-giac-rem(config-if)# ip access-group serlout out
gw-giac-rem(config-if)# exit
```

Testing

After applying access control lists, it is necessary to test them and verify that they function as desired. Some very simple tests can be performed to ensure access controls are working. For example to test ICMP rules, the traceroute command can be used from external sites or using web sources such as <http://www.network-tools.com/> with the router as the target. The results are expected to time out at the Border router. To test the handling of spoofed source addresses by egress filters, tools like Hping or Nmap can be used to target known Internet hosts while monitoring “show access-lists” command output on the router. The number of matches will increment each time a rules criterion is met.

```
gw-giac-rem #sho access-lists
Extended IP access list ser1-in
 deny ip host 255.255.255.255 any log
 deny ip 127.0.0.0 0.255.255.255 any log
 deny ip 224.0.0.0 15.255.255.255 any log
 deny ip 240.0.0.0 7.255.255.255 any log
 deny ip 10.0.0.0 0.255.255.255 any log (444 matches)
 deny ip 172.16.0.0 0.15.255.255 any log (21 matches)
 deny ip 192.168.0.0 0.0.255.255 any log (517 matches)
 permit ip 169.254.0.0 0.0.255.255 any (21773986 matches)
Extended IP access list ser1-out
```

```
deny icmp any any echo-reply log
deny icmp any any mask-reply log
deny icmp any any redirect log
deny icmp any any time-exceeded log
deny icmp any any timestamp-reply log
deny icmp any any unreachable log
permit ip 169.254.0.0 0.0.255.255 any (10251871 matches)
deny ip any any log (3532 matches)
```

Using the debug command while logging to the monitor will also show the effects of these rules. For example, this denied ICMP Ping-Reply packet:

```
21w6d: %SEC-6-IPACCESSLOGDP: list ser1-out denied icmp 169.254.92.12 ->
192.44.162.132 (0/0), 1 packet
```

A log entry uses the following fields:

```
Uptime; Process; List name; protocol; source address; destination address;
(ports or type/codes) number of packets seen
```

To test Telnet access filters, try to Telnet to the router from an unauthorized host and then try from an authorized host. Reconcile the log output to the results of the test and determine whether they agree.

Internal Router

A screening router has been deployed between the corporate network and the firewall. This router will incorporate access-lists as well. The goal of this router is to stop spoofing and outgoing undesirable traffic and add another defensive layer to the network.

The named access-list for the internal router filters spoofed traffic:

```
gw-giac#conf t
Enter configuration commands, one per line. End with CNTL/Z.
gw-giac(config)#ip access-list extended egress
gw-giac(config-ext-nac)# permit ip 192.168.11.0 0.0.0.255 any
gw-giac(config-ext-nac)# deny ip any any log
gw-giac(config-ext-nac)# exit
```

Applying the access list to the interface:

```
gw-giac(config)# int eth 0/0
gw-giac(config-if)# ip access-group egress in
gw-giac(config-if)# exit
```

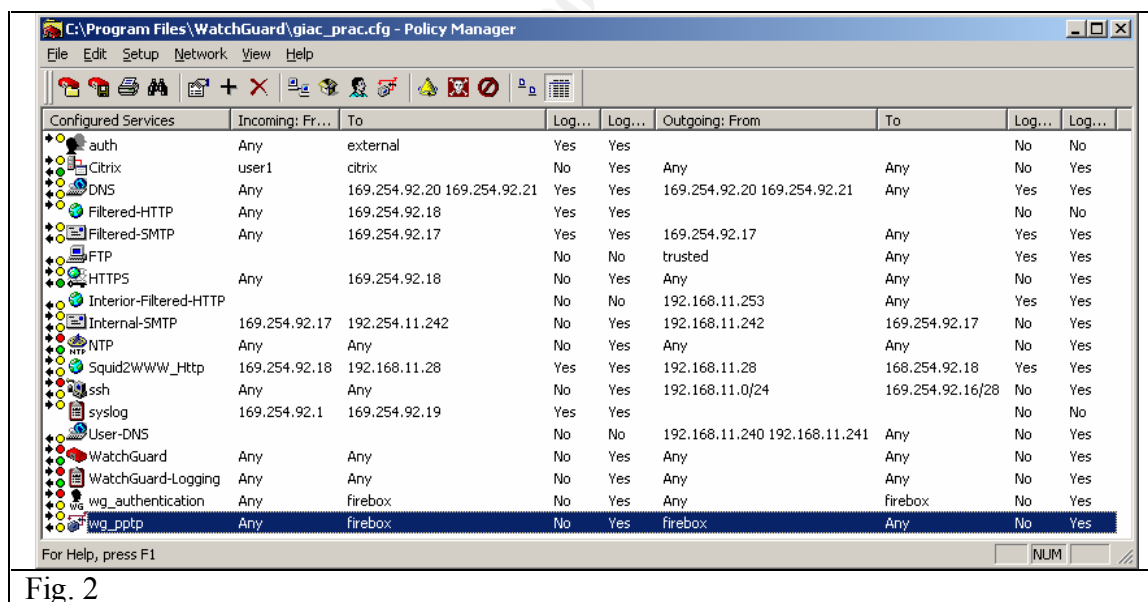
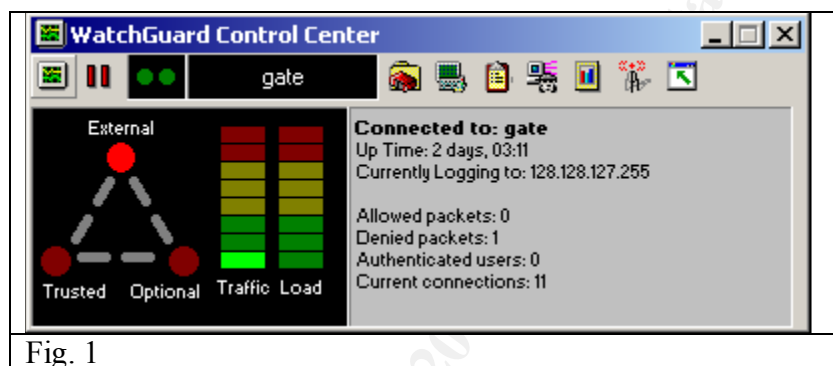
Cisco routers and IOS provide many options for traffic filtering. The router's functions are critical to communications. It is the front line in a layered approach to security design. Protecting it is critical and provides a natural point to begin filtering out attempts to gather information about a network.

Primary Firewall

The Watchguard Firebox II (<http://www.watchguard.com/>) is a firewall appliance. It is a Linux 2.0.XX based computer that runs in a flash disk. There is no shell or other program allowing real time interaction with the system. It is configured and managed by a Windows or Linux based management station.

Configuration and Rules

The Watchguard Control center (Fig.1) is the core management application. After logging in with a read only password, a system administrator is granted access to the management applications. The tools in the control center include the Policy Manager (Fig. 2) for system configuration; the System Monitor for viewing system statistics; the Log Viewer and the Host Watcher, which give a real-time graphical view of traffic seen by the Firebox; and the report generator. Backweb is used to send patches and notices to customers.



The Policy manager is the program used to configure a Firebox. It creates and uploads the configuration to the firebox to enforce the policy. The Read-write password is required to upload a new configuration file.

There are three interfaces built in to the system for “external,” “optional,” and “trusted” networks. The network menu in Policy manager is used to set each interface's IP address, routes,

default gateway, logging host(s) and VPN settings (see VPN section later). The setup menu sets system defaults such as default packet handling for spoofed addresses, port and address probe detection and blocking, and IP options handling and logging of events (See logging section later.) Network address translation is configured by using the Setup menu / NAT and identifying for which networks to perform the translation. GIAC's policy is to translate all traffic from the trusted network to the external and not to translate traffic from the trusted network to the optional network or the optional network to the external network.

Once the network settings are configured, rules are required. To add a rule, select the Edit menu and click Add Service. A list of pre-configured (stateful) packet filters, proxies, and user (custom) filters opens (Fig. 3). Select the filter and click Add. The service configuration dialog (Fig. 4) opens with three tabs, Incoming, Outgoing, and Properties. Rules are configured by object, address, or username. Objects are defined in the setup authentication menu and consist of user, host, network, or interface. On the incoming tab in the from field, click Add to define a source address or object that the rule should act upon. Repeat for the to field. Click the outgoing tab and set the to and from fields. Each rule can be set to "enabled and allowed," "enabled and denied," or "disabled" for each direction.

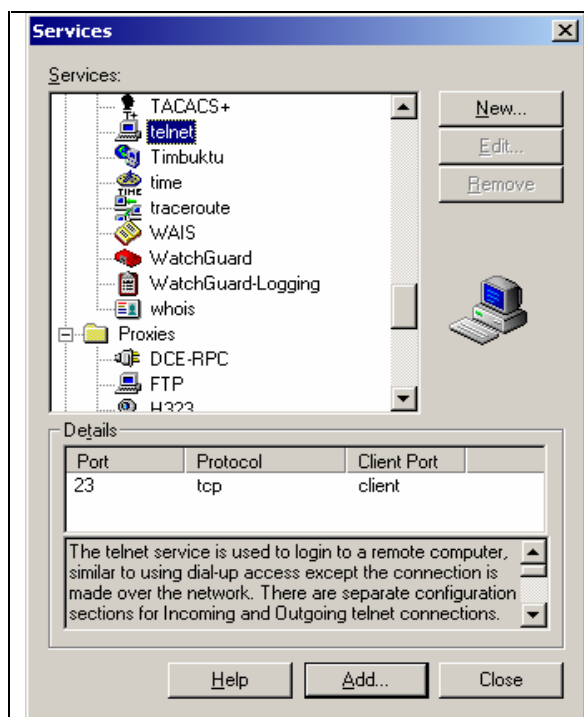


Fig. 3

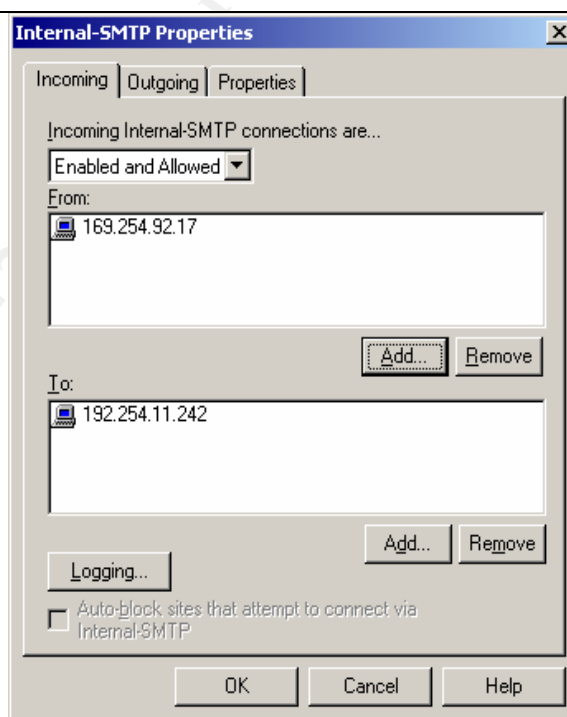


Fig. 4

The logging button will log allowed (Fig. 5) and denied packets in either direction for each rule. Furthermore, actions can be defined for notification by email, pager, popup window (NetBios) or custom programming. These features are configured from the logging button for each rule. The log host is configured with email addresses, pager numbers, NetBios names, and path to custom programs in the "Event Processor."

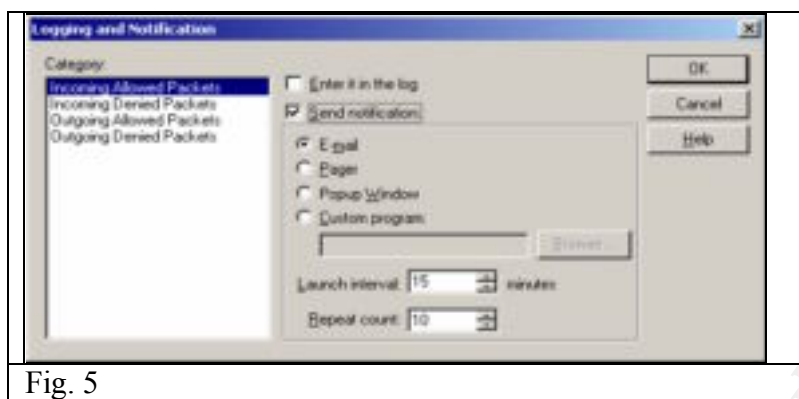


Fig. 5

The properties tab of the service dialog displays the destination port number, protocol (TCP, UDP, IP etc), and client port. The date the service was added and any administrative comments are displayed in the comments field.

A word on precedence: rules do not need to be configured in any order. They will be acted upon in a most specific to least specific order when the rule set is running. Thus a rule specifying access for a host will trigger before a more general rule that contains the same host.

Rules

GIAC has chosen to keep the rule set simple. The Firebox will allow direct Internet access only to the mail-relay host and proxy on the Screened Subnet. Internal hosts will not be allowed out of the “Trusted” subnet except for the Mail server and Proxy as specified. The overall stance is “Deny all that is not explicitly allowed.”

Rules will be broken into three groups:

Group 1 Internet Rules:

From any to host 169.254.92.18 TCP 80 and 443 WWW/SSL

From any to Network 169.254.92.0/29 TCP 113 Auth, Enabled and denied (to RST any Auth attempts and speed mail processing)

From any to host 169.254.92.17 TCP 25 SMTP

From any to host 169.254.92.20 and 21 UDP 53 DNS (NO TCP zone transfers not allowed)
DNS servers do not provide recursion

From “Cable and DSL nets” to host 169.254.92.2 TCP 1723 and GRE (IP protocol 88) PPTP

From Valid [pptp user] to host 192.168.11.251 TCP 1494 Citrix-ICA

“Cable and DSL nets” = 12.0.0.0/8 212.110.0.0/16 24.0.0.0/8

Group 2 Screened Subnet to Internal net

From host 169.254.92.18 to NAT Port TCP 80 and 443 forwarded to host (169.254.92.22) to 192.168.11.28 WWW/SSL

From host 169.254.92.17 to NAT Port TCP 25 forwarded host (169.254.92.23) to 192.168.11.242 SMTP

Group 3 Internal net

From 192.168.11.253 to Any TCP 20, 21, 80, 443 FTP/WWW/SSL

From 192.168.11.242 to 169.254.92.17 TCP 25 SMTP

From 192.168.11.240 and 241 to any UDP 53 DNS

From 192.168.11.28 to 169.254.92.18 TCP 80 and 443 WWW/SSL

Logging

Logging is used to track events that are considered to be significant to administrators based on the security policy. The log contains data about events that are marked for logging or are not recognized by the configuration and are treated as significant. Watchguard uses a proprietary logging system. Logs are sent over encrypted session to, and stored on the management PC that has a logging daemon running on it. An export of a sample log is displayed in the Logging application (Fig. 6). Below the log application is a text export of the log entry that shows detail cut off in the screen shot (Fig. 7).

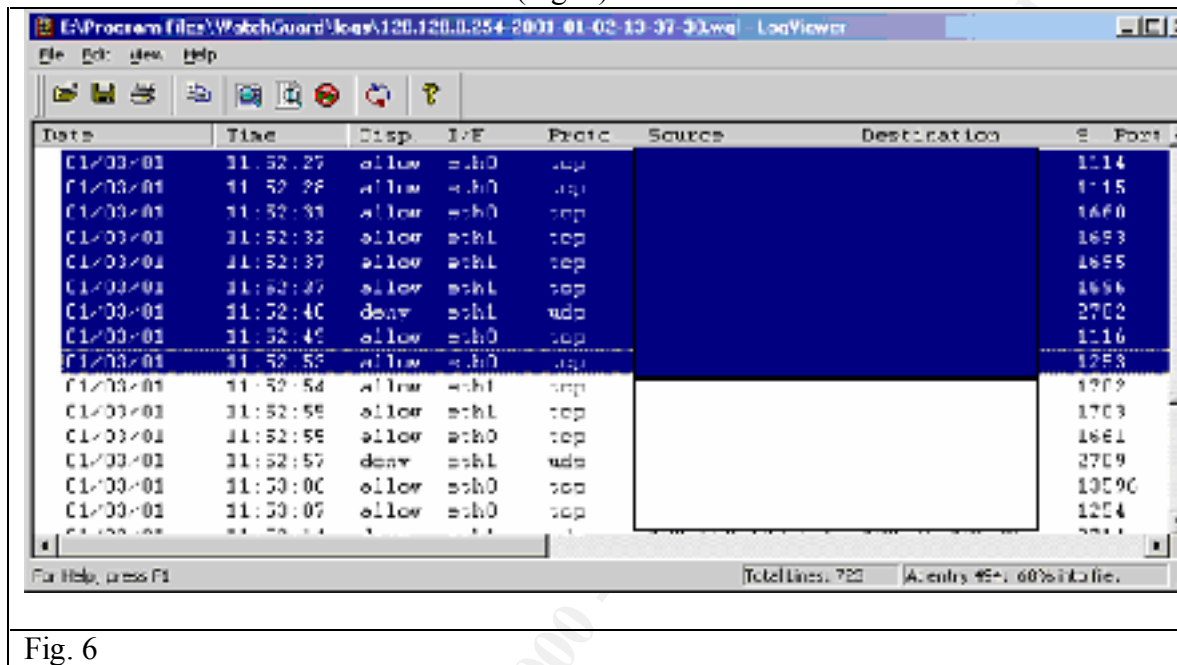


Fig. 6

Date	Time	Disp	Dir	Int	Hdr	Prot	IP Hdr	TTL	S. IP	
D. IP	S port	Dport	Flags	Rule	Comment					
01/02/01	14:35:02	deny	in	eth0	60	tcp		20	52	24.112.73.109
169.254.92.24	3450	111	syn	(blocked site)	(Address Space Probe)					
01/02/01	14:37:39	allow	in	eth0	44	tcp		20	46	63.162.15.250
169.254.92.6	4020	25	syn	(SMTP)						
01/02/01	14:50:31	deny	out	eth1	44	tcp		20	31	192.168.11.33
207.188.7.157	1157	80	syn	(default)						

Fig. 7

When a service is set to log activity in the policy, only packets with the SYN flag set are entered (Fig.5). Unusual flag patterns, detected port or address scans are logged. When a host is added to the dynamic blocked host list it is entered into the log. The corporate policy states that all logs will be saved online for three months and then moved to offline storage (CD or Tape) and stored for 6 months.

Firebox II logs can also be sent to syslog daemons (udp 514) on a *nix or other syslog hosts. Manually editing the configuration file outside of the Policy Manager is required. To do so save the configurations file to a local file. Open the file with a text editor. Near the beginning of the file add the line:

debug.syslog.host: 192.168.11.243 (or your syslog host)

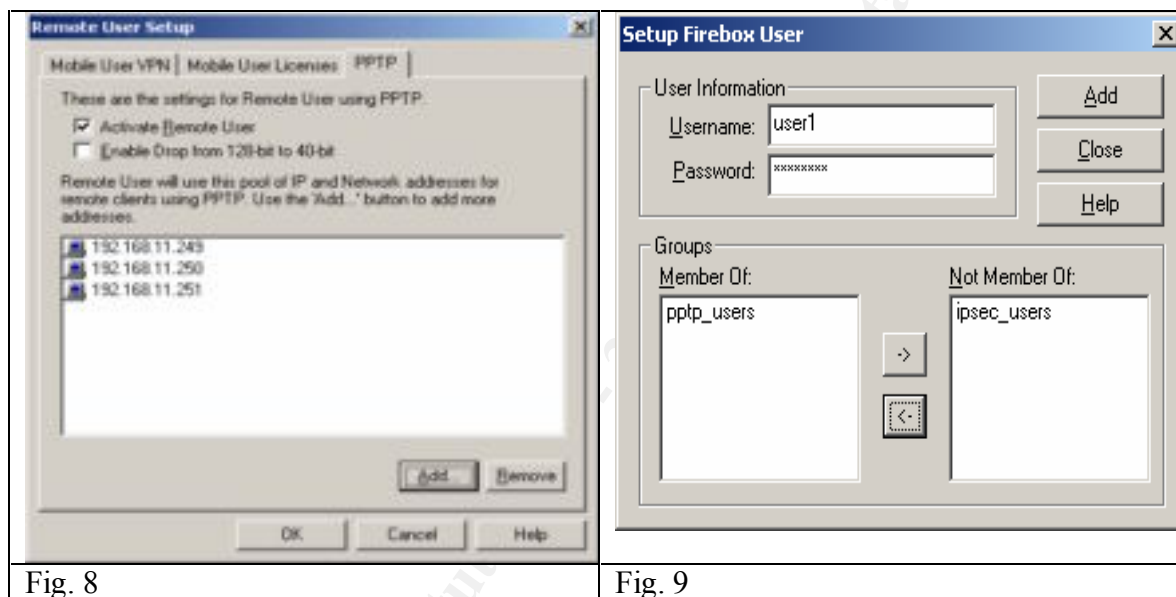
Save the file, reopen in Policy Manager and save to the firebox. It will begin to send syslog messages to the host(s) specified in the configuration.


Testing

Tests run using Nmap and Nessus on the Firebox II most often trigger the port or address scan detection module, resulting in the scanning station being unable to enumerate much data from the network. This is a nice thing when up against someone with little sophistication. Firewall also did not reveal much about the network. Logs and output from these programs are listed under the audit and assessment section. Attempts to connect to well known ports of advertised services such as WWW and SMTP ports were more fruitful.

Remote access VPN using PPTP

Configuration and Rules



The PPTP based remote user VPN is setup by using the Policy manager. Select the Network menu / remote user. The Remote User Setup dialog (Fig. 8) opens. “Activate Remote User” must be checked. This will cause the `wg_pptp` service to appear in the Policy Manager service list. Only check the Enable drop to 40 bit if users without 128bit PPTP clients must be supported. Click the Add button to create a network address pool. Addresses will be assigned from the pool for the duration of a PPTP session. Setup Firebox User (Fig. 9) is accessed from the Setup menu / Authentication. A local user directory for VPN users is used. Add a username and password for each PPTP user and add them to the `pptp_users` group with the  button. There are two more steps to PPTP setup on the Firebox. Add the `pptp_users` group or individual pptp users to services they are allowed to access in the From field of services that remote users are allowed to access (Fig. 10).

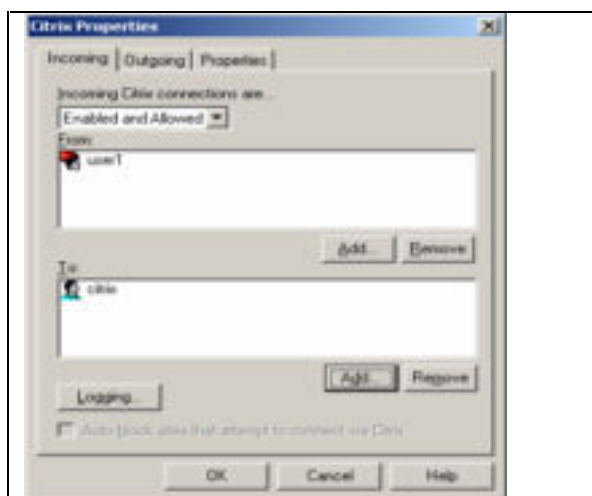


Fig. 10

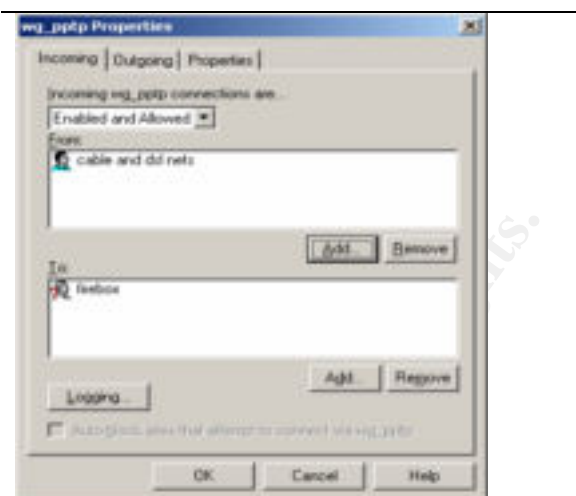


Fig. 11

In the wg_pptp service properties are restricted specific source network numbers (Fig. 11). GIAC only allows PPTP access from local Cable and DSL providers and has created an alias based on the network number called “Cable and DSL nets.” The alias is created in the Setup / Authentication applet in Policy Manager as depicted in Fig. 12 and 13.



Fig. 12

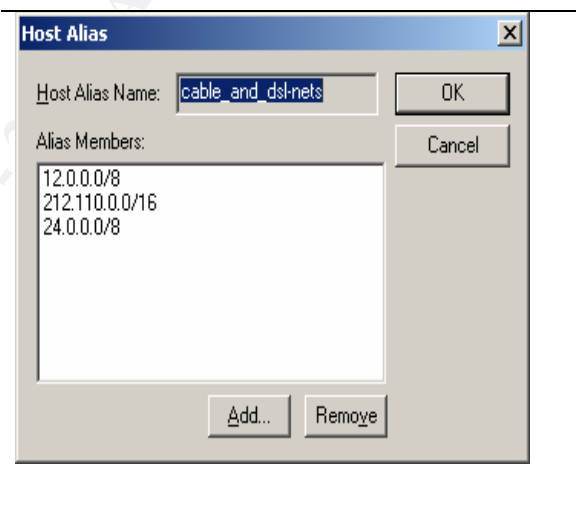


Fig. 13

GIAC's policy requires remote access passwords to be at least 7 alpha-numeric characters with 1 non-alpha-numeric in the password. They are changed by the administrators every 90 days and new passwords are delivered in person for remote access systems. PPTP is only used for remote access for employees.

VPN using IPsec

The Firebox supports two types of network to network VPNs: Watchguard and IPsec. The IPsec VPN was chosen by GIAC because it is an open standards-based protocol that is more likely to be interoperable with other vendors' equipment than the proprietary Watchguard protocol. GIAC has chosen to maintain IPsec tunnels with raw materials suppliers.

Configuration and Rules

The security association policy for IPsec tunnels set by GIAC requires ISAKMP key negotiation, Encapsulated Security Payload (ESP) to encrypt IP headers with data, with SHA1 authentication and 3DES encryption. New keys will be exchanged every 24 hours or 8Mb of data transferred.

The tunnel is built in Policy manager with the Network / Branch office VPN / IPSEC menu. In IPSEC configuration (Fig. 14), policies are created starting with a gateway where gateway's name, key negotiation type, remote gateway IP and a shared key (for ISAKMP only) are defined (Figs. 15 and 16).

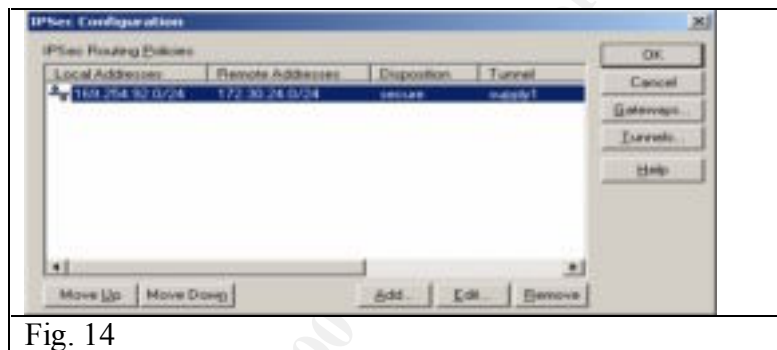


Fig. 14

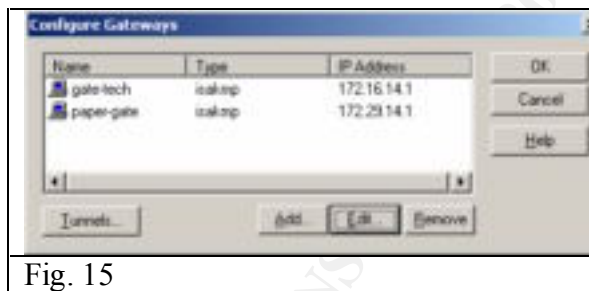


Fig. 15

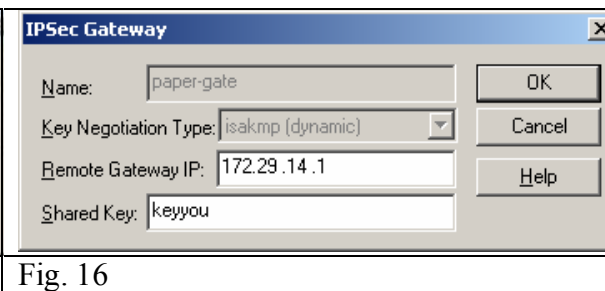


Fig. 16

Next, a tunnel is defined by clicking the Tunnels button in IPSEC configuration (Fig. 14). Click Add in Configure Tunnels (Fig. 17) and select a gateway (Fig. 18).

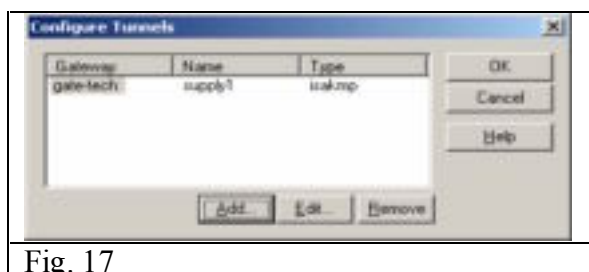


Fig. 17



Fig. 18

The tunnel needs a name in Configure Tunnel Identity (Fig. 19). In Dynamic security (Fig. 20), the Security association proposal type of Encapsulated Security Payload (ESP) is selected along with the authentication type of SHA1 and encryption type 3DES. The key expiration policy is checked and defined as every 8192 Kb, every 24 hours.

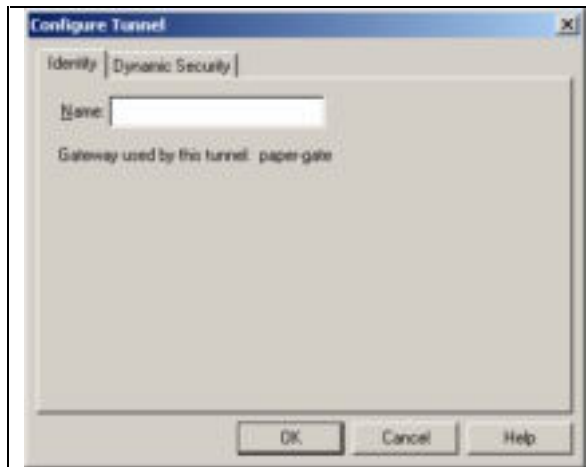


Fig. 19

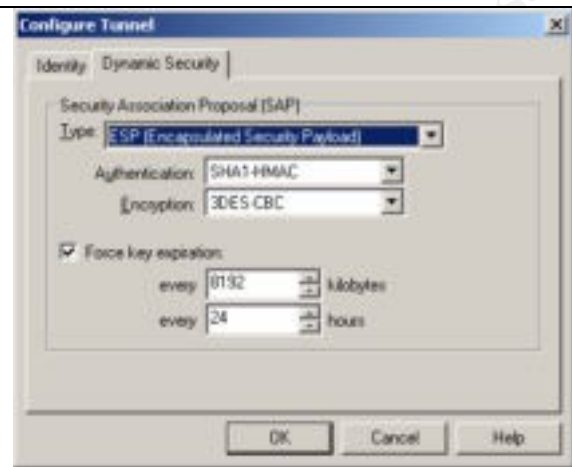


Fig. 20

The remote gateway's VPN settings need to match those of GIAC's gateway for the tunnel to be enabled. Routing tables specify the Firebox as the gateway to the remote network on both ends for hosts to begin using the tunnel.

The primary firewall and VPN device in this configuration takes advantage of a simple but powerful system to manage the different access controls and VPNs required. The Firebox has proxies built in for HTTP/S, SMTP, FTP and other protocols. They tend to have minor problems. There is also an authentication system allowing creation of local users for access control or using with Windows NT, radius or secure ID for authentication. This uses a Java applet for login, which can be cumbersome and buggy. Thus they are not implemented by GIAC.

Squid proxies

GIAC will use two Squid proxy caches. One for all user level outgoing traffic and one as a front end to the public HTTPD servers in "reverse proxy" or web accelerator mode. Both run on [Open BSD 2.8](#). Squid, which can be downloaded from <http://www.squid-cache.org>, is an open source high-performance proxy-caching server for web clients, supporting FTP, gopher, and HTTP data objects. It runs on *nix based systems.

The following features of Squid will be focused on for this implementation:

- HTTP Access control lists and content filters
- HTTP Proxy and Reverse proxy or Web acceleration
- Logging

The Squid software is configured in the squid.conf file usually located in /usr/local/squid/etc/squid.conf or /etc/squid/squid.conf. Both Squids will be set to run as user squid on their systems. This is achieved by creating the user Squid and adding the following lines to the squid.conf:

```
cache_effective_user squid
cache_effective_group squid
```

The `cache_dir /var/squid/cache/ 100 16 256` tag sets the directory that will hold cached data. In this example the directory is `/var/squid/cache`. The cache size is 100MB of disk space in 16 directories with 256 sub-directories each.

Before running Squid the first time, the cache directory structure needs to be created, rights set, and Squid needs to be initialized.

Squid initialization:

```
squid-out: # mkdir -c /var/squid/cache/
squid-out: # chown -c squid:squid /var/cache/
squid-out: # chmod -c 770 /var/cache/
squid-out: # /usr/sbin/squid -z
1999/06/12 19:15:34| Creating Swap Directories
skystone: #
```

Squid has been instructed to create the directory structure needed for the cache and exit with the `-z` parameter. It now can be run normally, after setting up access lists, acceleration settings and proxy settings.

Squid's ACLs operate from the top down. When a match is found, the specified action is performed. Support for external authentication programs is included. Rules can be based on time of day, regular expressions, protocols, and method-based rules. In the absence of rules, Squid will allow everything.

There are two parts to a Squid ACL. The ACL definition, which assigns a name, and holds the descriptor based on type such as domain; and the access tag, which, combined with a list name, performs an action such as permit or deny.

To define an ACL use the syntax:

```
acl aclname acltype string1 string2
```

An "aclname" is an arbitrary string used to refer to the ACL when defining actions. An "acltype" can be:

```
src          source IP address
dst          destination IP address
srcdomain    source domain(reverse lookup)
dstdomain    destination domain (from URL)
url_pattern  URL based pattern matching
urlpath_pattern  pattern matching in the URL path
port         TCP port
proto        Protocol such as HTTP or FTP
method       HTTP GET or POST,
browser      regexp match on user-agent-header
user         Ident the user
time         Day-SMTWHFA hour:min-hour:min
```

The following ACL definitions access tags are required to define action for the ACLs. If no access lines exist then all access is allowed. If a request traverses the entire list without a

match, then the opposite of the last rule is performed. If the last rule is deny, then all items not covered in the rule list are permitted.

After defining ACLs, actions need to be defined. Actions are allow or deny for http_access, ftp_access, snmp_access and other supported protocols. An exclamation point can be used for “not.” Action syntax is:

```
http_access allow|deny [!]aclname ...
```

This example allows the client to access the destination domain of giac.com and denies all other access.

```
acl example dstdomain .giac.com
acl all src 0.0.0.0/0.0.0.0
http_access allow example
http_access deny all
```

Reverse proxy (accelerator mode)

The Squid by default listens on TCP port 3128. This will be changed to TCP port 80.

http_port 80	Sets Squid's listening port to 80
http_accel_host 192.168.11.28	Tells Squid to accelerate www.giac.com
http_accel_port 80	Tells Squid that the accelerated server is listening on port 80
http_accel_with_proxy off	Tells Squid not to act as a standard proxy cache along with being a reverse proxy.
http_accel_uses_host_header on	Tells the cache to pass HTTP 1.1 host headers for servers hosting multiple sites (virtual servers)

Here is the complete access list including the Squid defaults

Default acls: acl all src 0.0.0.0/0.0.0.0 acl manager proto cache_object acl localhost src 127.0.0.1/255.255.255.255 acl SSL_ports port 443 563 acl Safe_ports port 80 21 443 563 1025-65535 acl CONNECT method CONNECT #my rules acl PUT method PUT acl rev_proxy dst 169.254.92.18/255.255.255.255 acl rev_proxy dst 192.168.11.28/255.255.255.0 #Default access http_access allow manager localhost http_access deny manager http_access deny !Safe_ports http_access deny CONNECT !SSL_ports #my access	#Everyone #Manager access for cachemgr.cgi #Defines https method Connect #Defines http method Put #ACL for Squid #ACL for target #Allow manager locally #Disallow manager remote #Limit port access #Limit Connect to SSL ports
---	--

http_access deny PUT all	#Deny all http put requests
http_access allow rev_proxy	#Allow access to target
http_access deny all	# Deny everything else

To access the Giac.com web page a user would point his browser to the Squid, at [HTTP://169.254.92.18](http://169.254.92.18) and the Squid would deliver the page to the user. The user will not know they are using a reverse proxy. In DNS, the GIAC zone would point www.giac.com to 169.254.92.18 the external Squid.

Squid Proxy-Cache

For internal users who need access to external HTTP and FTP resources there is another proxy running on the internal network. The Squid performs proxy services for HTTP, HTTPS, FTP and logs all access by username to identify uses and abuses of Internet connectivity.

To perform the required functions, the Squid ACLs set HTTP and FTP access tags. Using an external authentication program, Squid will prompt users for a password to access web and FTP resources. Squid comes with several authenticators for Unix and there are a few for server message blocks and NTLM as well. The NTLM method is available from <http://squid.sourceforge.net/ntlm> and the SMB method is at http://www.hacom.nl/~richard/software/smb_auth.html.

To configure authentication, compile and install the desired authentication module. Add the authentication program tag to squid.conf. Create the user listing desired. GIAC uses ncsa_auth and reads its users and passwords from a passwd file on the local system.

authenticate program /usr/bin/ncsa_auth /usr/etc/passwd

Configure the rest of squid.conf :

<pre> http_port 3128 ftp_user giac@giac.com #Default acl's: acl all src 192.168.11.0/255.255.255.0 acl manager proto cache_object acl localhost src 127.0.0.1/255.255.255.255 acl SSL_ports port 443 563 acl Safe_ports port 80 21 443 1025-65535 acl CONNECT method CONNECT # -- begin my rules -- acl passwd proxy_auth REQUIRED #Default access: http_access allow manager localhost http_access deny manager http_access deny !Safe_ports http_access deny CONNECT !SSL_ports # -- begin my actions -- http_access allow passwd http_access allow localhost http_access deny all authenticate_program /usr/bin/ncsa_auth /usr/etc/passwd proxy_auth_realm GIAC Internet </pre>	<pre> #Set the listening TCP port for the cache #Set email address for anonymous login #Define all Net ID #Define manager protocol #Define localhost #Define SSL ports #Define Safe_ports #Define SSL Connect method #Require external auth #Allow manager locally #Deny remote manager #Limit port access #Limit Connect to SSL ports #Allow succesful logins #Allow local for testing #Explicit deny all else #Set auth binary #Define realm for login screen </pre>
--	---

The above Squid configurations will provide an additional layer of protection for network and services. Denying direct access to web servers provides additional protection for public hosts.

Security Policy Conclusions

Before implementing any ACLs or Proxy settings on a network border, it is imperative to understand each protocol that will be allowed and to document its particular requirements. Any time a change is implemented the inventory and documentation is updated.

A preliminary check should be made of each ACL upon implementation to verify it is operating as expected. This can be done with a variety of tools like ping, traceroute, Nmap, Hping, and firewall.

© SANS Institute 2000 - 2002, Author retains full rights.

Audit and Assessment

Goals

- Verify that perimeter systems are operating as expected and attempt to identify any that are not.
- Assess the level of protection provided for GIAC enterprises information systems by the border router and firewall systems.

Considerations

- Testing should be performed during non-peak hours.
- IT staff should be on hand to address any systems that are left in an unstable state by testing.
- Probing should not knowingly result in crashing of any systems.
- A final "attack phase" should take place during normal business hours to demonstrate to GIAC's staff what an attack looks like.
- Senior management will document on letterhead that they are aware of and approve of the planned assessment and agree not to hold the assessment team liable. They will also clearly state any limitations that will be placed on the assessment process.

Limits

- Social engineering will not be a part of this test. This audit is focused on border and public system configuration. A training program for information security awareness will be included in standard company training and periodic spot checks can be performed if deemed necessary.
- Modem sweeps will not be a part of the investigation. A modem sweep should be performed, but this audit is focused on border and public system configuration.
- No actual break in will take place. A demonstrated avenue of entry is sufficient.

Implementation

A three phased approach will be used as described in [Hacking Exposed](#) by McClure, Scambray, and Kurtz.

- Phase #1 Footprinting – Target Acquisition
- Phase #2 Scanning
- Phase #3 Enumerating

Phase #1 Footprinting – Target Acquisition

Phase 1 consists of non-invasive techniques to establish possible points of entry to a network. A good place to start would be a review of public records such as, Whois databases, Corporate pages on public web sites such as Yahoo!, and Edgar at <http://www.edgar-online.com/>. Also press releases about partnerships and alliances can point to avenues of approach using links to partnering companies networks.

Sample Whois output

```

bash-2.04$ whois giac.com

Whois Server Version 1.3

    Domain Name: GIAC.COM
    Registrar: NETWORK SOLUTIONS, INC.
    Whois Server: whois.networksolutions.com
    Referral URL: www.networksolutions.com
    Name Server: NS1.GIAC.COM
    Name Server: NS2.GIAC.COM
    Updated Date: 08-mar-2000
>>> Last update of whois database: Sat, 20 Jan 2001 11:50:32 EST <<<
Registrant:
GIAC Enterprises. (GIAC-DOM)
  6 Cookie Run
  Cookietown, NY 12345
  US
  Domain Name: GIAC.COM

Administrative Contact, Billing Contact, Technical Contact:
  Goldberg, Dan (DG1234) Postmaster@GIAC.COM
  6 Cookie Run
  Cookietown, NY 12345
  US
  123-456-7890

NS1.GIAC.COM   169.254.92.20
NS2.GIAC.COM   169.254.92.21

```

Whois provides a wealth of information. Performing Whois on the IP address reveals the ISP providing services also. DNS can provide another data goldmine:

```

bash-2.04$ nslookup
Default Server: DNS.Somewhere.com
Address: 123.45.678.9
> ls giac.com
[DNS.Somewhere.com]
*** Can't list domain giac.com: Query refused

```

By querying DNS using somewhere.com's DNS servers an attempt to list the domain is denied. By changing servers in nslookup a listing was collected demonstrating an improperly configured DNS server.

```

>server 169.254.92.20
Default Server: 169.254.92.20
Address: 169.254.92.20
> ls giac.com
[169.254.92.20]
  giac.com.                server = ns1.giac.com
  giac.com.                server = ns2.giac.com
  giac.com.                169.254.92.18
  smtp                    169.254.92.17
>exit

```

Ls listed GIAC's domain, the only hosts listed are publicly accessible hosts. This list provides plenty of material with which to start. Bind should have access lists to restrict zone transfers!

Phase #2 scanning

Scanning can take on any number of forms. In its most basic form a system tries to connect to a series of TCP or UDP ports on a system or group of systems, then records the responses. This is done to see what services are listening on those systems. Scanning is used to discover what services are available. Some listening ports can also positively identify certain Operating Systems and other vulnerabilities. TCP scans can use a complete three-way handshake with each port, perform a Syn and not respond to the Syn / Ack (half open) or mix up the flags in any order to see how the target responds.

Some obvious starting points with the data gathered from the phase 1 are the www and mail servers. Using Nmap we can scan against the obvious port and try to identify the server without triggering any alarms.

```
bash-2.04# nmap -sT -vOP0 -p 80,443 www.giac.com
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Initiating TCP connect() scan against www.giac.com (169.254.92.18)
The TCP connect scan took 36 seconds to scan 2 ports.
Warning: No TCP ports found open on this machine, OS detection will be MUCH
less reliable
Interesting ports on www.giac.com (169.254.92.18):
Port      State      Service
80/tcp    filtered   http
443/tcp   filtered   https

Too many fingerprints match this host for me to give an accurate OS guess
TCP/IP fingerprint:
T5 (Resp=N)
T6 (Resp=N)
T7 (Resp=N)
PU (Resp=N)

Nmap run completed -- 1 IP address (1 host up) scanned in 249 seconds
bash-2.04#
bash-2.04# nmap -sT -vOP0 -p 25 mail.giac.com
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Initiating TCP connect() scan against smtp.giac.com (169.254.92.17)
The TCP connect scan took 37 seconds to scan 1 ports.
Warning: No TCP ports found open on this machine, OS detection will be MUCH
less reliable
Interesting ports on smtp.giac.com (169.254.92.17):
Port      State      Service
25/tcp    filtered   smtp

Too many fingerprints match this host for me to give an accurate OS guess
TCP/IP fingerprint:
T5 (Resp=N)
T6 (Resp=N)
T7 (Resp=N)
PU (Resp=N)

Nmap run completed -- 1 IP address (1 host up) scanned in 236 seconds
bash-2.04#
```

The NMAP flags used are -sT for connect() scan, -v Verbose, -O OS fingerprint, -P0 do not ping, -p <port(s)>

The NMAP output does not reveal much. A simpler attempt might be to Telnet to the desired service and look for a response:

```
bash-2.04# telnet www.giac.com 80
Trying 169.254.92.18...
Connected to www.giac.com.
Escape character is '^]'.
http/1.1 get <enter> <enter>
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/5.0
Date: Thu, 25 Jan 2001 03:10:18 GMT
Content-Type: text/html
Content-Length: 87

<html><head><title>Error</title></head><body>The parameter is incorrect.
</body></html>Connection closed by foreign host.
bash-2.04#
```

Thank you Microsoft! Nmap may not be the best tool all the time! Trying the same technique on the mail server is less useful or is it?

```
bash-2.04# telnet mail 25
Trying 169.254.92.17...
Connected to mail.
Escape character is '^]'.
220 mail.giac.com ESMTTP Sendmail 8.11.2 Wed, 24 Jan 2001 22:12:24 -0500 (EST)
```

There is enough information to start to get a foothold in this network with Sendmail 8.11 and IIS 5.0. The IIS host has to be a Windows 2000 server, the Sendmail is likely a *nix.

Phase #3 enumerating

Enumeration involves the research of the specific exploit to be used and its execution. For the purposes of GIAC enterprises audit, once the research is done, steps are recommended to repair the exposure.

Confirmation and recommendations

A study of the site's logs after the assessment can shed some light on how things worked during the assessment phase. Did all the bells and whistles work? In this case the firewall is set to add hosts that perform certain actions to a dynamic blocked host list. TCP port 111 and odd flag groupings are such actions. Here we see them working. Port and address scans also do this but must be in numerical order to be caught, so mixing up the ports or address numbers is a way around this feature.

The logs collected from the firewall after the above scans:

20:22:33	deny	in	eth0	tcp	bad.host	169.254.92.16	16856	848	syn	(default)
20:26:51	deny	in	eth0	tcp	bad.host	169.254.92.16	17754	631	syn	(default)
20:26:57	deny	in	eth0	tcp	bad.host	169.254.92.16	17754	631	syn	(default)
20:29:43	deny	in	eth0	tcp	bad.host	169.254.92.16	17710	80	syn	(Incoming_HTTP)
20:29:49	deny	in	eth0	tcp	bad.host	169.254.92.16	17710	80	syn	(Incoming_HTTP)
20:30:01	deny	in	eth0	tcp	bad.host	169.254.92.16	17710	80	syn	(Incoming_HTTP)
20:30:25	deny	in	eth0	tcp	bad.host	169.254.92.16	17710	80	syn	(Incoming_HTTP)
20:35:33	deny	in	eth0	tcp	bad.host	169.254.92.16	44089	80	syn	(Incoming_HTTP)
21:37:57	deny	in	eth0	tcp	bad.host	169.254.92.18	40210	41462	syn	(default)
21:37:57	deny	in	eth0	tcp	bad.host	169.254.92.18	40212	41462	f in	psh urg
21:37:57	deny	in	eth0	tcp	bad.host	169.254.92.18	40206	80	rst	(blocked site)
21:37:57	deny	in	eth0	tcp	bad.host	169.254.92.18	40208	80	rst	(blocked site)
23:13:54	deny	in	eth0	tcp	bad.host	169.254.92.6	47701	6008	syn	(default)
23:13:54	deny	in	eth0	tcp	bad.host	169.254.92.6	28816	111	syn	(blocked port)
23:13:54	deny	in	eth0	tcp	bad.host	169.254.92.6	19411	253	syn	(blocked site)
23:13:54	deny	in	eth0	tcp	bad.host	169.254.92.6	1527	607	syn	(blocked site)

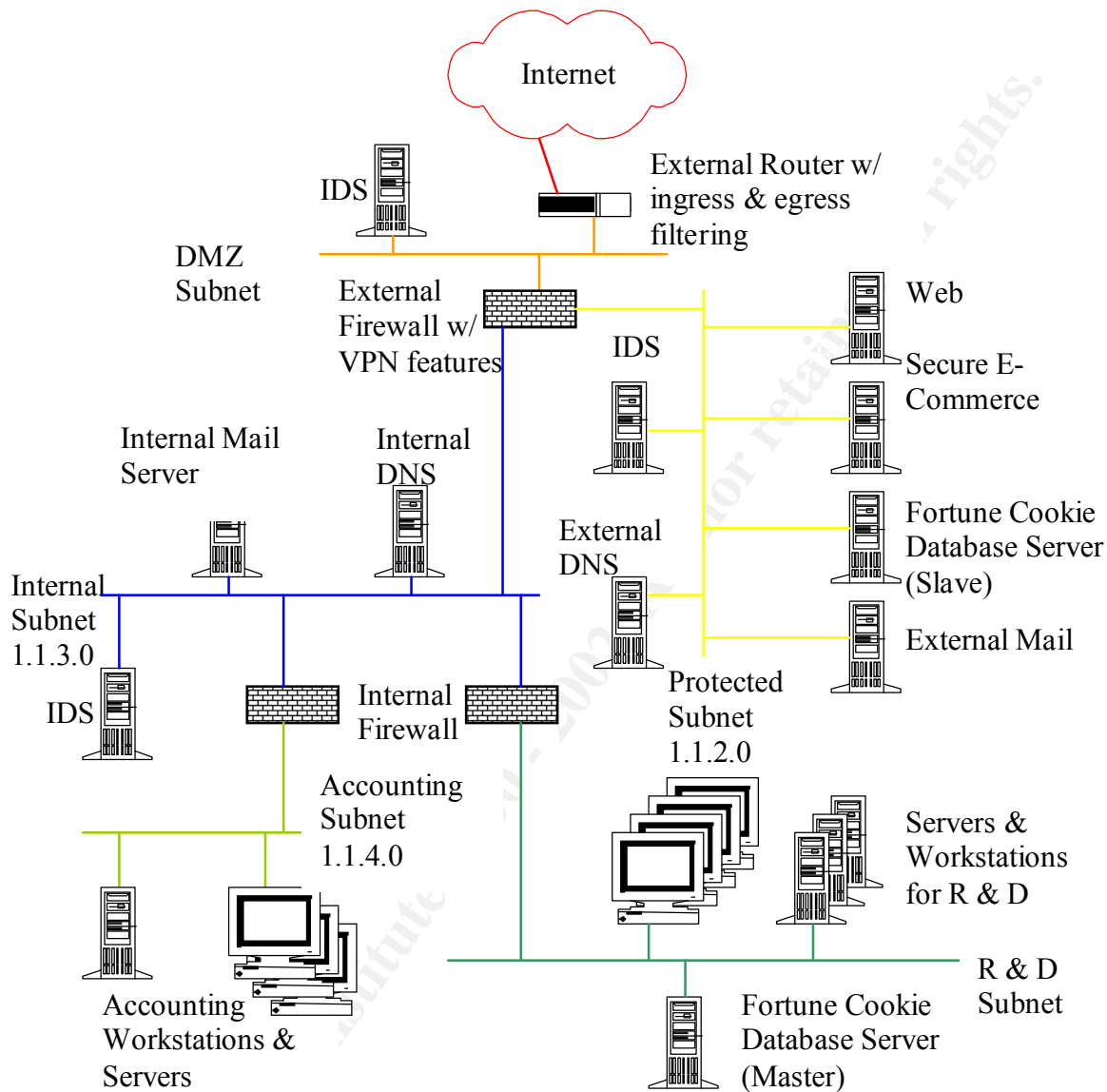
At 20:22:33 bad.host sent a TCP SYN to 169.254.92.16 on port 848 and it was denied sending a TCP RST. This used the default rule. The Incoming HTTP rule was used on port 80 and sent an RST.

At 21:37:57 a TCP segment with the FIN PSH and URG flags was received by the firewall, which triggered the system to add bad.host to the blocked site list. No RSTs were sent, all further packets were silently dropped. Later at 23:13:54, bad.host tried to connect with the SUN-RPC port 111 and again was added to the blocked site list. The administrator sets the blocked site duration.

After completion of phases 1 through 3, the auditor will have a pretty good idea of what he thinks the network looks like. The administrative and security teams will meet with the external auditor. Actual maps of the network and maps created during the audit will be compared. A list of vulnerabilities will be drawn up and a list of recommendations will be made to improve the perimeter. This process will be an ongoing with internal administrative and security teams reviewing the network and current activities on the Internet to keep systems current.

Design Under Fire

Practical Used: http://www.sans.org/y2k/practical/Dean_Denter.doc



(All subnets are assumed to have a 255.255.255.0 subnet mask)

Assumptions

We have undertaken the task of infiltrating GIAC. Until this point we have limited ourselves to footprinting, scanning, and enumeration activities. We feel we have built a reasonable map of which hosts exist on the network from Whois output, DNS output and public records. Attempts to map the DMZ and screened subnet have led to partial success using Nmap and Nessus type scanners. We have elected to target the firewall for additional mapping information, denial of service attacks, and finally, to compromise one or more internal systems starting with DNS. The choice to use DNS coincides with the ease of establishing its existence and address.

Attack the Firewall

Checkpoint FW-1 with certain configurations can be attacked when the fast mode command has been used. After locating a listening port on the firewall, it is possible to create unauthorized connects with hosts behind the firewall by sending fragmented TCP packets to the protected host. A demonstration of this vulnerability has been created by Thomas Lopatic <lopatic@tuv.net> and is available for download at:

<http://www.securityfocus.com/data/vulnerabilities/exploits/fm.c>

An attack like this relies on systems that are not kept up to date in order to succeed. FW-1 4.1 service pack 3 supposedly fixes this problem if fast mode is in use. Turning off fast mode also defeats this issue.

FW-1 fast mode will not be included in future versions. It is a feature that stops checking data in a TCP connection following a successful three-way handshake. This was supposed to improve performance for TCP conversations. This attack will fail since GIAC's administrators are on the ball.

DOS the firewall and explain countermeasures

The selected firewall for the above network is Checkpoint's Firewall-1, Version 4.1 Service pack 3, with relevant patches applied. One choice for a denial of service is to use a fragmentation attack. The Jolt2 tool is a simple enough method to effect this type of attack

According to Bugtraq ID 1312 and [CVE-2000-0482](#), illegally fragmented packets destined for a FireWall-1 host or a host behind it will force the firewall to use 100% of available processor time. There are no rules that will prevent this form of abuse.

Fragmentation normally occurs in TCP/IP when an IP datagram crossing from one network to another exceeds the maximum transmission unit (MTU) for that network. If the DF (don't fragment) bit is set, the router drops the datagram and returns an ICMP error with the MTU as its data.

When the DF bit is not set, the router will break the segments into smaller segments, copy (from memory) the IP header and ID on to the fragments with the MF bit (more fragments) set, generate the offset, and send them on their way. The last fragment does not have the MF bit set.

By using a tool like Jolt2 to create large fragments, the firewall tries to reassemble them for inspection and in the process consumes all CPU resources creating a DOS. A suggested workaround is to disable the console logging by typing the following command at the FireWall-1 module(s):

```
$FWDIR/bin/fw ctl debug -buf
```

More information on this DOS has been published at:

http://www.checkpoint.com/techsupport/alerts/ipfrag_dos.html by Checkpoint.

This vulnerability was discovered by and posted to Bugtraq on June 6, 2000 by Lance Spitzner <lance@spitzner.net>.

To repair this situation, after the firewall has been brought to its knees with fragmentation attacks coming from many sources, GIAC's security and administrative team will take the firewall offline and disable console logging. They will bring it back online and contact Checkpoint to see if there are any new patches or data on this DOS.

Compromise an internal system

On January 29th 2001 [CERT released multiple vulnerabilities](#) in ISC's BIND. The External DNS server and the Fortune Cookie database live on the same network segment in the targets screened subnet. Presumably there is a process that lives in the corporate network that takes customer input off the database slave and updates the main database. Taking control of BIND is a great stepping stone to owning the database server, which is likely to have access to and information about the main database server. The TSIG <http://www.kb.cert.org/vuls/id/196945> buffer overflow should grant that access unless administrators have had opportunity to update BIND to Version 8.2.3. Other opportunities to attack Bind involve the NXT overflow using any number of scripts available for download from your favorite Black Hat web site.

Once a foothold is established in the DNS system, local access to the network segment becomes available. One next step would be to load a sniffer to grab passwords and analyze traffic patterns. Getting local access into the web server and the database grants the ability to determine passwords for the databases. Presumably the master database trusts its slave and that relationship can be used to target this host and gain access to the internal network.

One way to improve this configuration is to split the screened subnet into multiple subnets. Place the public web server, DNS and the mail relay on one subnet, since these are the most vulnerable services. Place the E-commerce server and the slave database on a second subnet. Set the ACLs between these subnets to be restrictive only to the services required between them, if any. This will reduce the exposure of running high-risk services (SMTP, DNS, and HTTPD) on the same segment as critical services from sniffing attacks and from attacks that use the exposed hosts as starting points. It is also best to shut off all unneeded services on all hosts as usual.

References

- [Improving Security Cisco Routers](http://www.cisco.com/warp/public/707/21.html) <http://www.cisco.com/warp/public/707/21.html>
- [Building Bastion Routers Using Cisco IOS](http://www.attrition.org/~modify/texts/phrack/Phrack55/P55-10)
<http://www.attrition.org/~modify/texts/phrack/Phrack55/P55-10>
- “Essential IOS features Every ISP Should Consider”
<http://www.cisco.com/public/cons/isp/documents/>
- [A Stateful Inspection of Firewall-1](http://www.wittys.com/fw-1/Blackhat-8.pdf) <http://www.wittys.com/fw-1/Blackhat-8.pdf>
- Packetstorm <http://packetsorm.securify.com>
- DNS Security http://www.softpanorama.org/Security/dns_security.shtml
- Dean Denter’s GCFW practical http://www.sans.org/y2k/practical/Dean_Denter.doc
Thanks Dean!
- Cisco ACLs http://www.sans.org/infosecFAQ/firewall/blocking_cisco.htm
- For an introduction to Cisco IOS configuration, visit Cisco’s web site at http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgr/fun_c/index.htm
- The Squid FAQ located at <http://www.squid-cache.org/Doc/FAQ/FAQ.html>
- The Squid user’s guide located at <http://squid-docs.sourceforge.net/latest/html/book1.htm>
- [Rfc2267.txt, Network Ingress Filtering](ftp://ftp.isi.edu/in-notes/rfc2267.txt), RFC available at <ftp://ftp.isi.edu/in-notes/rfc2267.txt>.
- How to audit your network <http://www.enteract.com/~lspitz/audit.html>

Further Reading

- “Cisco IOS essentials” by ALBRITTON, McGraw Hill
- [Building You Firewall Rulebase](http://www.enteract.com/~lspitz/rules.html) <http://www.enteract.com/~lspitz/rules.html>
- "DNS and BIND" 3rd edition Lou, Albitz, and Loukides Orielly
- "Building Internet Firewalls" Zwicky, Cooper, and Chapman, Orielly