SANS Institute
InfoSec Reading Room

Performing Egress Filtering

Performing Egress Filtering

GCFW Gold Certification

Author: Dennis Distler, distlerdennis@gmail.com

Advisor: Joel Esler joel.esler@mac.com

Accepted: August 19, 2008

Outline

1. Introduction

Most of today's security administrators employ inbound filtering on Internet facing firewalls. The best practice for inbound filtering, also known as egress filtering, is to access connections for Internet services required for business purposes only. Although this practice is necessary to survive on the Internet today, security administrators sometimes miss a great opportunity to filter traffic leaving their network.

This paper is intended for anyone responsible for firewall management in their organization. The reader should have a working knowledge of TCP/IP communications, firewalls and Linux.

The purpose of this paper is to explain egress filtering and the risk that can be mitigated along with it. The paper will discuss the types of default policies while also comparing and contrasting these policies. The next section will describe the egress filters role in an organization's security policy. The paper will then discuss possible issues with egress filtering, along with component used in egress filtering including syslog servers, proxy servers, routers and firewalls.

The paper will describe how to gather requirements for egress filtering. This section will discuss who to talk to, what questions to ask, as well as tools to use to capture traffic to provide information for the egress filter.

Once the requirements are gathered, a fictional company named GIAC Rock will have egress filtering designed and implemented. The business requirements will be discussed, followed by GIAC Rocks security policy.

Next, the design that was completed in the previous section will be implemented. This section will show the rulebase applied to various filtering

device and the configuration of a syslog server to be used for monitoring.

After the implementation there will be issues and egress filtering will need to be monitored. This section will discuss what and how to monitor after implementing egress filtering.

2. Reasons For Performing Egress Filterin g

Egress filtering is defined as ensuring that your site does not emit packets from inappropriate addresses (Cheswick, Bellovin, & Rubin, 2003). Too often firewall administrators perform ingress filtering, but fail to perform egress filtering (SANS,2007). By not performing egress filtering firewall administrators put their organizations at risk.

Today there are two types of risk that must factor into a security professionals decision, business and technical risk.

Egress filtering can reduce risk to an organization. One risk is the loss of employee productivity. According to spectorcne.com (Spectorcne, 2008), on average $6250 per employee is lost to Internet abuse. For an organization of 50 employees, this translates to a loss of over $310,000 a year in productivity. Egress filtering alone can reduce Internet abuse by limiting access to certain applications. By adding content filtering with egress filtering, the business can filter websites unrelated to business needs, therefore improving employee productivity.

Another risk posed to an organization is litigation. For example, if ABC organizations system becomes compromised, ABC's system can be used to attack XYZ

organizations system. When this type of attack occurs, it is possible that XYZ brings litigation against ABC for loss of revenue. With litigation there is the risk of damage to the integrity and image the organization. Damage to integrity and image of an organization may be hard, if not impossible to overcome.

Another risk associated with not filtering traffic is bandwidth abuse. If an organization performs egress and content filtering, bandwidth can reduced, saving the organization in telecommunication money. For example, GIAC Rock has 500 locations and each location has a full T-1 (1.5 MB) at the cost of $850 a month. The cost to GIAC Rock is $425,000 a month. With the implementation of egress and content filtering, GIAC Rock is able to cut each full T-1 to a fractional T-1 (1.0 MB) at a cost of $700 a month. This would reduce the monthly telecommunications cost to $350,000 a month. That is an annual savings of $900,000 to GIAC Rock.

One technical reason to perform egress filtering is attack prevention and Internet etiquette. By performing egress filtering, most attacks from an organization's network can be prevented. If an individual connects a laptop loaded with malware, egress filtering could prevent malware from leaving the organizations network.

Another technical reason to perform egress filtering is to identify mis-configured devices on the network. Almost daily, the author sees new equipment brought on the network that is mis-configured. Two of the most common mis-configurations is SNMP enabled going to a non-existent monitoring server or a printer installed on a workstation that is configured to monitor the printer. The traffic is pointless and not needed. Mis-configured systems are easily identified

when monitoring firewall logs.

Although this paper discusses egress filtering on Internet gateways, egress filtering can be performed internally. The same tools and methods described in this paper can be used for internal egress filtering.

3. Egress Default Policy Types

We keep 0.1 inches space before and after each paragraph. The title of the paper will be used as header, the author and a page number is used as footer.

Before implementing egress filtering the default egress policy decision must be made. There are two types of egress filter policies that be deployed: default-allow and default-deny. Your organizations needs will drive which default egress policy will be deployed. However it is strongly recommended that a default deny policy is selected.

The default-deny policy offers advantages over the default-allow policy but it does have disadvantages. One advantage is that no traffic is permitted to leave the network unless the security administrator explicitly allows traffic out of the organizations network. (Egress Filtering, 2008) This is the most secure method of egress filtering. The downside to this is the administrative overhead. Whenever a system or application requires an outbound connection, an egress filter change must be made. (Egress Filtering, 2008)

The default-allow policy offers advantages over the default-deny policy but, it too, has its disadvantages. One advantage is that all traffic is allowed to leave the network unless specifically blocked by the egress filter. Although this

policy does not require as much administrative overhead as default-deny, the downside leaves the policy less secure (SANS, 2008). If default-allow policy must be used, Chris Brenton's Egress Filtering FAQ has an excellent section on what ports should be denied.

Although egress filtering is enforced by a firewall, the decision as to what default egress policy will be implemented should be made by only the highest level of IT Management. It should be noted, that even if management decides to use a default-allow policy, after a period of time, the security administrator will be able to show the effects of egress filtering. It might be possible to have management change the default policy to a default-deny policy to better reduce the risk to the organization.

4. Egress Filtering Role in Security Policy

Egress filtering should be used to reduce the overall risk to an organization. To accomplish risk reduction the organization's security policy should be consulted. After implementing egress filtering, the organizations security policies, processes, and procedures should updated with the change to the security posture.

5. Egress Filtering Issues

Before implementing egress filtering, there are some issues that must be described. These issues may be technology specific, ranging from how proxy servers deal with unsupported protocols to general egress filtering avoidance by using encryption.

The first issue with any type of filtering is the risk of filtering avoidance. Filtering avoidance is the ability for a user to send traffic through the filtering system that is not permitted to pass through the filter. Many of today's applications have this ability to bypass egress filtering built in the application. For example, look at any Instant Messaging application. Most firewalls today provide methods to ensure only the correct type of traffic is permitted. Intrusion Detection and Prevention Systems (IDS/IPS) can also be used to mitigate non-standard traffic using open ports. The ability to bypass filters based on port and protocol in the author's opinion the biggest issue with egress filtering.

When implementing filtering, the issue of encrypted traffic must be addressed. This is one of the biggest challenges a security administrator will face. Users can use encryption to avoid traffic filtering. However, users may also use encryption as part of their work. The question now becomes how can encrypted traffic be filtered? One possible solution is to require all encrypted traffic leaving the organization to pass through an SSL bridge so that the traffic may be inspected and filtered as needed.

When performing egress filtering one of the components that may be used is a proxy server. Proxies are firewalls that examine the entire packet to ensure compliance with the protocol (Northcutt, Zeltser, Winters, Fredrick, & Ritchey, 2003). Although proxy servers are secure by nature there are two issues the reader should be made aware of.

The first issue deals with the port the proxy server will listen on. It is very common for a proxy server to listen on TCP 8008 or TCP 8080. For the most

part this configuration works without incident. However, an issue may occasionally arise is when the web site a user is attempting to access listens on the same port as the proxy server.

The second and harder issue to deal with is unsupported protocols. By looking at the list of protocols from Wikipedia (Protocols, 2008), the reader can easily see how this is an issue. The default behavior of most proxy servers is if the protocol is not supported, the protocol will be denied.

6. Requirement Gathering

Before gathering the business requirements for egress filtering, the default policy must be determined. Again this must be a decision made by IT management as to what the organizations default egress policy will be.

Once the default egress policy is determined it is time to overcome one of the biggest hurdles in egress filtering: who to talk to about the business applications. The first piece of information required is to identify which business applications are in use. Hopefully, the organization maintains a list of all business applications. If such a list does not exist, the best person to talk with will be the person in charge of business applications. This individual should be able to provide you with the following three critical pieces of information:

- Name of the Application
- Application/System Owner

- Application/System Administrator

It is important to understand, unless the organization is a small organization, the Application/System Owner and Application/System Administrator may be different.

The next step is to briefly meet with the application/system own and explain that the organization will be implementing egress filtering to better improve the overall security of the organization. Since most application/system owners are high level non-technical business individual's, this meeting should be keep short and high level. If the application owner "owns" several applications be sure to include all applications in one meeting.

After meeting with the application owner it will be time to meet with the application/system administrator. This meeting will be technical and where you will start to gather the requirements for the egress filter. One key item of note is to make it clear to both the application/system owner and applications/system administrator that egress filtering intent is not to break the applications, rather increase the security of the organization.

The application/system administrator should be able to provide the answers to the following three questions:

- Does your application require Internet access?
- What protocols (and ports) need to be allowed to the Internet?
- What are the applications destinations?

If the application/system administrator cannot provide you the answers to all

of these questions there are several ways to find out the required information.

Be sure that the question/answer to be asked of the application/system administrator makes sense. For example, if you were to ask the Windows XP administrator the destinations for Internet Explorer, you would probably get a puzzled look. The answer to this question be obvious in that the destination is the Internet.

The first method is to merely block all traffic and observe log files. Although usually the simplest method it is not recommended. This method has a good chance of breaking applications and having adverse effects on the business, as well as possibly endangering the reader's career.

The second method is to use sniffing application such as tcpdump/windump, or Wireshark to capture traffic. Either application will work for this purpose and is ultimately up to the reader. The author usually captures the traffic via tcpdump and the reviews the traffic with Wireshark.

When capturing traffic, the best approach is to write a filter to capture the packets required for analysis. For example, if you are monitoring a database server name DB01 with an IP address of 10.10.10.123, a simple filter to capture all traffic from DB01 to anything that is not on the internal network of 10.0.0.0/24 would be:

```
tcpdump -i eth0 ip src host db01 dst !10.0.0.0/24
```

This filter will capture all traffic from DB01 to any network not on the internal network. One nice by product of this filter is identification of mis-

configured systems. For example, if DB01 is sending Simple Network Management Protocol (SNMP) traps with the community of PUBLIC to 192.168.1.1, this traffic will be captured. Not only can risk be reduced to the organization, but unnecessary network traffic can be eliminated.

7. Components Used in Egress Filtering

Before describing the roles various network devices can play into egress filtering, it is important for the reader to understand different types of filtering technologies.

Packet filtering - These devices rely solely on the IP headers of individual packets to permit or deny traffic. A packet filter will look at a combination of source and destination IP address, source and destination ports, and traffic direction (inbound and outbound) (Northcutt, Zeltser, Winters, Fredrick, & Ritchey, 2003). If the packet matches the filter rules the traffic is allowed to pass.

Stateful Inspection - These devices create a table with state information about every established connection (Vladimirov, Gavrilenko, & Mikhailovsky, 2005). Also some application layer information is also inspected (Northcutt, Zeltser, Winters, Fredrick, & Ritchey, 2003).

Application Filtering - These devices inspect traffic at the application layer. The content of the traffic is inspected as well. (Application Layer Firewall, 2008)

When creating an egress filtering strategy, there are several components that

may be utilized. Some components will have active roles in egress filtering, while others will have support roles.

Before describing the roles various network devices can play into egress filtering, it is important for the reader to understand different types of filtering technologies.

Syslog Server

Syslog is used as the central logging solution. The syslog server should have the ability to accept log files from all egress filtering components.

The information stored on a syslog server is critical information to events happening in an organizations network. Therefore this server becomes a high value target for anyone trying to cover their tracks. Exceptional care should be given to the configuration and hardening of this server. If the attacker takes control of this server, nothing in the log files can be considered accurate.

Proxy Server

A proxy server acts as an intermediary in all interactions of a given service type (FTP, HTTP, etc.) between internal hosts and untrusted/external hosts. (Bauer, 2005)

A proxy server can be implemented as either an Internet gateway, content filtering server or a combination of both.

A Proxy server filter's at the application layer, which makes proxies by default the most secure firewalls available. Because of this, performance may be an

issue.

Proxies are commonly deployed to filter traffic for known protocols such as HTTP and use additional software to perform content filtering. It is important to note that a proxy can also deployed as a reverse proxy to better protect Internet facing systems.

Internet Router

When connecting an organization to the Internet, a router is typically used for the Internet connection. When performing egress filtering on a router, there are two options: using ACL to block traffic or use Black-hole routing.

Because the router is the central point that connects an organization to the Internet this is a great place to perform filtering. Routers can act as a simple packet filter, and most have some form of stateful packet inspection. It is important to remember that the routers main function is to route traffic, not filter traffic.

If your organization uses only a router to protect themselves on the Internet, the defense of the router is absolutely critical. Great considerations should be given to purchasing or building of a firewall to better protect the organization.

Black-hole routing refers to a place in the network where incoming traffic is silently discarder without informing the source that the traffic is discarded (Black Hole Routing, 2008) Black-hole routing can be used to stop denial-of-service (DoS) or distributed-denial-of-server (DDoS) attacks from being performed against

your organizations network. Black-hole routing can be set up to remotely trigger black-hole routing, as outlined in this excellent document from Cisco (Cisco, 2008).

The Internet router will be the last line of defense in egress filtering. When performing egress filtering on the Internet router, simple packet filtering should be performed. The filter should allow out only the public IP address space of the organization. All other packets should be logged and denied.

Firewall

The perimeter firewall should be the central egress point in a network design. It is acceptable to have multiple egress points in a network, as long as each egress point passes through a firewall. Because the firewall is the central egress point the firewall's role in egress filtering is critical.

The egress filter rulebase on the firewall will vary on the organization's needs. If using a default-deny policy, only open the ports required and if possible limit the destinations for the open port. All rules in the rulebase should only allow the source IP address of internal networks.

Normally a firewall will also be a Network Address Translation (NAT) device that hides the internal address of the network outside the organization. Although used to converse address space on the Internet, some do consider NAT a security feature. *Do not* rely solely on NAT to provide security.

Network Intrusion Detection System

A network IDS (NIDS) typically is deployed to give insight into network traffic. Most NIDS can be used as a sniffer, a packet logger, or an intrusion detection system. Using a NIDS in egress filtering can be beneficial, however there are drawbacks with NIDS. The benefits include capturing network traffic to be used for troubleshooting issues or monitor for possible security incidents in the network.

Using NIDS as a packet logger, the NIDS could provide an explanation why egress connection attempts are failing. For example, if an application starts a session using HTTP on TCP/80, the user clicks on a link that redirects the user to a web server listening on TCP/8008 the packet is denied at the firewall. Using a NIDS as a packet logger this traffic could be captured for examination. Using this information the egress filtering policy on the firewall can modified to allow this type of connection out the firewall.

Using NIDS as an intrusion detection system, any system that attempts an attack could possibly be detected if the attack is seen by the NIDS. For example a laptop is taken home overnight, the next day brought back onto the network. While this laptop was offsite it became infected with the Storm worm, a NIDS could alert the security administrator that there is an infected machine on the network. The IDS alert coupled with an increased denies at the firewall would allow the security administrator to proactively respond to the incident and possibly reduce the damage to the network.

Although there are benefits to using a NIDS in egress filtering, there are disadvantages that need to be addressed. The issues with NIDS are not just related to egress filtering, but exist in all NIDS deployments. The first

disadvantage is the amount of data a NIDS collects can be overwhelming. The data collected can have large data storage requirements. Another disadvantage is parsing the data collected by a NIDS. Tools can be used to reduce the amount of data to be analyzed, but the data must be analyzed.

There are many components that can be used in egress filtering. Not all components are required or needed depending on the organizations need and size. When implementing egress filtering use a defense-in-depth philosophy to protect the organization.

8. GIAC Rock's Design

Before implementing egress filtering gather the business requirements including what will be the default egress filtering policy and what services will be needed for the business to continue operations.

The first policy decision to make is what will the default egress policy be? GIAC Rock IT management decided their default policy is default deny. Although the default-deny policy has the greater chance of breaking applications, it is a greater risk reduction to the organization.

With the default policy defined, the next step is to determine what services must be allowed out of GIAC Rocks network. By talking with IT personnel, application owners, application administrators, and end users, GIAC Rock IT management has determined that the following services must be allowed out for the business to continue to function:

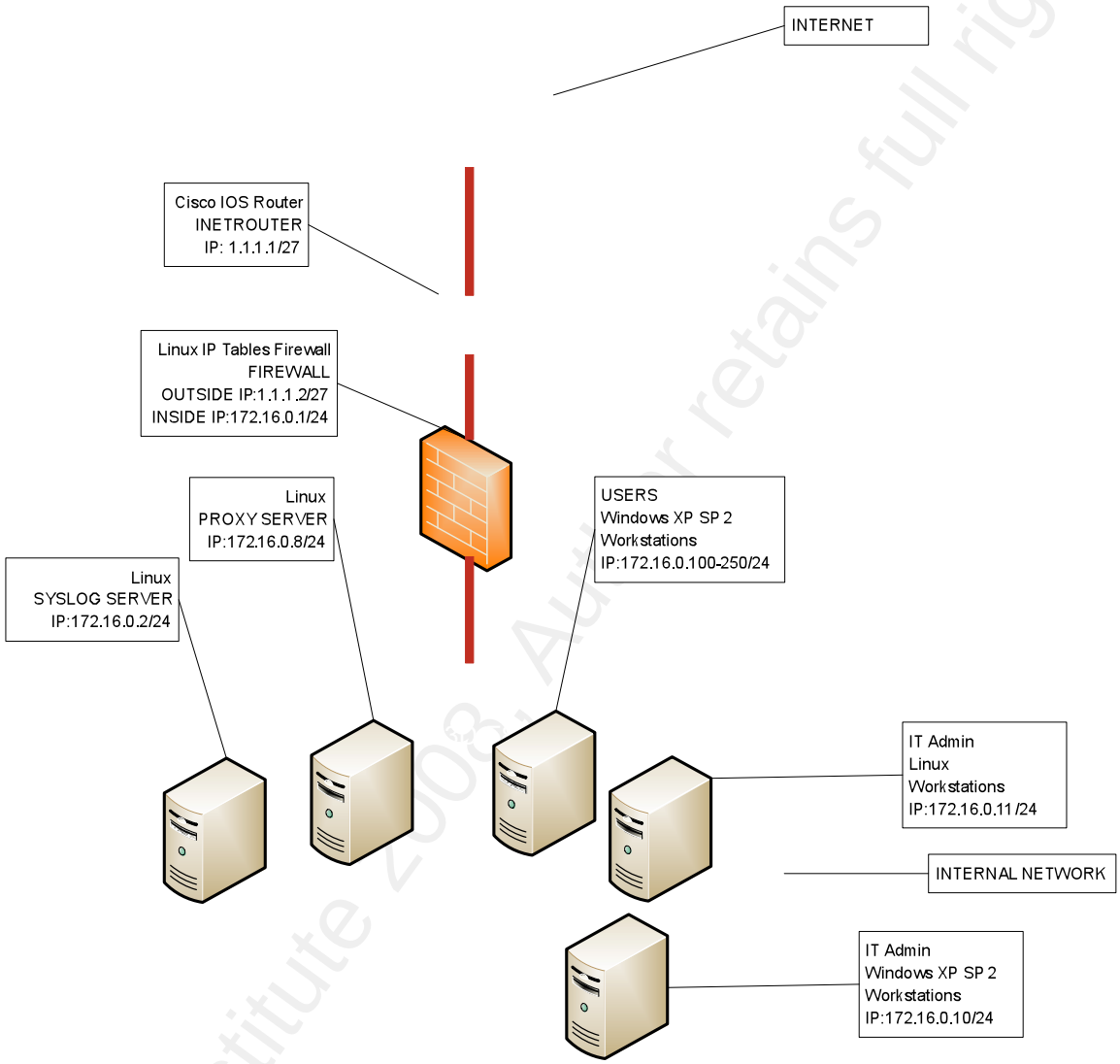
- Internet browsing

- Email
- File transfer application
- Name resolution
- Time synchronization
- Network diagnostics
- Windows operating system updates
- Linux operating system updates
- Easy Rock Making Updates (Custom Applications)
- Rock Sorting Updates (Custom Applications)

These are very general services that are required for the business to function. Other common services are SSH, Citrix, Blackberries, VPN, various software updates, custom applications, and third party proprietary applications that may require Internet access.

An organization's security policy may place constraints on egress filtering that must be taken into consideration.

With the business requirements gathered, it is time to implement the egress filtering strategy. Since business requirements are for content filtering a proxy server will also be implemented. To assist with the implementation, the following network diagram was created:



9. GIAC Rock's Implementation

With the completion of the design, it is time to implement egress filtering. Before implementation, the author has hardened all systems to reduce the risk of

compromise.

GIAC Rock's business requirements were put into the following matrix to assist with implementing egress filtering:

	IT Win	IT Linux	DNS	Yum	NTP	SMTP	All	Proxy	DC1
Internet								X	
Email						X			
FTP	X	X							
DNS			X						
NTP					X				
ICMP	X	X							
Windows Update									X
Linux Update				X					
Easy Rock							X		
Rock Sorting							X		

The above systems are assigned the following IP addresses:

Syslog server - 172.16.0.2

DNS server - 172.16.0.3

YUM server - 172.16.0.4

Windows server - 172.16.0.5

NTP server - 172.16.0.6

SMTP server - 172.16.0.7

Proxy server - 172.16.0.8

Windows IT workstation - 172.16.0.10

Linux IT workstation - 172.16.0.11

Before implementing egress filtering the syslog and proxy server must be configured. The first network device to have egress filtering applied will be the Internet router, and finally egress filtering will be implemented on the firewall.

Syslog Server

Syslog server allows for centralization of log files that can generate an overall view of egress filtering. The information contained on the syslog server can assist when troubleshooting an egress filtering issue. It can also be used to detect possible incidents in your organization.

After hardening the system, it is time to configure syslog to access messages from remote devices and log messages to /var/log/firewall. Syslog does not provide any type of access control, so another form is required. For this paper, iptables will be used provide access control to the syslog server.

First the syslog daemon needs to be configured to access syslog from remote systems. To ensure that syslog will always listen for remote connections, edited

the syslog.conf file located in the /etc/ directory. Using a text editor, change the line that reads SYSLOGD_OPTIONS to this:

SYSLOGD_OPTIONS=" -m 0 -r"

The -r syslogd option tells syslog to listen for remote connections.

After saving the change restart the syslog service. Next, configure iptables to accept connections on UDP 514 from source IP address of the remote system that will be logging to the central logging server. For GIAC Rock syslog server, the following iptables command will accept syslog messages from the firewall:

iptables -I INPUT 3 -i eth0 -p udp -dport 514 -s 172.16.0.1/32 -d 172.16.0.2/32 -j ACCEPT

One final step that is not required but highly recommended, is adding the host name's of a remote device's to the syslog server hosts file.

The remote device, in this example is the firewall, must be configured to send syslog messages to the central syslog server. Start by adding the syslog server and IP information in the firewall hosts file. Then, enter the hostname of the syslog server to /etc/syslog.conf on the firewall:

kern.* @syslog.giacrock.com

The above configuration will send all kernel level facilities to the remote log server /var/log/messages file.

On the syslog server edit syslog.conf to send firewall messages to

/var/log/firewall. The following entry will send iptables messages from both the syslog server and firewall to the firewall log file:

```
kern.warning      /var/log/firewall
```

To test if syslog if configured correctly, use the tail -f command on the /var/log/firewall file. From a workstation attempt to connect to the firewall on TCP port 80, which will be denied. The following entry will be seen in the log:

```
Jul 10 21:32:37 firewall kernel: FW_IN IN=eth1 OUT= MAC=
00:0c:29:80:66:08:00:0c:29:36:37:a2:08:00 SRC=172.16.0.10 DST=172.16.0.1
LEN=48 TOS=0x00 PREC=0x00 TTL=128 ID=181 DF PROTO=TCP SPT=1050 DPT=80
WINDOW=64240 RES=0x00 SYN URGP=0
```

If this log entry is seen the firewall is correctly sending log messages to the syslog server.

Proxy Server

GIAC Rock's proxy server will be used as a caching and content filtering server, which will block websites based on content. GIAC Rock's deployment will consist of a squid proxy server and squid-guard for content filtering

After installation, a few configuration steps must be completed to ensure that squid is working correctly. All configuration steps performed are in the squid.conf file. The four changes below are required for squid to work in GIAC Rock's environment.

Configure squid to listen on port 8080 for http client request by entering this line in the squid.conf file:

```
http_port 8080
```

Next, an Access Control List (ACL) entry for the local area network is configured:

```
acl lan src 172.16.0.0/24
```

Configure squid to act as a cache server by entering in the following entry into the squid.conf file:

```
cache_dir ufs /usr/local/squid/var/cache/ 100 16 256
```

Finally, configure squid to accept connections from systems on the network:

```
http_access allow lan
```

Now that squid is configured, it is time to build the swap directories using the command squid -z. Once the swap directories are made, start squid in debug mode to ensure the proxy is configured correctly:

```
squid -d 1 -N
```

When configured correctly and squid is ready to accept new connections this message should be seen:

```
| Ready to serve request
```

Configure the Windows IT admin client to use proxy.giacrock.com:8080 for the proxy. Connect to www.sans.edu and www.giac.org to test that the proxy is configured correctly. Once squid is working correctly, it is time to configure

squid to perform content filtering.

With squid functioning correctly, configure websites to be blocked based on content is next. The squid add-on squidGuard (http://www.squidguard.org) will be used for content filtering. After installing squidGuard, a blacklist must be downloaded. For the purpose of this paper the blacklist form urlblacklist.com will be used.

GIAC Rock's management stated for content filtering all websites are denied unless explicatively allowed. Websites allowed must fall into the following categories:

Government

Banking

Financial

Search Engines

Weather

Whitelist

Copy the orginal squidGuard.conf to squidGuard.conf.orginal. A text editor will be used to create a new squidGuard configuration named squidGuard.conf.

Begin by declaring the locations of the blacklists and the location of the log file. Add the two lines below to declare the locations of the blacklist and log

files:

```
Dbhome /usr/local/squidGuard/db/blacklists

Logdir /usr/local/squidGuard/logs
```

Next, the type of sites that GIAC Rock's policy allows will have to be defined. Before defining the domainlist and urlslist ensure that both file exists, otherwise the file conversion will fail. The dest keywork defines the category to be allowed the keyword gov is the set of domainlist and urllist to be used as seen below:

```
dest gov {
        domainlist        government/domains
        urllsit           government/urls
}
```

The same format needs to be defined for all categories that will be allowed out through the proxy.

After all of the categories are defined, the ACL is created that will allow traffic based on categorization as:

```
acl {
        default{
```

```
        pass whitelist gov banking financial searchengines wearther !in-
        addr none

        redirect http://localhost/blocked.html

    }

}
```

The redirect is set to a custom html file that informs the user to contact IT to get the violating website whitelisted. For this feature to work, ensure that this file is created and a web server is running.

Next, the downloaded blacklist needs to be converted to the db format (squidGuard, 2008). First extract the blacklist file using the tar -zxvf command. Next usethe following command to convert to blacklist text files to the db format:

```
squidGuard -C all
```

This process may take a while. Upon successful completion of the file conversion, the following output should be seen:

```
2008-07-22 04:59:12 [5228] squidGuard 1.3 started

2008-07-22 04:59:12 [5228] db update done

2008-07-22 04:59:12 [5228] squidGuard stopped
```

Next, the user squid must be giving ownership of the blacklist files using the following command:

chown -R squid /usr/local/squidGuard/db/blacklist/*

When the configuration of squidGuard is complete, the configuration should be tested. The following command will test the squidGuard configuration:

echo http://www.weather.com 172.16.0.10/ - - GET | squidGuard -c /usr/local/squidGuard/squidGuard.conf

The first entry is the url to be tested, the second entry is the IP address of the client to test and the third entry is for user authentication. The above command should return the following output that squidGuard is ready to receive request:

2008-07-22 05:05:03 [5245] squidGuard 1.3 started (1216721103.560)

2008-07-22 05:05:03 [5245] squidGuard ready for requests (1216721103.569)

2008-07-22 05:05:03 [5245] squidGuard stopped (1216721103.595)

Finally, squid must be configured to send all request to squidGuard for content filtering. To achieve this, add the following line to the squid.conf file:

redirect_program /usr/local/bin/squidGuard -c /usr/local/squidGuard/squidGuard.conf

With squid and squidGuard configured, the windows IT admin workstation is used to test the configuration. Using the windows IT admin workstation, connect to www.weather.com. This website should be correctly displayed. Finally to ensure that content filtering is configured correctly attempt to connect to www.espn.com, a web

page with the custom block statement should be displayed.

Internet Router

The first network device to implement egress filtering will be at GIAC Rock's Internet Router, which is a Cisco IOS router. The device hostname is INETROUTER and will be configured with a simple egress filter that only permits GIAC Rock's public IP address space out. Below are the command that will create an ACL named EGRESSFILTER that will meet the requirement of permitting out only GIAC Rock's public IP address space.

```
INETROUTER#conf t

INETROUTER(config)#ip access-list ext EGRESSFILTER

INETROUTER(config-ext-nacl)#10 permit ip 1.1.1.0 0.0.0.31 any log

INETROUTER(config-ext-nacl)#20 deny ip any any log

INETROUTER(config-ext-nacl)#exit
```

The first rule will log and permit any packet with the source IP address of 1.1.1.0/27 destined to any destination. ACL logging is used to provide information in the event of any type of investigation. The second rule logs and denies packets from all other source IP addresses. Log entries can help locate infected or compromised host, a mis-configured host, or spoofed packets that have bypassed the firewall. With the ACL created, the commands to apply the ACL to the correct interface, in this case FastEthernet0/0, are seen in this example:

```
INETROUTER(config)# int f0/0
```

```
INETROUTER(config-if)#ip access-group EGRESSFILTER in
```

Now that the egress filter is written and applied to the "inside" interface of the router, it is time to test the filter. Because ICMP is allowed out, a test ping to www.google.com (192.168.1.1) is performed. The ping will succeed because the source address is 1.1.1.2/32 as seen in this log entry:

```
Jul 22 19:10:09.268: %SEC-6-IPACCESSLOGDP: list EGRESSFILTER permitted icmp
1.1.1.2 -> 192.168.1.1 (0/0), 1 packet
```

This simple test proves anything with the source address of 1.1.1.0/27 will work. Now it is time to test the second entry in the ACL. Using nmap (www.insecure.org) packets with spoofed source address of 80.25.1.4 (a randomly picked IP address) will be sent. Since the egress filter only permits traffic from GIAC Rock's public IP space this traffic is denied by the EGRESSFILTER as seen in this log entry:

```
Jul 22 19:18:07.010: %SEC-6-IPACCESSLOGDP: list EGRESSFILTER denied tcp
80.25.1.4 (0) -> 192.168.1.1 (0), 1 packet
```

The above log entry verifies that IP addresses outside of GIAC Rock's public space will be logged and denied.

Firewall

The final implementation stage will be GIAC Rock's firewall, a dedicated Linux server running iptables v1.3.8. The firewall hostname is FIREWALL and contains two interfaces; inside and outside. The firewall will be responsible for Network Address Translation (NAT) from the internal network to the Internet.

When acting as a firewall, iptables process traffic flow through three tables, for the purpose of this paper the *filter* table will be focused on. For a detailed description of how traffic flows through iptables please look at packet-filtering-howto chapter 6 on the www.netfilter.org website (Netfilter, 2008). When traffic is flowing through the firewall the FORWARD chain is used.

When implementing iptables, it is critical to know the default policy accepts all packets. Since GIAC Rock's default policy is to deny traffic, the FORWARD chain default action must be changed. To silently drop packets on the FORWARD chain, issue the following command:

```
iptables -P FORWARD DROP
```

To simplify management of the FORWARD chain the following three chains will be created:

```
TCP_FORWARD

UDP_FORWARD

ICMP_FORWARD
```

The above chains were created using the following three commands:

```
iptables -N TCP_FORWARD

iptables -N UDP_FORWARD

iptables -N ICMP_FORWARD
```

Update the FORWARD chain to jump to the chains created above based on the protocol that is attempting to traverse the firewall. Use the following command to jump from the FORWARD chain to the chains defined above based on protocols:

```
iptables -I FORWARD 1 -p tcp -j TCP_FORWARD

iptables -I FORWARD 2 -p udp -j UDP_FORWARD

iptables -I FORWARD 3 -p icmp -j ICMP_FORWARD
```

Since all packets meet GIAC Rock's default deny security policy, logging will be turned on for all dropped packets. Dropped packets can be used to investigate issues that may occur with egress filtering, as well as provide clues in the event of an incident. Using iptables log-prefix extensions allows customizable labels and makes searching logs a bit easier. The following commands will label dropped packets with the label "FW_FORWARD ":

```
iptables -I TCP_FORWARD 1 -j LOG --log-prefix "FW_FORWARD "

iptables -I UDP_FORWARD 1 -j LOG --log-prefix "FW_FORWARD "

iptables -I ICMP_FORWARD 1 -j LOG --log-prefix "FW_FORWARD "
```

Using the defined business requirements the firewall will be configured to allow the required services to leave GIAC Rock's network.

The order that egress filtering is implemented will have to be determined by the administrator. GIAC Rock will implement by service starting with network diagnostic protocols, followed by DNS, HTTP, HTTPS, Linux updates, Windows updates,

NTP, SMTP, and finally FTP.

GIAC Rock requires that the network diagnostic tools are only permitted from the two IT workstations, one running Windows XP and the running Fedora 9. The IP address of the Windows IT workstation is 172.16.0.10, and the IP address of the Linux IT workstation is 172.16.0.11. The following network diagnostics protocols are required:

Echo-Request (Ping)

Echo-Reply (Pong)

Traceroute

First ping and ping replies will be configured. Since GIAC Rock's policy is to allow ping and ping replies from the two IT workstations the following four rules must be created:

```
iptables -I ICMP_FORWARD 1 -i eth1 -o eth0 -s 172.16.0.10/32 -p icmp --icmp-type ping -m state --state NEW -j ACCEPT

iptables -I ICMP_FORWARD 2 -i eth0 -o eth1 -d 172.16.0.10/32 -p icmp --icmp-type pong -m state --state ESTABLISHED -j ACCEPT

iptables -I ICMP_FORWARD 3 -i eth1 -o eth0 -s 172.16.0.11/32 -p icmp --icmp-type ping -m state --state NEW -j ACCEPT

iptables -I ICMP_FORWARD 4 -i eth0 -o eth1 -s 172.16.0.10/32 -p icmp --icmp-type pong -m state --state ESTABLISHED -j ACCEPT
```

Now that ping is filtered, traceroute will be configured on the firewall. The Windows operating system traceroute uses the ICMP protocol to conduct a traceroute. The Linux operating system by default uses UDP packets with a destination port range of 33434 to 33600 (Joe, 2008) for traceroute stimulus. The response packets are ICMP packets (Inetdaemon, 2008).

Following GIAC Rock's security policy, traceroute from the two IT workstations will be allowed to traverse the firewall. The first two rules are for Windows IT workstation to use traceroute. The next two rules allows the Linux IT workstation to complete the traceroute:

```
iptables -I ICMP_FORWARD 5 -i eth0 -o eth1 -d 172.16.0.10/32 -p icmp --icmp-type 11 -m state --state RELATED -j ACCEPT

iptables -I ICMP_FORWARD 6 -i eth1 -o eth0 -s 172.16.0.10/32 -p icmp --icmp-type 8 -m state --state ESTABLISHED -j ACCEPT

iptables -I UPD_FORWARD 1 -i eth1 -o eth0 -s 172.16.0.11/32 -p udp --dport 33434:33600 -m state --state NEW -j ACCEPT

iptables -I ICMP_FORWARD 7 -i eth0 -o eth1 -d 172.16.0.11/32 -p icmp --icmp-type 3/3 -m state --state RELATED -j ACCEPT
```

The IT workstations can now successfully perform network troubleshooting using built in tools.

Because Domain Name System (DNS) is critical to many systems, the firewall will be configured to allow DNS traffic from the DNS server (172.16.0.3). All

systems requiring DNS in the GIAC Rock network will point to the DNS server. To allow DNS query and response from the DNS server, the following rules must be created:

```
iptables -I UDP_FORWARD 2 -i eth1 -o eth0 -s 172.16.0.3/32 -p udp --dport 53
-m state --state NEW -j ACCEPT

iptables -I UDP_FORWARD 3 -I eth0 -o eth1 -d 172.16.0.3/32 -p udp --sport 53
-m state --state ESTABLISHED -j ACCEPT
```

With DNS working, Internet browsing will be configured. Only the squid proxy server will be allowed out the firewall, with all systems requiring Internet access using the proxy server to access the Internet. The following rules were created to allow the proxy server access to the Internet:

```
iptables -I TCP_FORWARD 1 -i eth1 -o eth0 -s 172.16.0.3/32 -p tcp --dport 80
-m state --state NEW -j ACCEPT

iptables -I TCP_FORWARD 2 -i eth0 -o eth1 -d 172.16.0.3/32 -p tcp --sport 80
-m stae --state ESTABLISHED -j ACCEPT

iptables -I TCP_FORWARD 3 -i eth1 -o eth0 -s 172.16.0.3/32 -p tcp --dport 80
-m state --state ESTABLISHED -j ACCEPT

iptables -I TCP_FORWARD 4 -i eth1 -o eth0 -s 172.16.0.3/32 -p tcp --dport 443
-m state --state NEW -j ACCEPT

iptables -I TCP_FORWARD 5 -i eth0 -o eth1 -d 172.16.0.3/32 -p tcp --sport 443
-m state --state ESTABLISHED -j ACCEPT
```

```
iptables -I TCP_FORWARD 6 -i eth1 -o eth0 -s 172.16.0.3 -p tcp --dport 443 -m
state --state ESTABLISHED -j ACCEPT
```

GIAC Rock Linux systems are configured to use yum for operating system updates, with a central repository server (172.16.0.4). All Linux systems point to the yum server that obtains its updates from the Internet. The yum server is configured to use the proxy server to retrieve Linux updates.

For Windows updates, GIAC Rock had deployed a Windows Server Update Service (WSUS) for Windows workstations to receive approved updates. WSUS is configured to use the proxy server to retrieve Windows updates.

Network Time Protocol (NTP) is required to ensure that all systems are synchronizing their time. GIAC Rock deploys one central NTP server (172.16.0.6) for network devices to synchronize their time. To allow NTP from the time server the following rules must be applied to the firewall:

```
iptables -I UDP_FORWARD 4 -i eth1 -o eth0 -s 172.16.0.6/32 -p udp --sport 123
--dport 123 -m state -state NEW -j ACCEPT
```

```
iptables -I UDP_FORWARD 5 -I eth0 -o eth1 -d 172.16.0.6/32 -p udp --sport 123
--dport 123 -m state --state ESTABLISHED -j ACCEPT
```

GIAC Rock's policy is to permit email out from only from the email gateway (172.16.0.7). To allow email out the following three rules must be created:

```
iptables -I TCP_FORWARD 7 -i eth1 -o eth0 -s 172.16.0.7/32 -p tcp --dport 25
-m state --state NEW -j ACCEPT
```

```
iptables -I TCP_FORWARD 8 -i eth0 -o eth1 -d 172.16.0.7/32 -p tcp --dport 25
-m state --state ESTABLISHED -j ACCEPT

iptables -I TCP_FORWARD 9 -i eth1 -o eth0 -s 172.16.0.7/32 -p tcp --dport 25
-m state --state ESTABLISHED -j ACCEPT
```

GIAC Rock has two custom applications, Easy Rock Making and Rock Sorting. These applications use non-standard ports for updates. To allow the applications out the following rules must be created:

```
iptables -I TCP_FORWARD 10 -i eth1 -o eth0 -s 172.16.0.0/24 -p tcp --dport
12345 -m state --state NEW -j ACCEPT

iptables -I TCP_FORWARD 11 -i eth0 -o eth1 -d 172.16.0.0/24 -p tcp --sport
12345 -m state --state ESTABLISHED -j ACCEPT

iptables -I TCP_FORWARD 12 -i eth1 -o eth0 -s 172.16.0.0/24 -p tcp --dport
12345 -m state --state ESTABLISHED -j ACCEPT

iptables -I TCP_FORWARD 13 -i eth1 -o eth0 -s 172.16.0.0/24 -p tcp --dport
6113 -m state --state NEW -j ACCEPT

iptables -I TCP_FORWARD 14 -i eth0 -o eth1 -d 172.16.0.0/24 -p tcp --sport
6113 -m state --state ESTABLISHED -j ACCEPT

iptables -I TCP_FORWARD 15 -i eth1 -o eth0 -s 172.16.0.0/24 -p tcp --dport
6113 -m state --state ESTABLISHED -j ACCEPT
```

GIAC Rock uses File Transfer Protocol (FTP) for file transfer. Because of how

the FTP protocol works, there is a module that should be loaded to assist with FTP configuration. To accomplish this, edit the file /etc/sysconfig/iptables-config and add ip_nat_ftp and ip_conntrack_ftp to IPTABLES_MODULES section as seen in this output:

```
IPTABLES_MODULES=" ip_nat_ftp ip_conntrack_ftp"
```

The following iptables entries will support both passive and active FTP:

```
iptables -I TCP_FORWARD 16 -i eth1 -o eth0 -s 172.16.0.0/24 -p tcp --dport 21
-m state --state NEW -j ACCEPT

iptables -I TCP_FORWARD 17 -i eth0 -o eth1 -d 172.16.0.0/24 -p tcp --sport 21
-m state --state ESTABLISHED -j ACCEPT

iptables -I TCP_FORWARD 18 -i eth1 -o eth0 -s 172.16.0.0/24 -p tcp --dport 21
-m state --state ESTABLISHED -j ACCEPT

iptables -I TCP_FORWARD 19 -i eth0 -o eth1 -d 172.16.0.0/24 -p tcp --dport
1024:65535 -m state --state ESTABLSIHED,RELATED -j ACCEPT

iptables -I TCP_FORWARD 20 -i eth1 -o eth0 -s 172.16.0.0/24 -p tcp --sport
1024:65535 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

With FTP allowed out, all business requirements set forth in the design phase have been meet and the implementation of egress filtering at GIAC Rock is complete. Although the implementation phase of egress filtering is complete, there will be a need to monitor logs to search for possible incidents or issues that may arise from egress filtering.

10. Monitoring and Troubleshooting

Once egress filtering is implemented, the logs must be monitored. Without log monitoring, the value of egress filtering is lost. Log monitoring can be used to detect issues or security incidents that may occur.

To assist with log monitoring two methods will be discussed, manual monitoring using built in Linux tools that are useful for troubleshooting or to look for past trends. For automated log review and notification Swatch (http://swatch.sourceforge.net) will be used.

When performing manual log searches, there are several built-in Linux commands that can be used. These tools include tail, grep, sort, and cut. These powerful command line tools will be used to search the logs for events. Although these tools are powerful and great when searching for specific events, tools such as swatch should be used to monitor and notify for events.

For our example, command line tools will be used to search the firewall logs for any system attempting to make SMTP, HTTP, and HTTPS connections that are being denied. The same commands will be used to search the logs for the host with the IP address of 172.16.0.75 to see what destination ports connection attempts are being made to.

Performing a list on the firewall log, the file is over 40 MB. Although the log file can be parsed reading the entire contents, this is a time consuming process. To save time the command grep is used to parse through the log file to extraxt the "FW_FORWARD" log entries into a file named forward.log by using this command:

grep -I “FW_FORWARD” /var/log/firewall > forward.log

To look for host attempting to make connections on TCP/25, the following commands are issued that send the grep output to smtp.log:

grep -I “DPT=25” forward.log > smtp.log

All systems attempting connections to TCP/25 are now in smtp.log. To identify the host attempting to send connections the cat command coupled with the cut command are used to identify the source IP:

cat smtp.log | cut -d “ “ -f 9

The output from this command will list all connection attempts. To see unique hosts, the sort and uniq commands are used with the output sent to the file smtpfinal:

Cat smtp.log | cut -d “ “ -f 9 | sort | uniq > smtpfinal

Performing a list on the file smtpfinal the file size is 748 bytes as seen in this output:

```
[root@syslog log]# ls -aul smtpfinal

-rw-r--r-1 root root 748 Jul 23 10:54 smtpfinal
```

To perform the search for HTTP and HTTPS connections the same commands will be used with the output going to httpfinal and sslfinal, this time, the following commands are issued:

```
grep -I “DPT=80” forward.log | cut -d “ “ -f 9 | sort | uniq >
httpfinal

grep -I “DPT=443” forward.log | cut -d “ “ -f 9 | sort | uniq >
sslfinal
```

To search for host not using the proxy server the cat command was run on the httpfinal file. The output from the cat command is seeing below:

```
[root@syslog log]# cat httpfinal

SRC=172.16.0.15

SRC=172.16.0.112

SRC=172.16.0.121

SRC=172.16.0.123

← OUTPUT REMOVED →

SRC=172.16.0.92

SRC=172.16.0.99
```

Each host would investigated to see why the host is still attempting connections to destination port TCP/80. The likely cause is these systems are not configured to use the proxy server, or attempting to bypass the proxy server, or possibly infected with Malware.

To perform a search that counts destinations and destination ports for host 172.16.0.75 while sending the output to host75 the following command was issued:

```
Grep -I "SRC=172.16.0.75" forward.log | cut -d " " -f 9,10,18 | sort | uniq > host75
```

The output from the cat command provides the requested information:

```
[root@syslog log]# cat host75

4 SRC=172.16.0.75 DST=10.1.1.138 DPT=25

4 SRC=172.16.0.75 DST=10.1.1.138 DPT=443

6 SRC=172.16.0.75 DST=10.1.1.138 DPT=80
```

From the output, it appears the host is attempting to make email, web and SSL connections to host 10.1.1.138. With this information, the host should be investigated to see why these attempts are being made.

Log monitoring by using Linux command line tools is great for troubleshooting or searching logs for specific events. However, these methods are not useful for monitoring logs in real time. To monitor the logs in real time GIAC Rock will use swatch.

Swatch is a tool used to actively monitor log files and perform some predefined action. Before using swatch it can be downloaded from http://swatch.sourceforge.net and installed from source or install using a package manager application such as yum.

After installing swatch, a configuration file must be created. The GIAC Rock swatch configuration file will be located at /etc/swatch.conf. Using a text editor, create the swatch.conf file.

When configuring swatch, the first step is to tell swatch what to watch for. To tell swatch what to look for the command watchfor is used. An example of this is shown below:

```
watchfor /FW_FORWARD/
```

With swatch configured to look for an event, an action when the event happens must be defined. Swatch is capable of several actions including echoing the log entry to a terminal, making a sound, executing a command or sending an email. GIAC Rock wants to notify the security administrator via email when an event occurs. To accomplish this notification, the following line is added to the swatch.conf file:

```
watchfor /FW_FORWARD/
        mail=secadmin@giacrock.com,subject=Firewall_Deny
```

The above swatch configuration will send an email every time there is a firewall deny. This configuration could easily overwhelm a email system and should NOT be deployed.

GIAC Rock's wants it security administrator notified when the firewall denies HTTP, HTTPS and SMTP connections. When denies occurs on the firewall, an email should be sent with a custom subject with the type of deny that occurred.

An example of this configuration is seen below:

```
watchfor /FW_FORWARD/&/DPT=80/
        mail=secadmin@giacrock.com,subject=Firewall_HTTP_Deny
watchfor /FW_FORWARD/&/DPT=443/
        mail=secadmin@giacrock.com,subject=Firewall_SSL Deny
watchfor /FW_FORWARD/&/DPT=25/
        mail=secadmin@giacrock.com,subject=Firewall_SMTP_Deny
```

This configuration may still overwhelm an email system so threshold configuration should be used. The type of threshold is first defined, followed by the count, and finally the elapse time. GIAC Rock's threshold configuration will send an email if any event matches, then not send another email for 10 minutes as seen in this configuration:

```
watchfor /FW_FORWARD/&/DPT=80/
        mail=secadmin@giacrock.com,subject=Firewall_HTTP_Deny
            threshold type=limit,count=1,seconds=600
            continue
watchfor /FW_FORWARD/&/DPT=443/
        mail=secadmin@giacrock.com,subject=Firewall_SSL Deny
```

```
        threshold type=limit,count=1,seconds=600
        continue
watchfor /FW_FORWARD/&/DPT=25/
    mail=secadmin@giacrock.com,subject=Firewall_SMTP_Deny
        threshold type=limit,count=1,seconds=600
        continue
```

Now that the swatch configuration is complete, swatch must be started. When starting swatch, it must be told which configuration file to use, which log file to watch, and to run as a daemon. To accomplish, this the following command is entered:

```
swatch -c /etc/swatch.conf -t /var/log/firewall -daemon
```

With swatch running, when any of the above events fire, the security administrator will be notified. When notified, the commands used earlier in this section can be used to parse the log files to find systems that are mis-configured or have possibly been compromised.

Remember that swatch and other log analysis tools are meant to assist a security administrator in parsing logs, not replace the person reviewing the logs.

11. Conclusion

The reader should now have solid understanding of egress filtering. Throughout the paper, examples were given to provide a better understanding of egress filtering.

Reasons to perform egress filtering, including business and technical reasons were covered. Default egress policy types were explained, while stating pros and cons of each type. Potential issues, including egress filtering avoidance were discussed. Methods to gather requirements and information needed to implement egress filtering were discussed as well.

A fictitious network was used to design, and implement egress filtering. After implementation, troubleshooting and monitoring log files is covered. In this section tools used to search the logs for events that happen were covered, as well as a tool used to proactively monitor logs

12. Credits

I would like to thank my advisor, Joel Esler for all of his direction. His advice and knowledge were invaluable in the completion of this paper.

13. Reference

Andre Vladimirov, Konstantin Gavrilenko, & Andrei Mikhailovsky (2005). Hacking Exposed - Cisco Networks. Emeryville CA: McGraw-Hill Osborne

Application Layer Firewall, Application Layer Firewall, retrieved 2008, May 2 from http://en.wikipedia.org/wiki/Application_layer_firewall

Black Hole Routing, Black hole (networking), retrieved 2008, March 1 from http://en.wikipedia.org/wiki/Black_hole_%28networking%29

Cisco, Remotely Triggered Black Hole Filtering - Destination Based and Source Based, retrieved 2008, March 1 from http://www.cisco.com/warp/public/732/Tech/secuirty/docs/blackhole.pdf

Egress Filtering, Egress Filtering, retrieved 2008, March 1 from http://en.wikipedia.org/wiki/Egress_filtering

Inetdaemon, How Traceroute Works, retrieved 2008, July 18 from http://www.inetdaemon.com/tools/traceroute/definition.shtml

Joe, Joe's Bit Bucket: Linux traceroute vs Windows tracert, retrieved 2008, July 11 from http://joesbitbucket.blogspot.com/2006/10/linux-traceroute-vs-windows-tracert.html

Merike Kaeo (2004). Designing Network Security, Second Edition. Indianapolis, IN: Cisco Press

Michael D. Bauer (2005). Linux Server Security. Sebastopol, CA: O'Reilly

Netfilter, Linux 2.4 Packet Filtering HOWTO: How Packets Traverse The Fitlers, retrieved 2008, June 2 from http://www.netfilter.org/documentation/HOWTO/packet-filtering-HOWTO-6.html

Protocols, List of network protocols, retrieved 2008, March 1 from http://en.wikipedia.org/wiki/List_of_network_protocols

SANS, Egress Filtering FAQ, retrieved 2008, March 1 from https://www.sans.org/reading_room/whitepapers/firewalls/1059.php

Spectorcne, Reduce Inappropriate Use of the Internet and Increase Employee Productivity, retrieved 2008, March 1 from http://www.spectorcne.com/Solutions/Productivity.html

squidGuard, SquidGurad, retrieved 2008, July 22 from

http://squidguard.org/Doc/configure.html

Stephen Northcutt, Lenny Zeltser, Scott Winters, Karen Kent Fredrick, & Ronald W. Ritchey (2003). Inside Network Perimeter Security. Indianapolis, IN: New Riders Publishing

William R. Cheswick, Steve M. Bellovin, & Aviel D. Rubin (2003). Firewalls and Internet Security: Repelling the Wily Hacker. Boston, MA: Addison-Wesley

Last Updated: May 28th, 2017

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Madrid 2017	Madrid, ES	May 29, 2017 - Jun 03, 2017	Live Event
SANS Atlanta 2017	Atlanta, GAUS	May 30, 2017 - Jun 04, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CAUS	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DCUS	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TXUS	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Milan 2017	Milan, IT	Jun 12, 2017 - Jun 17, 2017	Live Event
SEC555: SIEM-Tactical Analytics	San Diego, CAUS	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Charlotte 2017	Charlotte, NCUS	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Secure Europe 2017	Amsterdam, NL	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, COUS	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MNUS	Jun 19, 2017 - Jun 24, 2017	Live Event
DFIR Summit & Training 2017	Austin, TXUS	Jun 22, 2017 - Jun 29, 2017	Live Event
SANS Paris 2017	Paris, FR	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, AU	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MDUS	Jun 26, 2017 - Jul 01, 2017	Live Event
SEC564:Red Team Ops	San Diego, CAUS	Jun 29, 2017 - Jun 30, 2017	Live Event
SANS London July 2017	London, GB	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, JP	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, SG	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Los Angeles - Long Beach 2017	Long Beach, CAUS	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS ICS & Energy-Houston 2017	Houston, TXUS	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, DE	Jul 10, 2017 - Jul 15, 2017	Live Event
SANSFIRE 2017	Washington, DCUS	Jul 22, 2017 - Jul 29, 2017	Live Event
Security Awareness Summit & Training 2017	Nashville, TNUS	Jul 31, 2017 - Aug 09, 2017	Live Event
SANS San Antonio 2017	San Antonio, TXUS	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, CZ	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MAUS	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Hyderabad 2017	Hyderabad, IN	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UTUS	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS New York City 2017	New York City, NYUS	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Chicago 2017	Chicago, ILUS	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Adelaide 2017	Adelaide, AU	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Stockholm 2017	OnlineSE	May 29, 2017 - Jun 03, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced