Proxy Servers
Building and Deploying
System Improvement and Protection

Elvin Smith & Jose Robles
IT 4444 Capstone Spring 2013
Outline

Topic: *Proxy Server*

1. Introduction
   a. Technical definition of a proxy server
   b. Reason for research paper
      i. Build and configure a proxy server at low cost
      ii. Improve network performance
      iii. Improve network security
      iv. Create web filtering for the work place
      v. Traffic monitoring

2. Body
   a. Background
      i. Introducing squid
      ii. History of squid
      iii. Software and hardware requirements
      iv. Installation
   b. Configuration
      i. Configure as a web filter
      ii. Configure authentication and access
      iii. Configure as a caching server
      iv. Configure to monitor traffic
      v. Configure transparent mode
   c. Results
      i. Test access to websites
      ii. Monitor Traffic

3. Conclusion
   a. Summary
Abstract

Proxy servers act as intermediaries between computing devices, to provide security and to shield the hosts on an internal network. It is important that we understand the purposes and processes of this technology, in order to help protect our information, systems and networks. This research will provide us with the ability to build, configure and deploy a Squid proxy server. This paper will describe Squid proxy servers, discuss configuration and discuss results of authentication, performance and Web access testing.

Introduction

The uses of internet connectivity by Universities, Businesses and Organization networks, have grown exponentially in the last twenty years. With that connectivity explosion, comes the potential vulnerability exploitation by attackers, malicious code and other threats being directed toward those networks. The ability to secure the network and control access from workstations has become a difficult task for network administrators. The use of Access Controls, Intrusion Detection Systems, Firewalls and Scanners, all help network administrators secure those networks but may not provide the level they want. With the addition of a secure Proxy Server, they can add an additional level of security, improve network performance and monitor web activity that is being accessed over their network.

As we attempt to increase our security of the network by adding a proxy server, we must understand that Proxy servers are basically separated into two types, Transparent and Anonymous Proxies. Transparent Proxies forward user requests to a destination without hiding or concealing any information. This type of proxy server is usually used on internal networks, where the need to obscure the IP address in not warranted because the computers on that network are safe from external threats. Anonymous Proxies, on the other hand, allow users to surf the web while keeping their IP address anonymous or hidden. Most universities, large businesses and organizations today use these proxies to act as a middle point between a user and the destination address. This middle point or “Proxy” makes a request on behalf of a user on the network and obscures their IP address from that destination site. These
proxies also improve the performance of the network by using a caching system to save network users recent request responses from the web to its local hard drive. By caching this web data, it eliminates the need for other users on the network to download that same information from the Web again, providing faster access to those sites, images and files and saving network bandwidth. This anonymous type of Proxy server will be used in the purposes of this paper, as we concentrate our efforts toward the building of a Squid Server.

Efforts in building a secure server at a low cost, configuring that server to increase network performance while aiding in network security from outside threats and monitoring traffic on the network are key for a network administrator’s success. We will use this research paper to enhance our knowledge and ability to accomplish this in our future endeavors as IT professionals.

**Previous Work**

Over the last 15 years internet usage in the United States alone has increased. In 1995 one out of ten adults accessed the internet. More than a decade later 78% of adults and 98% of teenagers accessed the internet. (2) Increase internet usage and complexity resulted in an increase in security incidents. In 1995 2540 incidents were reported. Just five years later 3234 incidents were reported. From 2000 to 2003 incidents reported climbed to 82,064 reports. (3) One of the first proxy servers used was a windows service named Wingate. Wingate was used to share internet dial up connection among multiple devices. The program came with a security hole and people quickly learned how to connect to Wingate externally. External users could then piggy back of Wingate and telnet to other devices obscuring their source information. (4) Modern Day network administrators have to be extra careful when configuring proxy servers. New trends and increase in proxy server abuse are greater now than ever before. (4) Understanding the different methods will help correctly configure our squid proxy server. Two methods that will be explored are the transparent method and the traditional method. The traditional method will capture traffic by configuring the web proxy settings of network devices to point traffic to the proxy server. (5) The transparent method will place the proxy server in line with the gateway. The only
physical path to the gateway will be through the proxy server. In transparent mode, users will not see that the proxy server is being used. (5) Companies with heavy usage networks turn to caching to cater the needs of the user. (7) There are many factors on effectiveness of using a caching server. Incorrect configuring of a proxy server can lead to more bandwidth usage as opposed to bandwidth conservation. (6) Correctly configuring the squid proxy server for security and performance will be a major focus in this paper.

**Background**

*Introducing Squid*

Squid is a software that provides the ability to implement access controls, traffic optimization, authorization, and logging. Squid provides full features for the http/1.0 proxy. [8]

*History*

The history of squid can be traced back to the early 1990’s. It is a branch off of a caching project with the name of Harvest project. During the completion of Harvest the project broke off into two different paths. One path become a project known as Netapp’s Netcache. The other path resulted in the project creating Squid. Initially the funding was provided by a grant. The grant was part of a project called IRCache which in turn created processes for the squid caching software. Eventually funding for IRCache would run out. Squid would be eventually developed by donations and volunteers. To this day Squid is an open source software ran by donations and volunteers with some investments. [8]

*Software and Hardware Requirements*

The platform chosen for the installation of Squid is CentOS Linux operating system. There are many advantages for a company to use CentOS. To understand the advantages of CentOS we will introduce Red Hat Linux. Red Hat Linux has built a reputation for being dependable, supported, and resourceful operating system in a serve environment. [15] The advantage of using CentOS over Red Hat Linux is the lower cost. Red Hat Linux must be purchased with warranties and technical support. CentOS can be obtained freely with no cost, making it the cheaper alternative to Red Hat Linux. CentOS is one hundred percent
compatible with Red Hat Linux. Both Red Hat Linux and CentOS are open source software. The consumer is free to manipulate code to suit their needs. For the purpose of this paper CentOS 6 will be used over Red Hat Linux to keep costs to a minimal. The disadvantage of using CentOS 6 over Red Hat Linux is the loss of technical support and warranties. Although CentOS lacks tech support, solutions to problems with the operating system can be researched through the CentOS community. CentOS community provides forums for users to share their solutions and problems with the internet community. As long as the server is not responsible for critical tasks that require minimal downtime CentOS can be used as a cheap alternative to Red Hat. The CentOS operating system will be installed on a clean desk environment. To install the operating system users must download CentOS from the CentOS website http://www.centos.org. Instructions on installing and updating can also be found on the support section of the website. Recommended requirements for hardware include 10 MB of RAM per GB of the total of all cache_dirs plus an additional 20 MB. The hardware used in this research is a rack mounted server with an Asus PMR15 mother board. The server has three 1Gig Ethernet ports. Two sticks of ddr2 ram at two Gigabytes per stick. A 64 bit processor with a 500 Gigabyte hard drive.

Installation

The version of Squid that will be installed is version 3.3.1. When downloading squid the user will find that there are multiple methods to choose from. The easiest method for CentOS is to use the command “yum install squid”. CentOS will automatically download and install the necessary packages for running Squid. Another method can be achieved by browsing the Squid website and downloading preconfigured binaries. For this paper we will use the method of downloading the source code and compiling it. Downloading the source code and compiling it gives the user more control of configuration and location of the directories. The source packages can be found at the Squid website. In this case the latest stable version was downloaded (Version 3.3). The source code will be a compressed file available in a tar.gz format. After downloading the source code the compressed file has to be extracted into
a directory. In this case the compressed file was extracted to the directory named squid-3.3.1. After the directory is created the "./configure" command is executed to configure the source code and choose the directory. Within the configure command, features can be enabled or disabled depending on what features the user needs. Features enabled during configuration and what value they are set to can be found in figure 1. Completion of the configure command will create files that are executable. These file are called make files in the linux world. Final step of installation can be made by executing the command “make install”. This will complete the installation of Squid. After installation, the “squid” command is executed to start the services provided by squid.

---

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix</td>
<td>/local/usr/squid</td>
</tr>
<tr>
<td>--with-logdir</td>
<td>/var/log/squid</td>
</tr>
<tr>
<td>--with-pidfile</td>
<td>/var/run/squid/pid</td>
</tr>
<tr>
<td>--enable-store</td>
<td>Ufs,aufs</td>
</tr>
<tr>
<td>--enable-removal-policies</td>
<td>Lru,heap</td>
</tr>
<tr>
<td>--enable-icmp</td>
<td></td>
</tr>
<tr>
<td>--enable-useragent-log</td>
<td></td>
</tr>
<tr>
<td>--enable-referer-log</td>
<td></td>
</tr>
</tbody>
</table>

---

Fig.1

**Configuration**

*Configure as a web filter*

To be able to configure services on a squid server a user must locate and edit the file named squid.conf. This configuration file will contain most of the configuration options available in squid. The location of the squid.conf for the server used in this paper can be found by changing to the directory in this path “/usr/local/squid/etc/squid.conf”.

Locations of files and directories will be different depending on initial configurations during installation. A snippet of the configuration file can be found in fig.3. Filtering can be accomplished by configuring access control lists. The configuration of access control lists are configured by editing the squid.conf file. The syntax of the access control list start with the name of the access control list or (acl). Followed by a directive that will control the action of the will happen once a match is made with the (acl). For an example, to block yahoo you will create an acl named block_yahoo.
This acl can be anything the user wishes to name it. The following line will be entered into the squid.conf file:
“acl bad_yahoo destdomain .yahoo.com”. This acl will match up any request that has yahoo.com in the domain. After naming the acl we have to define what we want to do with the request. The directive http_access deny will deny http access to the acl. We can deny a user connected to the proxy by adding the line http_access deny bad_yahoo. Squid will now match requests containing yahoo.com and deny access to this request. It is important to add http_deny all to the bottom of the directives in the file. Squid will work from top to bottom matching acl’s and deny access to anyone else. For this example we add the lines to allow access to squid from clients on the 192.168.0.0/24 network. The following lines were added to the squid.conf file: “acl safe_network src 192.168.0.0/24” “http_access allow safe_network.” This will allow any machine with the source ip address in the 192.168.0.0/24 network http access. Multiple websites can be denied In one acl by using the following syntax: “ acl all_bad_sites dstdomain .youtube.com .google.com .ytimg.com .yahoo.com .woopra.com”. Then add the line bad_sites http_deny access. Squid can also deny access to a specific mac address. This can be done by adding the line “acl bad_mac arp “(mac address goes here)” More options using acl can be found in chapter 2 of Squid Proxy Server 3.1.

### Recommended minimum configuration:

```text
# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing should be allowed

dacl localnet src 10.0.0.0/8     # RFC1918 possible internal network
dacl localnet src 172.16.0.0/12   # RFC1918 possible internal network
dacl localnet src 192.168.0.0/16

dacl localnet src fc00::/7       # RFC4193 local private network range

dacl localnet src fe80::/10      # RFC4291 link-local (directly plugged)
```

[12]
<table>
<thead>
<tr>
<th>Machines</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl notworkreg dstdomain</td>
</tr>
<tr>
<td><a href="http://www.youtube.com">www.youtube.com</a> .googlevideo.com</td>
</tr>
<tr>
<td>.theblaze.com .facebook.com</td>
</tr>
<tr>
<td>.nbcnews.com .abcnews.go.com</td>
</tr>
<tr>
<td>.twitter.com .foxnews.com</td>
</tr>
<tr>
<td>.newsok.com .kswo.com .abc.go.com</td>
</tr>
<tr>
<td>.aol.com .tds.net .woopra.com</td>
</tr>
<tr>
<td>acl my_machine src 192.168.0.254</td>
</tr>
<tr>
<td>acl SSL_ports port 443</td>
</tr>
<tr>
<td>acl Safe_ports port 80 # http</td>
</tr>
<tr>
<td>acl Safe_ports port 21 # ftp</td>
</tr>
<tr>
<td>acl Safe_ports port 443 # https</td>
</tr>
<tr>
<td>acl Safe_ports port 70 # gopher</td>
</tr>
<tr>
<td>acl Saf</td>
</tr>
</tbody>
</table>

Fig. 3

**Configure authentication and access**

**Basic authentication**

There are many options for configuring authentication and access for the squid server. For this paper basic configuration will be discussed. For the complete list of options refer to chapter 7 in Squid Proxy Server 3.1.

[12] Basic authentication on squid is the easiest to configure and used the most simple encryption. It is also the most insecure of the squid authentication methods. Basic configuration transmits in a Base64-code string format. This format is easily decoded by an observer using a sniffer on the network. For the private environment in this test lab, basic authentication can be used without concern of intrusion. Authentication can be implemented by using the auth_param directive in the squid configuration file. The auth_param directive comes with many options, one option is the “auth_param basic program” option. This option will specify what directory to send authentication requests to check credentials. The directory in this Squid configuration can be found in “/usr/local/squid/libexec/”. An acl also has to be created to configure authentication. Figure 4 will give an example of basic authentication code in the squid configuration file.

```
auth_param basic program
/usr/local/squid/libexec/basic_pam_auth

acl authenticated proxy_auth
```
Configure as a caching server

Creating cache directories

When configuring squid for a caching server the cache directives can be used. To create a directory for caching web documents, the rule in “cache_dir aufs /squid_cache/ 51200 32 512” is placed in the squid.conf file. This directive creates a directory named squid_cache with 50 GB of free space. Squid organizes files into hierarchical levels. The previous rule created the first level with 32 directories, and the second level with 512 directories. This is the minimal configuration for the server to start caching objects. There are many more options for the cache directive that can be found in chapter 2 of Squid 3.1. [12]

Monitoring traffic

Access and error log

Squid has two very useful log files. One log is the access log, this log file will log all connections to the squid proxy server.

The error log file logs errors in the squid server. To monitor traffic check both of these files for information. The access log is in a format that is hard to read by humans. A Pearl script can be used to decipher the logs. Adding the “tail –f” command to the beginning of the script will let a user see the access attempts in real time without having to refresh.

Cache manager

Squid also has a gui interface that can be accessed by a web server. In this example Apache is installed on the centos machine. A pearl script is added to the Apache configuration file that will point clients to the cgi file in the Squid directory. The cgi file in squid is named cachemgr.cgi and is an executable file. The location of the file in the directory is “/usr/local/squid/libexec/”. Accessing the cache manager allows you to monitor a wealth of useful information. Some examples of information on the cache manager are general runtime information, IP cache stats and contents, http header statistics, traffic resource counters, and request forwarding statistics. These are just a few statistics that can be monitored by the network. A complete list of options with the Squid cache manager can be found on chapter 6 of Squid Proxy Server 3.1. [12]

Transparent mode

```
REQUIRED

http_access allow authenticated
http_access deny all

Fig. 4

```
Intercept

To configure Squid in transparent mode we have to use a router to forward the traffic to the server. In the router forwarding rules, all traffic using port 80 needs to be directed to the IP address of the Squid server. The router also has to redirect traffic to the port Squid server is listening to. In this case traffic is redirected to 192.168.0.181 and port 3128. On the Squid server machine IP tables is used to handle incoming traffic to the Squid server. In the squid.conf file we need to add the directive “http_port 3128 intercept”. Configure these three steps and clients will not be able to see evidence of a proxy server being used.

Results

Test Lab

The test environment includes a lan on the 192.168.0./24 network. The Squid server shares the network with five PC’s with windows 7 installed. There will be five machines including my PC that will be connecting to the squid proxy server. Refer to figure 5 for the diagram of the network topology.

Testing web Filtering

For the first test my PC will be pointed to the Squid server by configuring my browser to use proxy setting with IP address 192.168.0.181. To test the access control list I pointed the address to www.google.com:897 to verify the squid access denied page. I also accessed the url www.yahoo.com, and confirmed the same result. The additional four pcs were pointed to the squid server and access denied pages were verified for the domains that were inserted in the squid.conf file.

Testing monitoring

To test the monitoring the following command was executed: “tail –f /var/log/squid/access.log | ccze –CA”. This will convert the time stamps to a readable format and also log the access attempts in real time. Monitoring the log allowed me to observe repeated connection from www.woopra.com. A google search showed this website to be a tracking software recording what website clients were visiting on the network. The site was blocked using Squid’s access control lists.
Future Works

Authentication

Squid Proxy Server has a lot of functionality. Some of the function provided by Squid Proxy Server is out of the scope of this paper. For example when implementing authentication there are several methods for encrypting user name and passwords sent by the client. There are different methods for storing the user name and passwords. Squid can use database software such as MySql for storing usernames and passwords. Radius servers can also be implemented to handle authentication requests.

Peer Caching

Squid has the ability to coordinate between other proxy servers. Multiple server can be set up in a hierarchical pattern. The parent server is placed in line with the network gateway. Child servers can be set behind this parent server. Child servers can also be parents of other child servers. Specific content can be handled by a child server to optimize performance on the network.

Reverse Proxy Mode

Squid also provides a function to cache static data for websites. The Squid configuration is called accelerator mode or reverse proxy mode. It will do the busy work of providing static data to clients reducing the client requests to the server. When dealing with large amounts of clients on the internet, this reduction can increase the performance of the web servers.

Conclusion

Summary

Squid Proxy Server provides protection on two fronts. It protects the client from accessing harmful data, and protects the network from incoming intrusion using access control lists. Performance is increased by using cache directories and fronting data to the clients. Performance and security can be monitored by access logs and Squids graphical interface cache manager.
Bibliography


## Plan of Work

<table>
<thead>
<tr>
<th>Dates</th>
<th>Week</th>
<th>Work Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 09 -11</td>
<td>1</td>
<td>Integration of teams.</td>
</tr>
<tr>
<td>Jan 12-24</td>
<td>2</td>
<td>Write abstract, bibliography, and outline.</td>
</tr>
<tr>
<td>Jan 25</td>
<td>3</td>
<td>Present abstract, bibliography, and outline.</td>
</tr>
<tr>
<td>Jan 26-feb 1</td>
<td>4</td>
<td>Write introduction, submit outline, bibliography, abstract, and plan of work.</td>
</tr>
<tr>
<td>Feb 02-08</td>
<td>5</td>
<td>Present introduction and Plan of Work.</td>
</tr>
<tr>
<td>Feb 09-15</td>
<td>6</td>
<td>Submit introduction.</td>
</tr>
<tr>
<td>Feb 16-22</td>
<td>7</td>
<td>Work on research installing linux CentOS and Squid.</td>
</tr>
<tr>
<td>Feb 23-march 1</td>
<td>8</td>
<td>Work on research to configure squid as a caching server.</td>
</tr>
<tr>
<td>March 2-08</td>
<td>9</td>
<td>Work on research to configure squid to filter web content.</td>
</tr>
<tr>
<td>March 25-29</td>
<td>10</td>
<td>Work on research to configure squid to monitor traffic.</td>
</tr>
<tr>
<td>March 30-april 5</td>
<td>11</td>
<td>Run tests on network performance, filtering, and monitoring.</td>
</tr>
<tr>
<td>Apr 6-12</td>
<td>12</td>
<td>Finish conclusion, finalize research paper and present research.</td>
</tr>
</tbody>
</table>