Protection of Web Sites By Using URL Rewriting Method In Reverse Web Proxy Service

Ei Phyu Hlaing, Su Su Lwin
University of Computer Studies, Magway.
eiphyuhlaing1@gmail.com

Abstract

Many people use the internet or the web as a source of information. In an enterprise that uses the internet, a reverse proxy server is a server that acts as an intermediary between a workstation user and internet so that the enterprise can ensure security, administrative control and caching service. Using a reverse proxy is to ease burden on a local web server that provides both static and dynamic content. Reverse proxy can determine who can and cannot connect to your local servers and to control the content that users are allowed to access. This paper proposed an effective web based solution for the internet user who wants to access an internal website through reverse proxy services. The system in this paper is also developed to provide various internal private web resources for trusted users of an organization when they are away from internal network.

1. Introduction

Some home networks, corporate intranets, and Internet Service Providers (ISPs) use proxy servers (also known as proxies). Proxy servers act as a "middleman" or broker between the two ends of a client/server network connection by intercepting all requests to the real server to see if it can fulfill the requests itself. If not, it forwards the request to the real server. Proxy servers work well between Web browsers and servers, or other applications, by supporting underlying network protocols like HTTP [1].

Proxy servers have two main purposes. One thing it can do is that it can dramatically improve performance for groups of users. This is because it saves the results of all requests for a certain amount of time. Consider the case where both user X and user Y access the World Wide Web through a proxy server. First user X requests a certain Web page, which will be called Page 1. Sometime later, user Y requests the same page. Instead of forwarding the request to the Web server where Page 1 resides, which can be a time-consuming operation, the proxy server simply returns the Page 1 that it already fetched for user X. Since the proxy server is often on the same network as the user, this is a much faster operation. Real proxy servers support hundreds or thousands of users. The major online services such as America Online, MSN and Yahoo, for example, employ an array of proxy servers.

Another feature of proxy servers is that it can filter requests. For example, a company might use a proxy server to prevent its employees from accessing a specific set of Web sites.

Proxies can do many other things. For example, they could translate multiple languages. They could shrink the size of a response so it fits on ones mobile phone web screen. They could also filter nasty language or subjects.

A reverse proxy server facing the Internet can be used to communicate to a fire walled server internal to an organization, providing extranet access to some functions while keeping the servers behind the firewalls [1].

In this paper, a reversed proxy service will be implemented to provide the trusted users of an organization to access the web servers inside the organisation’s Local Area Network when they are working outside of LAN.

This system is also intended to learn and enhance the security on the reversed proxy server to protect the internal servers and to prevent the internal webs from direct outside attacks.

2. Related works

A team from University of California, created an Anomaly-driven Reverse Proxy based on routing web requests to provide the detection of web based attacks for venerable web application in side the their University's system [2].

A team from University of Manitoba Libraries also implemented a reversed proxy solution to provide eligible library patrons with location-

independent access(may be through WAN IP addresses) to such resources.

The solution they developed is based on proxy server technology. First, they set up a proxy server on their campus network, and assigned to it an IP address that could access our IP-restricted resources. Then, they instructed users with external, unauthorized IP addresses to connect through this proxy server, rather than directly. The proxy server authenticates each user against their patron database, and then acts as a relay for their subsequent interaction with IP-restricted sites. This provides their users with the same access granted to users with authorized Local IP addresses.

Unlike the solution in this paper, they just used the open source proxy named Squid [3].

3. Proxy Server

A proxy server is a computer that offers a computer network service to allow clients to make indirect network connections to other network services. A client connects to the proxy server, and then requests a connection, file, or other resource available on a different server. The proxy provides the resource either by connecting to the specified server or by serving it from a cache. In some cases, the proxy may alter the client's request or the server's response for various purposes [9].

3.1 Caching Proxies

A proxy with a cache, acting as a server to clients and a client to servers is called a caching proxy. Caching proxies are often referred to as "proxy caches" or simply "caches". They are used to accelerate Web sites, to reduce traffic over expensive transoceanic links, and to reduce latency for groups of users in enterprises or at ISP sites [11].

3.2 Web proxies

A common proxy application is a caching Web proxy. This provides a nearby cache of Web pages and files available on remote Web servers, allowing local network clients to access them more quickly or reliably.

When it receives a request for a Web resource (specified by a URL), a caching proxy looks for the resulting URL in its local cache. If found the result URL, it returns the document immediately. Otherwise it fetches it from the remote server, returns it to the requester and saves a copy in the cache. Web proxies can also filter the content of Web pages served [9].

3.3 Types of web proxies

There are four different types of proxy servers:

- Transparent Proxy - This type of proxy server identifies itself as a proxy server and also makes the original IP address available through the http headers. They are transparent in the terms that user's IP address is exposed, not transparent in the terms that users do not know that they are using it.
- Anonymous Proxy - This type of proxy server identifies itself as a proxy server, but does not make the original IP address available. This type of proxy server is detectable, but provides reasonable anonymity for most users.
- Distorting Proxy - This type of proxy server identifies itself as a proxy server, but make an incorrect original IP address available through the http headers.
- High Anonymity Proxy - This type of proxy server does not identify itself as a proxy server and does not make available the original IP address [8].

3.4 Advantages of using proxies

By using proxies, the systems will get the following benefits.

- Performance: By saving a copy of the pages that it fetches, a proxy can reduce the need to create connections to remote servers.
- Content Filtering and Transformation: The proxy can inspect the requested URL and selectively block access to certain domains, reformat web pages (for instances, by stripping out images to make a page easier to display on a handheld or other limited-resource client), or perform other transformations and filtering.
- Privacy: Normally, web servers log all incoming requests for resources. This information typically includes at least the IP address of the client, the browser or other client program (user-Agent) that they are using, the date and time, and the requested file. If a client does not wish to have this personally identifiable information recorded, routing HTTP requests through a proxy is one solution. All requests coming from clients using the same proxy appear to come from the IP address and User-Agent of the proxy itself, rather than the individual clients [10].

3.5 Reverse Proxy Cache with Web Server

Reverse proxy cache, also known as Web Server Acceleration, is a method of reducing the load on a busy web server by using a web cache between the server and the internet. Another benefit that can be gained is improved security. It's one of many ways to improve scalability without increasing the complexity of maintenance too much. A reverse proxy is a proxy server that is installed in the neighborhood of one or more web servers.

All traffic coming from the Internet and with a destination of one of the web servers goes through the proxy server. The following are several reasons for installing reverse proxy servers:

- Security
- Encryption / SSL acceleration
- Load balancing
- Serve/cache static content
- Extranet Publishing and so on... [5].

This system implements a reversed proxy for protecting local web sites and allowing trusted users to the local web servers. So the additional security features such as URL rewriting and encoding methods and are needed to use for implementing this system.

3.6 BASE 64 encoding used in this system

URL rewriting is the technique used to "translate" a URL like the last one into something the server can understand but the eave droppers cannot understand.

e.g. . .
Input : http://192.168.0.101/
Output: aHR0cDovLzE5Mi4xNjguMC4xMDE=

The system receives a URL as the input and returns an array that contains encoded string by using URL rewriting and encoding method

Base64 is is a method of Encode and decode for Strings, byte arrays, and streams.

It is needed in many places other than its original use as an encoding format for transferring attachments in email. It can be used anytime binary or arbitrary data needs to be represented in common printable characters.

For example to connect to a web page that requires a username and password (basic authentication) user need to Base64 encode the username and password. (See the example)

Example: Using base64 to add a basic authentication to an HTTP request [6].

```
URL url = new URL("http://....");
HttpURLConnection connection =
(HttpURLConnection)url.openConnection();
connection.setRequestProperty(
  "Authorization",
  "Basic " + Base64.encode(
    username + ":" + password
  ) );
InputStream in = connection.getInputStream();
```

- Best For URL applications

Base64 encoding can be helpful when fairly lengthy identifying information is used in an HTTP environment. For example, a database persistence framework for Java objects might use Base64 encoding to encode a relatively large unique id (generally 128-bit UUIDs) into a string for use as an HTTP parameter in HTTP forms or HTTP GET URLs. Also, many applications need to encode binary data in a way that is convenient for inclusion in URLs, including in hidden web form fields, and Base64 is a convenient encoding to render them in not only a compact way, but in a relatively unreadable one when trying to obscure the nature of data from a casual human observer [6].

3.7 MD5 Used in this System for authentication

In cryptography, MD5 (Message-Digest algorithm 5) is a widely used cryptographic hash function with a 128-bit hash value. As an Internet standard (RFC 1321), MD5 has been employed in a wide variety of security applications, and is also commonly used to check the integrity of files. An MD5 hash is typically expressed as a 32-digit hexadecimal number [7].

e.g. .
Input string of password : admin
Output MD5 :
21232f297a57a5a743894a0e4a801fc3

In this system, MD5 hash is used in authentication, When a user login to the system, the user's password coded to MD5 hash and the result hash will be matched with the previous store hash of that user in the user table. If both the hashes are matched, then the user can login to the system successfully.

If the matching of MD5 hashes are failed, the site will display the incorrect password error to the user.

4. System implementation

As everybody knows address of WAN / Internet and local area network (LAN) IP address will not be the same. When a trusted user is outside of LAN, the

user cannot access the LAN web server address directly. So it leads to write a proxy program that can reverse the WAN IP request to LAN web server as shown in figure1.

This reversed proxy services can accept the WAN IP address (e.g. 203.168.0.0/24) then translate and send that address to local web servers (e.g. 192.168.0.0/24).

For reversed proxy user, the user can access this web site any where outside from LAN after authenticating with correct user name and password provided by reversed proxy administrator. The user's request to available LAN web sites are translated and reversed back into the LAN servers.

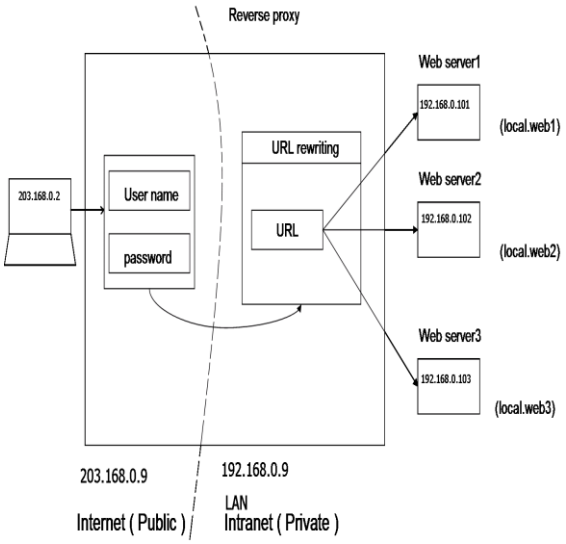

Figure 1. Reversed proxy needs between different IP of WAN and LAN.

Users can also maintain their user profiles and passwords and can also view their own login information including MD5 hash , Login time and their available web links at their home page after login.

For network administrators, it is easy to implement this services on own web server which is connected both Internet and LAN. Unlike the normal reversed proxy which is written in traditional Linux languages and running only in Linux, this web services can be implemented to any apache web server even running on Windows O/S.

Administrators can access this from anywhere in LAN or WAN, any time of the day. Administrators can maintain and watch who is currently online and using this service from which IP address. It is also easy to facilitate in creation of user accounts with MD5 hash and to add available web links info to each user.

In this system, checking logs for usage of users is also available to administrators in Linux console. To provide and complete those checking, URL rewriting - encoding/decoding features and, MD5 hash features are also provided only to administrators.

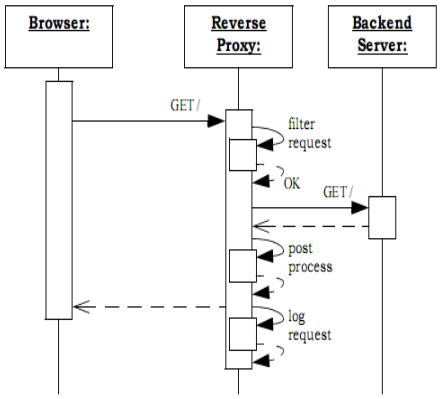

Figure 2. Successful access to Local Web

The above and following figures show the works of reversed proxy system.

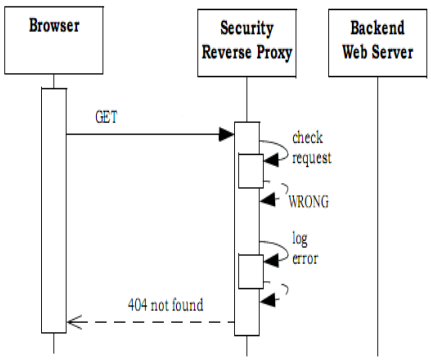

Figure 3. Protected access to Local Web

4.1 System flow diagram

In this section, the detail of the system flow will be described.

Firstly, users must have the public URL address of the reversed proxy web server. If users cannot type that URL correctly, users cannot access the reversed proxy web service.

When users get to the login page after typing correct URL, the system will ask the username and password of the user. User's passwords are hashed with MD5 and check through the user Database.

If the MD5 hash checking is passed, users can access the reversed proxy services according to their access levels and roles.

Users with administrative role can watch others' access to the system and control and check the URL through Web logs file by using URL rewriting decoders.

Normal user can access local web sites sit behind the reversed proxy according to their available web links info and modify their own user profiles.

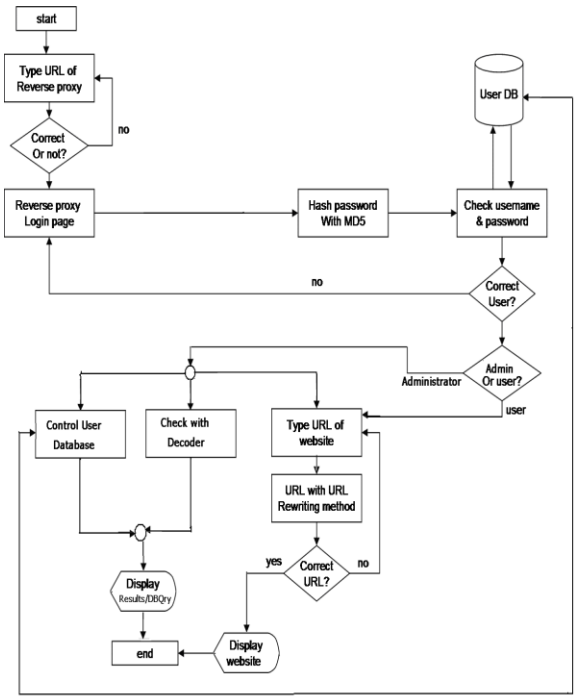

Figure 4. System flow diagram for reversed proxy

The above system flow diagram shows the forwarding process of reversed proxy to local web in step by step.

5. Conclusion

This system provides recommended practices to secure local web applications behind a firewall and leverage access and authentication with the Linux platform products. In this thesis, security is accomplished two ways: Secure reverse proxy, which acts as a stand-in for local web content server to provide an additional barrier between back-end environments and the possibility of malicious attack. Authentication based on MD5 and Secure Socket Layers (SSL) certificates is used to guarantee the identity of the clients requesting data from the back-end systems.

Putting a web server or an application server directly on the Internet gives attackers direct access to any vulnerabilities of the underlying platform (application, web server, libraries, operating system).

However, to provide a useful service to Internet users, access to LAN web server is required. Although a packet filter firewall can shield local web server from attacks on the network level, In addition, this Reverse Proxy web service protects the server software on the application protocol level.

6. Limitation and further extension

This system is aimed to be a cost effective web service which can be running on any apache web server with MYSQL database, so it has some limitations.

Firstly, it was written in web technology and language such as PHP, it cannot access all necessary Linux servers' kernel and system logs. This web system also cannot display the user's current activities on server through in real-time messages at Linux console and X- Window.

Another limitation of the system is MD5 itself, as MD5 is a one way hash widely used in authentication, this system cannot convert to display the user's MD5 hash into readable text for administrative purpose.

As an extension, if this system can be implemented to store each user's browsing behaviors into database and for proving those behaviors to each user as a browsing history and for analyzing purpose to administrators; it will become more effective proxy services for organizations.

7. References

[1] Anh-Duy Nguyen, "Securing Web Applications Through a Secure Reverse Proxy" , Sun ONE Product Technic Support Sun BluePrints™ OnLine—November,2003.

[2] Fredrik Valeur, Giovanni Vigna, Christopher Kruegel, Engin KirdaAn, " Anomaly-driven Reverse Proxy for Web Applications" , SAC 06 Dijon, France, April, 2006.

[3] Jonathan Esterhazy,"Providing Authenticated Access to Web Resources", University of Manitoba Libraries, Winnipeg, Manitoba R3T 2N2, February 1999.

[4] Peter Sommerlad, "Reverse Proxy Patterns", Erlenstrasse , CH-8832 Wollerau, Switzerland, 2003.

[5] Rafael Hecht, and Prof. Joseph Herbst, "Introduction to Proxy Servers", MSIS 640—Data Communications. November 23, 2009.

[6] <http://en.wikipedia.org/BASE64>

[7] <http://en.wikipedia.org/reversed proxy>

[8] <http://whatismyipaddress.com/staticpages/
index.php/how-do-I-use-a-proxy-server>

[9] <http://whatismyipaddress.com/staticpages/
index.php/proxy-server>

[10] <http://www.cs.princeton.edu/courses/archive/spr07/
cos461/web_proxy.html

[11] <http://www.ietf.org/rfc/rfc3040.txt>