International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 12, December - 2013

The Security Considerations with Squid Proxy Server for Windows

Mr.E.Srinivasa Rao [1], Mr.B.Srinu [2]

[1]M.Tech (SE) of IT Department, Gayathri Vidya Parishad College of Engineering, Visakhapatnam. [2]Asst.Professor of IT Department, Gayathri Vidya Parishad College of Engineering, Visakhapatnam.

Abstract

Securing and controlling workstation access to the web has never been an easy task for security professionals. Firewalls and access control list on routers alone may not bring an acceptable level of security for your organization. Even if their primary role is to reduce network traffic and improve performance, HTTP proxy servers (also called cache servers) are likely to be installed as an additional security layer and as a web surfing monitoring system. Having secure proxy servers is critical because many users depend on it for their work. Several proxy server products are available nowadays. Since commercial products are very expensive, alternative products such as open source software are often considered. The most popular and widely used proxy server in this category is indisputably Squid. This paper will cover various security aspects and recommendations to improve Squid's overall security during its installation time. Software configuration parameters and an overview of logging and content filtering software will also be approached.

Keywords: squid log analyzer, Internet service protocols, GUI, Proxy servers.

INTRODUCTION

Squid is a Unix-based proxy server that caches Internet content closer to a requestor than its original point of origin. Squid supports caching of many different kinds of Web objects, including those accessed through HTTP and FTP. Caching frequently requested Web pages, media files and other content accelerates response time and reduces bandwidth congestion. A Squid proxy server is generally installed on a separate server than the Web server with the original files. Squid works by tracking object use over the network. Squid will initially act as an intermediary, simply passing the client's request on to the server and saving a copy of the requested object. If the same client or multiple clients request the same object before it expires from Squid's cache, Squid can then immediately serve it, accelerating the download and saving bandwidth.

Internet Service Providers (ISPs) have used Squid proxy servers since the early 1990s to provide faster download speeds and reduce latency, especially for delivering rich media and streaming video. Website operators frequently will put a Squid proxy server as a content accelerator, caching frequently viewed content and easing loads on Web servers. Content delivery networks [2] and media companies employ Squid proxy servers and deploy them throughout their Networks to improve the experience of viewers requesting programming, particularly for load balancing and handling traffic spikes for popular content. Squid is provided as free, open source software and can be used under the GNU General Public License (GPL) of the Free Software

Foundation. Squid was originally designed to run on Unix-based systems but can also be run on Windows machines. Squid was originally an outgrowth from the Harvest Project, an ARPA-funded open source information gathering and storage tool. "Squid" was the code name used to differentiate the project when development in the new direction was initially begun.

Squid offers a rich access control, authorization and logging environment to develop web proxy and content serving applications. Squid is based on the Harvest Cache Daemon developed in the early 1990's. It was one of two forks from the code base after the Harvest project ran to completion. The Squid project was funded by an NSF grant (NCR-9796082) which covered research into caching technologies. The ir cache funding ran out a few years later and the Squid project continued through volunteer donations and the occasional commercial investment. Squid is currently being developed by a handful of individuals donating their time and effort to building current and next generation content caching and delivery technologies. An ever-growing number of companies use Squid to save on their internet web traffic, improve performance, deliver faster browsing to their end-clients and provide static, dynamic and streaming content to millions of internet users worldwide.

Many corporate companies embedded Squid in their home or office firewall devices; others use Squid in large-scale web proxy installations to speed up broadband and dialup internet access. Squid is being increasingly used in content delivery architectures to deliver static and streaming video/audio to internet users worldwide. The developers of the HTTP protocol identified early on that there was going to be exponential growth in content and, concerned with distribution mechanisms, added powerful caching primitives. These primitives allow content developers and distributors to hint to servers and end-user applications how content should be validated, revalidated and cached. This had the effect of dramatically reducing the amount of bandwidth required to serve content and improved user response times. Squid is one of the projects which grew out of the initial content distribution and caching work in the mid-90s. It has grown to include extra features such as powerful access control, authorization, logging, content distribution/replication, traffic management and shaping and more. It has many, many work-around, new and old, to deal with incomplete and incorrect HTTP implementations.

For ISPs: Save on bandwidth, improve user experience

Squid allows Internet Providers to save on their bandwidth through content caching. Cached content means data is served locally and users will see this through faster download speeds with frequently-used content. A well-tuned proxy server (even without caching!) can improve user speeds purely by optimizing TCP flows. It's easy to tune servers to deal with the wide variety of latencies found on the internet - something that desktop environments just aren't tuned for. Squid allows ISPs to avoid needing to spend large amounts of money on upgrading core equipment and transit links to cope with ever-demanding content growth.

For Websites: Scaling application without massive investment in hardware and development time

Squid is one of the oldest content accelerators, used by thousands of websites around the world to ease the load on their servers. Frequently-seen content is cached by Squid and served to the end-client with only a fraction of the application server load needed normally. Setting up an accelerator in front of an existing website is almost always a quick and simple task with immediate benefits.

For Content Delivery Providers: distributing content worldwide

Squid makes it easy for content distributors and streaming media developers to distribute content worldwide. CDN providers can buy cheap PC hardware running Squid and deploy in strategic locations around the internet to serve enormous amounts of data cheaply and efficiently. A large

number of companies have deployed servers running Squid in the past in exactly this manner.

Web Caching

Web caching is when the server stores web pages and images that have been accessed by clients for future Internet requests. If a user accesses a web site like cnn.com those pages are saved, or cached so that when the next user accesses cnn.com the pages are delivered from the cache not from ccn.com. Of course, the Squid server verifies that the pages have not changed since it stored those pages initially. When viewing logs you will see several terms that need to be understood so you know what is happening on the Squid box. The term **cache hit** is used when the page that was requested actually came from the cache. The **cache hit ratio** is the percentage of requests has been filled from cache. The **byte hit ration** indicates the volume of data that was filled from the cache. A **cache miss** means that the request could not be filled from the cache but had to be filled with an actual connection to the web page. The term **uncatchable** refers to data that could not be cached, either because the instructions from the web server accessed tells Squid not to cache the data or because the settings in Squid itself are set not to cache the specific data format that was requested. For example Squid may be set not to cache large file formats like a QuickTime movie.

Cache validation refers to the testing of the data so that Squid provides information that is current and not stale information. Often before providing a web page Squid will verify the information and replace it if it is out of date. The way that Squid will verify the information is that each time it saves data to the cache a timestamp is placed on it. This use of a timestamp maintains the integrity of updated information. Caching is a way to store requested Internet objects (e.g. data like web pages) available via the HTTP, FTP, and Gopher protocols [3] on a system closer to the requesting site. Web browsers can then use the local Squid cache as a proxy HTTP server, reducing access time as well as bandwidth consumption. This is often useful for Internet service providers to increase speed to their customers, and LANs that share an Internet connection. Because it is also a proxy (i.e. it behaves like a client on behalf of the real client), it can provide some anonymity and security [1]. However, it also can introduce significant privacy concerns as it can log a lot of data including URLs requested, the exact date and time, the name and version of the requester's web browser and operating system, and the referrer.

A client program (e.g. browser) either has to specify explicitly the proxy server it wants to use (typical for ISP customers), or it could be using a proxy without any extra configuration: “transparent caching”, in which case all outgoing HTTP requests are intercepted by Squid and all responses are cached. The latter is typically a corporate set-up (all clients are on the same LAN) and often introduces the privacy concerns mentioned above. Squid has some features that can help anonymize connections, such as disabling or changing specific header fields in a client's HTTP requests.

System Analysis

Problem Description:

Problem statement is one of the basic and important phases of project phase. When the basic problem is determined, it is documented and the symptomatic problem is analyzed, then the current list of basic problem is completed. A system is simply a set of components that interact to accomplish some purpose.

The squid is used for providing the internet access to all the computers in the network. In the corporate networks there will be many users and computers. There should be a monitoring system to monitor the usage of the bandwidth. INTERNE BAND WIDTH MONITORS can also be used for monitoring the band width of the internet connection. The squid generates a log file which contains details of the request which is sent by the client to the proxy server the proxy server stores the client request and the actions of the client in a format. Which are used for analysis and generate the reports. The project is developed with core java and jsp with mysql as back end in windows system.

IMPLEMENTATION

MODULE DESCRIPTION

The project has been divided into four modules based on the functionalities

1. GUI FOR SELECTING LOG FILE
2. INSERTING LOG FILE INTO DATABASE
3. PROCESSING THE DATABASE
4. GENERATION OF REPORTS
    a. REPRESENTING IN TABULAR FORM
    b. REPRESENTING IN GRAPHS

In the above modules the log file is taken as input, it is inserted into the database, processes and analyzes the internet traffic based on the user queries.
Initially a GUI is designed for selecting a log file using java swings; progress bar is also placed while inserting values into the database

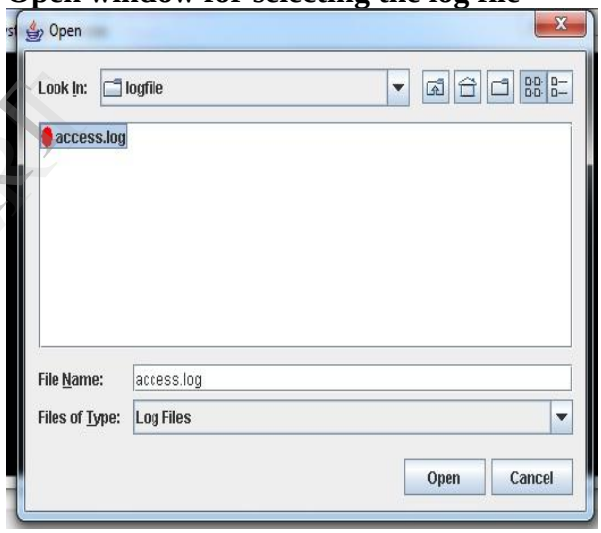
A log consisting of raw format is taken as input which consists of details of the internet traffic from the past one year. The database contains 12 tables which are created dynamically. Data is inserted into the database according to months.

Input is obtained from the user for processing the database in order to get desired output from the admin. Database consists of large number of records. After processing the database based on the user query charts and tables are generated dynamically this results in better understanding of the user and helps in analyzing the internet traffic.

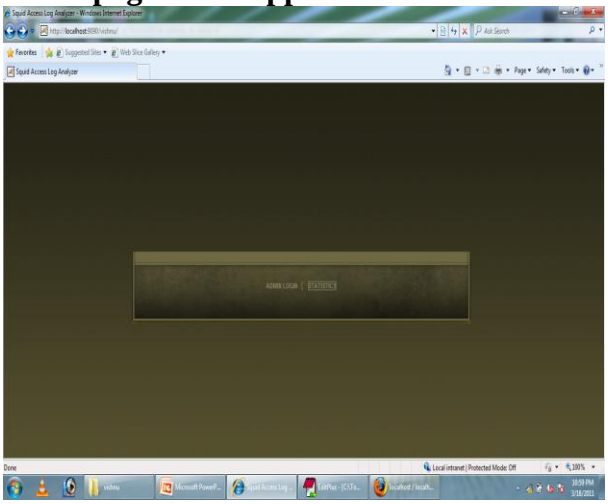
Result Analysis

GUI for selecting the log files for processing

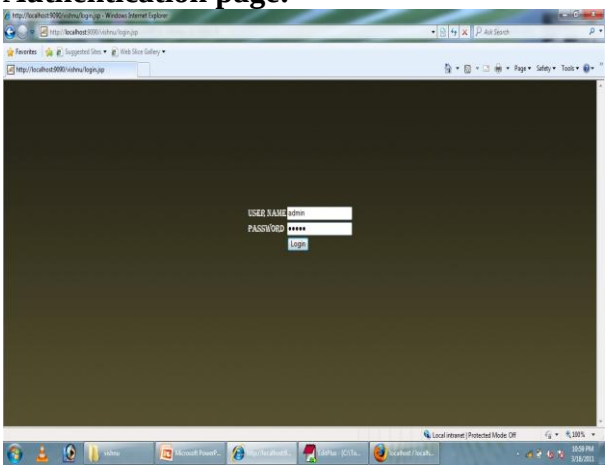
Open window for selecting the log file



Home page of the application:



Authentication page:



Output of query this is taken from the user:

Error page for invalid username and password:

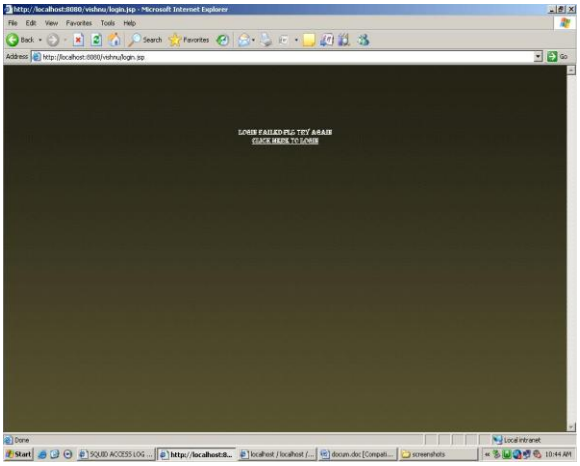
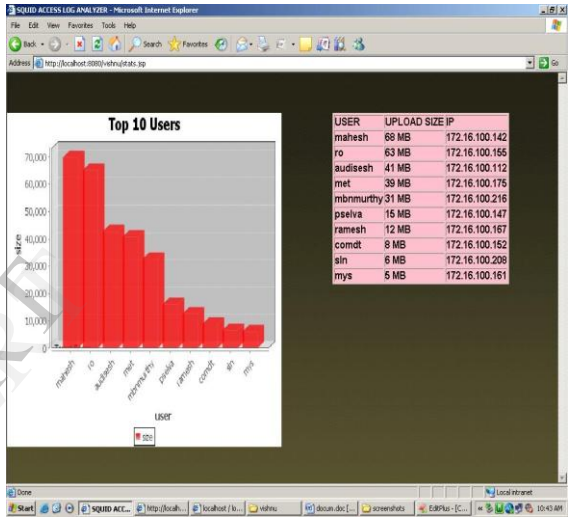
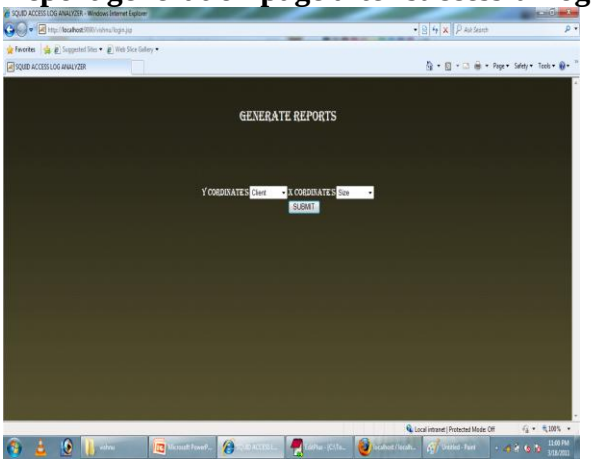


Chart generated for the inputs:



Report generation page after successful login:



CONCLUSION

Squid log analyzer is an application intended for processing the log files which are created by the squid proxy server which helps us in analyzing internet traffic .It is used analyzing log files created, it allows quickly analyzing the efficiency of corporate access to the internet, logging traffic, and generates reports on who visited which sites when.

It is a proxy server which is used to provide internet access to all the computers in a corporate network. It has a wide variety of uses, from speeding up a web server by caching repeated requests; to caching web, DNS and other computer network[4] lookups for a group of people sharing network resources; to aiding security by filtering traffic. By using log file analyzer internet traffic such as traffic distribution by users, traffic distribution by IP addresses,

traffic distribution by protocols, traffic distribution by content type (pictures, video, text, music), traffic distribution by hours, traffic distribution by week days, traffic distribution by dates and months, traffic distribution by visited sites, proxy authentication errors, graphs and tables are generated on the above specified criteria.

References

[1]K.Lavanya,C.NagaRaju"An approach to enhance level of security to the ATM customers by hiding face Biometric data using steganography" has been published in *i-manager's Journal on information Technology, Vol.* 1 *No. 3 June - August 2012*

[2]E.Suresh Babu, C.Nagaraju, MHM Krishna Prasad "An Implementation and Evaluation Study of DSR with DoS Attack in Mobile Ad hoc Networks" has been published in International Organization of Scientific Research (IOSR), August-2013

[3]E.Suresh Babu, C.Nagaraju**,** MHM Krishna Prasad "An Implementation and Performance Evaluation of Passive DoS Attack on AODV Routing Protocol in Mobile Ad hoc Networks" has been published in International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) August, 2013

[4]C.NagaRaju and K.L. Siva shaker Reddy "Abnormality Detection by Generating Random Fields Based on Markov Random Field Theory" has been published in International journal of Computer Science and Network Security VOL.8 No.5, May 2008. Page no260-263

5. HERBERT SCHILDT," The Complete Reference Java J2SE 5 Edition" TATA McGraw – HILL Edition.

6. Java Server Pages –Hans Bergsten SPD O'Reilly

7. ROGERS. PRESSMAN, "SOFTWARE ENGINEERING A practitioner's approach", 6th edition, McGraw – HILL International edition.

8. MYSQL REFERENCE 5.0

9. HERBERT SCHILDT," The Complete Reference JSP" TATA McGraw – HILL Edition

10. Jfree Tutorial

About The Authors

Mr. E. Srinivasa Rao received his B.Tech degree in Information Technology from Laki Reddy Bali Reddy College of Engineering Mylavaram, M.Tech in Software Engineering from Gayathri Vidya Parishad College of Engineering Visakhapatnam.

Mr.B.Srinu received his B. Tech, Computer Science and Engineering with 60.46%: 2005-2009Raghu Engineering College, Visakhapatnam, J.N.T.University Kakinada, Andhra Pradesh ,M.Tech, Information Technology with CGPA 7.91 (in the scale 10): 2009-2011 Tezpur University (A Central University),Tezpur, Assam, Currently, he is working as Asst. professor in Gayathri Vidya Parishad College of Engineering, Visakhapatnam.