DESIGN AND IMPLEMENTATION OF AN INTRUSION TOLERANT WEB SERVER

Sheril Nagoor[1], *K. R. Narasimha Murthy [2]

1- M.Sc. [Engg.] student, 2-Assistant Professor, Department of Computer Engineering,
M. S. Ramaiah School of Advanced Studies, Bangalore – 58.

Abstract

The most widely deployed distributed data servers are web servers, and more web servers are being deployed over the Internet, serving a wide range of both public and private users. Since these servers are exposed over the Internet with the intention of sharing information, the security of the information stored is a major concern. Most of the existing security mechanisms focus on external users who are connecting to servers over the Internet. Apart from external threats, security and intrusion threats from internal networks are also persistent and conventional protective measures are not able to handle such intrusions and security threats.

This paper presents the design and implementation of an intrusion tolerant web server with CIA (Confidentiality, Integrity and Availability) goals. The intrusion tolerant architecture is developed by considering diversity, redundancy, intrusion detection and IP reputation based filtering. Redundancy is added and a load balancer ensures that the server is available to serve users continuously. Diversity is included in the architecture to increase the system's resilience to attacks by use of multiple flavours of operating systems. Integrity is achieved by checking the message headers with the help of an intrusion detection system.

The intrusion tolerant functionality is implemented and tested with two servers running windows and Linux operating systems. A sample file uploading application is developed with reputation based IP filtering. Snort is configured in both servers to detect attacks and alert users. The performance of the developed architecture is compared against standalone server. The web server temporarily blocks requests from the address from which an attack appears to originate. In a 5 minutes 28 seconds long test, the total hits received for the web sites are 106468 with peak transfer speed of 1.1 Mbps, and only 1 page request failed. The application can further be developed as an applet which can be used on all web servers to perform the reputation based blocking.

Key Words: Web Server, Intrusion Detection/Tolerance, Redundancy, Availability, Security.

Abbreviations

ASP	Active Server Pages
CIA	Confidentiality Integrity Availability
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
ID	Identification
IDS	Intrusion Detection System
IIS	Internet Information Services
IP	Internet Protocol
MIME	Multipurpose Internet Mail Extension
PHP	Hypertext Pre-processor
TCP	Transmission Control Protocol
URL	Uniform Resource Locator

1. INTRODUCTION

A Web server is a program that runs over Hyper Text Transfer Protocol which has Client Server model to serve clients with files and other details which are stored on the server. Every computer on the Internet that contains a web site must have a web server program. Two leading web servers are Apache, the most widely installed web server and Microsoft's Internet Information Server. Internet is rapidly growing now and the web server is currently the most widely deployed type of distributed data server. Web servers are providing dynamic content now rather than static content; which was the case during early stages of internet. Dynamic feature of web servers have opened up many security flaws. The major success factor of a web site is to maintain the popularity, reputation and the quality of service observed by users, especially the service availability. A service that is frequently unavailable may have negative effects on the reputation of the web site service provider.

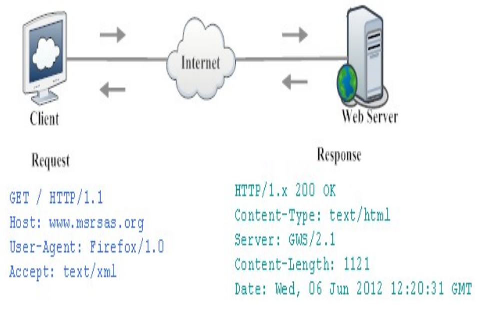
With the development and scope of Cloud Computing there is a tremendous shift in the web hosting industry. Most of the users prefer a server in the cloud due to ease of maintenance and low cost of the infrastructure.
Web server security and availability need to be considered with highest priority considering the threats from internal and external networks. All this while, threats from external network was only considered as threat for an organization, but now since technology is heading towards cloud computing era, internal attacks are equally weighed as external attacks.

This paper presents the design and implementation details of an intrusion tolerant architecture of web servers with diversity, redundancy. The need of considering internal web server security is discussed by comparing enterprise hosting versus cloud hosting. This architecture will help to attain improved service availability and attack tolerance for web server instance specifically.

A web server is a program that accepts connections in order to service requests by sending back responses; its primary function is to serve web pages to its clients via

Hyper Text Transfer Protocol. The Clients are those who most commonly request pages with user agents such as Internet Explorer, Mozilla, and Firefox etc. Connection initiation is always done by the client for information or files stored on the web server. With the HTTP client can submit web forms and upload files on the servers. The behaviour of Web server can be controlled by scripting, for dynamic and interactive behaviour using scripting language such as ASP, PHP and many more. The actual web server softwares are IIS and Apache which are widely used. The only software required at the client side is a web browser which is available by default in all the operating system. A Web page is composed of one or more HTML objects - text, graphics etc. To see a complete Web page, users have to wait until all objects in a page are retrieved from the web servers. A page can be retrieved by various HTTP methods such as a GET method or POST method; these two methods are widely used. GET method is used to retrieve information from web page while POST is used to send information to a web page, for example uploading a file to webserver or commenting on a forum uses POST method.

HEAD, PUT, DELETE, TRACE, CONNECT are some of the other methods to be named. Most widely used in day to day life is GET and POST method.



```
GET / HTTP/1.1
Host: www.msrsas.org
User-Agent: Firefox/1.0
Accept: text/xml
```

```
HTTP/1.x 200 OK
Content-Type: text/html
Server: GWS/2.1
Content-Length: 1121
Date: Wed, 06 Jun 2012 12:20:31 GMT
```

Fig. 1 Overview of HTTP

As shown in Figure 1, the interaction between client and a webserver occurs via HTTP protocol, HTTP is an application level protocol for distributed, collaborative, hypermedia information systems. HTTP has different versions starting from HTTP/0.9. HTTP/1.0, HTTP/1.1. HTTP/1.1 is being used commonly now and its heading towards HTTP/2.0 version. As it proceeded to greater version the reliability is much better. HTTP allows an open ended set of methods and headers that indicate the purpose of a request.

2. NEED FOR SECURITY

With the wide use of web servers, web servers are prime targets for hackers since they are publically available. The data residing on the server should be protected and served only to legitimate users. Each major technological advance in computing raises new security risks and threats that necessitate new security solutions and technology moves quicker than the rate at which such solutions can be developed. With the emergence of cloud computing, there arise related security concerns. The best way to mitigate this security concerns and best hope is to come up with a tolerant architecture with appropriate techniques, a system can be built that provides reasonably high assurance of the effectiveness of its security controls. Web server security is concerned much because almost all the transactions over internet happen with clients and web servers. The unavailability of servers can cause loss of many lakhs of rupees

2.1 Security and Availability Methods and Techniques

There are various methods and techniques available for security and availability. The main methods reviewed for Security are, Intrusion Detection, Proxy, Filtering by IP reputation.Methods for availability are Load Balancing techniques, Diversification in redundancy.

3. PROBLEM DEFINITION

As discussed in the previous sections, security and availability of web servers are important. The data residing on the server should be protected and served only to legitimate users because almost all the transactions over internet happen with clients and web servers. The best way to mitigate this security concerns and best hope is to come up with a tolerant architecture with appropriate techniques. Every business has different needs, different infrastructure but a common needs is to maintain a secured business infrastructure. Basic security needs such as Confidentiality, Integrity and Availability (CIA). CIA is most widely used benchmark for evaluation of information systems.

The aim is to bring up a solution with blend of all these, one of the goal shouldn't fail while importance is given to other, for example when confidentiality is developed it should not impact availability or when integrity is concerned it shouldn't impact confidentiality and availability.

There is need of a robust, attack tolerant webserver which can provide maximum uptime. The solution for this can be achieved by an architecture which has IP filtering for internal traffic, since now a days most of the attacks are from internal user segment, "According to the Computer Security Institute (CSI) in San Francisco, California, approximately 60 to 80 percent of network misuse incidents originate from the inside network."(TechRepublic 2009) and corporate networks do not have ample protects to track down attack from internal subnet. A high available or load balanced architecture which could server customers even though there is an interruption for an server of a server farm and mainly diversity in the platform on which the application runs, so that attacks which could bring down a specific platform may not be able to take down a different one, hence the need of diversity. Here a file uploading application is used as a model since now a days in the support industry file uploading is one of the main webpage used by customers/client where they uploads files, these files are requested by support engineers for troubleshooting and analysis, there are cases where rogue users can upload malicious files since these websites are open and does not have any authentication mechanism, it is open to all.

4. METHODOLOGY

Literature review on existing proxy technique, intrusion detection technique, load balancing algorithms and warning systems was carried out by referring books, journals, technical papers and related documents. High level requirements for adaptive redundancy and intrusion tolerance have been identified based on the literature review. High level design specifications of web server have been developed based on literature review. A web service with distributed components was developed which is platform in depended to support diversity. Functional block diagram of network, attack, intrusion, vulnerability and load balancing blocks was developed using design specifications. High level design specification of all the blocks was translated to low level design and integrated all the blocks. Methods for reputation based user profile creation and countermeasures for internal attacks have been identified. The developed architecture for intrusion tolerance has been tested using various penetration testing techniques and by simulating attacks. Performance analysis of the implemented system was carried out using web performance tool.

4.1 Diversification in redundancy

It is the key technique to withstand an attack, components running on multiple Operating Systems can be a solution where attacks are platform specific, most of the attacks are platform specific since attacks are carried after Operating System fingerprinting during reconnaissance phase of the attack. Diversification means that the redundant components will be implemented with as much diversity as possible. These techniques are to give the system the possibility of continuing its service even if some of its components are compromised. Therefore, while the corrupted components are repaired, the legitimate users continue receiving a correct service, even during attacks. Diversity is achieved by using multiple flavours of Operating Systems, specifically Windows and Linux. So that the attacks which compromise windows machine will not affect Linux machine and vice versa. Any vulnerability affects a particular software on a particular platform and consequently, an attack exploiting a vulnerability is dedicated to one platform and will be inefficient on the others, a buffer overflow exploit for Linux will be ineffective on Windows OS running on x86.

4.2 Load Balancing

Mission critical application or servers should run 24 hours a day, seven days a week and networks need the ability to scale performance to handle large volumes of client requests without creating unwanted delays. To handle large volumes of client requests without creating unwanted delays and when coming to availability load balancing is primary technique available. This technique makes sure that service is not interrupted and the service is available to users continuously (Salchow 2007).

4.3 Filtering by IP reputation

This method is used to mitigate mainly attacks from trusted network. With this idea, it is possible to keep track of requests made to server, the server can track the requestors IP address and store it in server's database as a list. Based on legitimacy and suspicious activity or requests the server can move the requestor IP address to a Black List and White List. Black List is for blocking an IP and server will not serve the requestor IP for a particular amount of time in case it is identified as attacking source. The list requirement of keeping the status of senders and retrieving special name server records in order to verify existing trust relationships before accepting a request is also covered under this technique. The output of IP address reputation can be used for effective and accurate attack filtration in various ways (Akella, Esquivel and Mori 2010).

4.4 Intrusion-detection system

IDS analyses network traffic and the state of the web servers and report suspected intrusions. Some IDS modules execute on one or more dedicated hardware platforms or external to servers, while others reside in the web servers. There are Network IDS and Host Based IDS. Intrusion Detection Systems feature a wide diversity of event sources, inference techniques and detection paradigms. They include host, network and protocol monitors. Different sensors cover different portions of the detection space and have different detection rates, false alarm ratios and operational conditions, for e.g. the maximum rate of incoming events that can be handled. Their combination allows detecting more known attacks as well as anomalies arising from unknown ones. The advantages of heterogeneous sensors come at the cost of an increased number of alerts. To effectively manage them, they must be aggregated and correlated (Valdes 2002).

5. IMPLEMENTATION

Figure 2 shows how the traffic flow of the proposed tolerant architecture is, even if the access is from internal or external, the requests go via load balancer and detection systems.

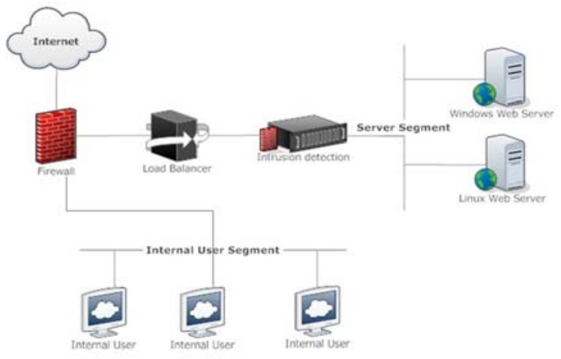


Fig. 2 Network Architecture

The reputation based IP filtering mechanism is built within the webserver, the sample application which runs in the webserver will have the code to interpret if the request is an attack or not, based on that the IP will be blacklisted and log entry is made. If the request is legitimate a simple log entry is made with the IP address of requesting client and action performed. Web Application is platform independent hence the possibility of running in diverse platform.

As shown in Figure 3, the architecture is split up into blocks which perform specialized task such as load balancing the request in case a server is unavailable or is busy with too many requests and stop serving further client requests, Intrusion block to identify the attacks. Attack block/ Vulnerability block to check the type for files user uploads and to verify the user requests and finally an actual web server to serve the clients.

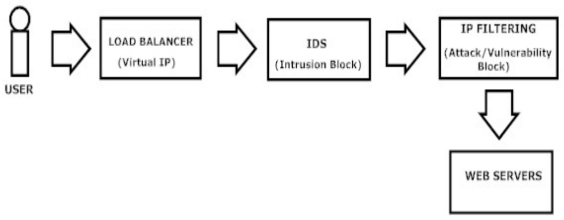

USER
LOAD BALANCER (Virtual IP)
IDS (Intrusion Block)
IP FILTERING (Attack/Vulnerability Block)
WEB SERVERS

Fig. 3 Block Diagram

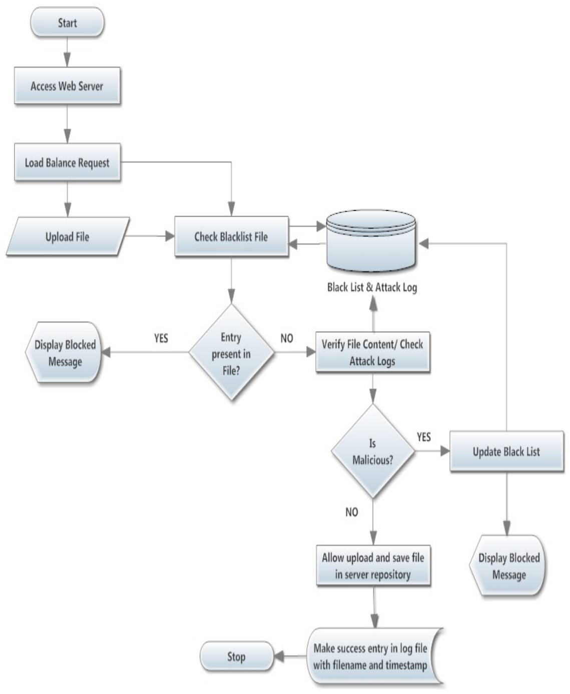


Fig. 4 High level Flowchart

Figure 3 shows flowchart, the attack prevention/tolerance happens at the beginning itself i.e. once the user requests a webpage, here a sample file uploader application is used.

The request will be load balanced based on availability of webserver, the request can go to either windows or Linux servers, upon receiving request the first check of security is initiated which is to cross verify if the requester’s IP address is present in the blacklist and attack logs generated by IDS; if his reputation is good or not. If the requester’s IP is found on the blacklist or attack logs then an error page is served, if not; next check is for file content to verify if it is valid file extension. If found suspicious, again the server throws a block page to user and makes an entry to the blacklist file; the entry remains for an amount of time. Time out to unblock the user can be set, if the file is clean an entry to a log file is made with time, date, IP address of client and file name and the files are stored to server’s repository

The first prototype has been completed, focusing on the reputation based IP filtering mechanism, Web server deployment, IDS and load balancer. The web servers used are Apache apache-tomcat-6.0.36 running under Windows 7 and Ubuntu 11.10 i386, and compatible browsers are Internet Explorer 8 or Mozilla Firefox for better user interface experience. Figure 3 shows the main components of implementation.

Glancing into the core part of the work carried; the web application for file uploader is implemented using servlet; Servlets are the java technology for extending and enhancing web servers. Servlets are platform independent, component based method for building web based applications. The moment user requests the web page; servlet is invoked and the security validations are initiated. The servlet performs series of security checks such as verifying the reputation of requestor’s IP, valid file type and checks the attacks detection logs before serving the requestor. The servlet also invokes functions to write log entries, make entries into black list file and to throw exceptions when required. It also performs the upload action for a clean request by invoking appropriate functions.

Snort is installed in both Windows and Linux platform , the configuration are done in snort.conf file, this file need to be tuned as per the requirement and fine-tuned for avoiding false alerts.

Web service availability deals with the readiness for correct service. It can be viewed as a function A(t), which is the possibility that the system is operational and delivers the correct service at instant of time t.

A system fails to deliver correct service due to the following reasons:

- Due to any faults caused by system.
- Server overloaded condition, i.e. the server becomes busy and it is not able to deliver a correct service or server new requests.

The architecture does not prevent system from hardware and software faults, load balancer supports by sending the traffic to next available server so that the user experience is positive and good. The load balancing setup is simple; users will hit the virtual IP address created on the load balancer and the load balancer diverts the request based on the availability and health of the servers. Figure 5 depicts high level view of how load balancing is done; load balancer listens on port 80 while the web servers are listening for connection on port 8080.

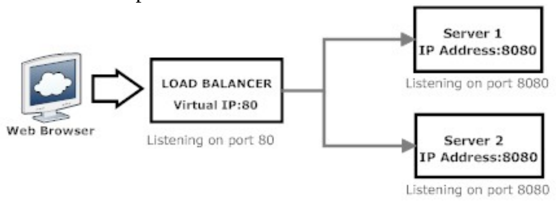


Fig. 5 Load balancer block diagram

The first thing to perform as part of implementing load balancer is to create the server pool, add the IP address of the web servers which is hosted before and which will serve the actual content, to the pool and mention the port number the servers are listening.

6. RESULTS AND DISCUSSIONS

Table 1 shows security mechanisms and its intended functions which was used for developed architecture. The functionality of the developed architecture was analyzed, with single client and multiple clients, including a mobile phone device. The application ran successfully in both the operating systems without failure and produced all the expected results.

IDS fired alerts up-on suspicious traffic, load balancer performed load balancing and diversion of traffic to available server when either of the servers was unavailable. Reputation based functionality also worked as expected and blocked the attacking hosts temporarily.

Table 1. Security mechanism and functions

Security Mechanism	*Intended Function*
Diversity of OS Platforms	Unlikely to be vulnerable to common attacks and failure modes aimed towards a specific platform.
IDS	Analyses the network traffic. Reports suspected intrusions.
Load Balancing	Checks the state of the servers. Diverts traffic to available server.
Reputation Based IP Filtering	Temporarily blocks requests from the address from which an attack appears to originate.
Regular Reboots	Prevents unreliability due to software aging. Defeats long-term, incremental intrusions.

6.1 Performance Analysis

Load test was carried on a single webserver for comparing the performance, the performance degraded considerably, there were page failures. Table 2 shows the test results and analysis for single/ standalone web server. Table 2 shows the summary of the test performed for standalone server.

Table 2. Top level Metrics for Standalone Server

Summary	Values
Start Time	2/19/2013 16:40
Duration	0:06:16
Completed Pages	73,195
Total hits	73,177
Peak hits/sec	1339.5
Peak transfer speed	1.1 Mbps
Total Pages Failed	18

Number of users were 25, the duration of the test was 6 minutes 16 seconds, total hits received for the web sites are 73177 with peak transfer speed of 1.1 Mbps, and 18 pages were failed. This test was done using web performance tool.

Figure 6 point out URL completion Rate Graph for standalone server. It shows total number of HTTP transactions (URLs) completed per second relative to the elapsed test time. In a well-performing system, this number should scale linearly with the applied load, here hits per second is dropping as the user counts increases.

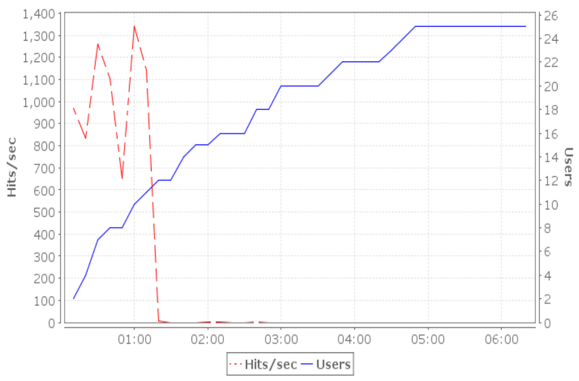


Fig. 6 Standalone Server - URL Completion Rate Graph

Dotted lines represent hits per second and line represents number of users.

Figure 7 shows the failures section chart; it illustrates how the total number of page failures and the page failure rate changed throughout the test relative to the elapsed test time. In a good system or architecture the page failure should be minimal.

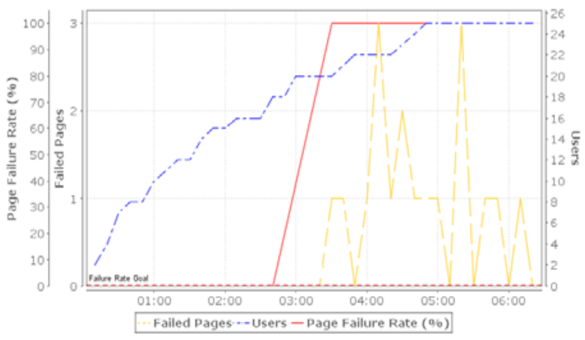


Fig. 7 Standalone Server - Page Failure Graph

Figure 8 shows the bandwidth consumption graph for standalone server, outgoing bandwidth refers to data sent by the server to the browser. Outgoing bandwidth is dropping as the load /users increase. In a good system/ architecture the bandwidth-in and bandwidth-out should scale linearly with the applied load i.e. users.

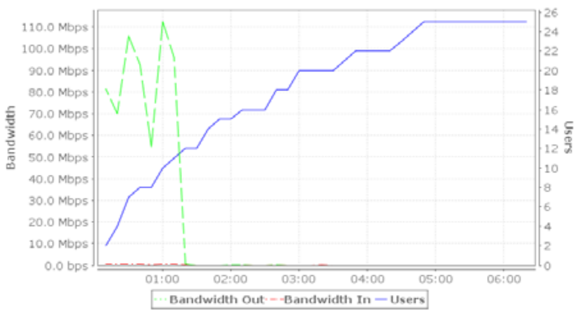


Fig. 8 Standalone Server - Bandwidth Consumption Graph

Performance of the proposed intrusion tolerant architecture is measured by time based performance and load tests by performing the tests multiple times and took the average of the results. Table 3 shows top level metrics of the overall performance of the system; Maximum user count was 25 users for the testing.

Table 3. Top level Metrics for Intrusion Tolerant Architecture

Summary	Values
Start Time	2/19/2013 15:40
Duration	0:05:28
Completed Pages	560
Total hits	106,468
Peak hits/sec	877.6
Peak transfer speed	1.1 Mbps
Total Pages Failed	1

The duration of the test was 5 minutes 28 seconds, total hits received for the web sites are 106468 with peak transfer speed of 1.1 Mbps, and only 1 page was failed. Figure 9 shows URL completion rate, which shows statistics of the total number of HTTP transaction completed per second relative to the test time interval. Figure 9 show the hits per sec and user's statistics along with the time took for testing. Here as the users increase hits decreases. The tests were performed with a tool named web performance tester.

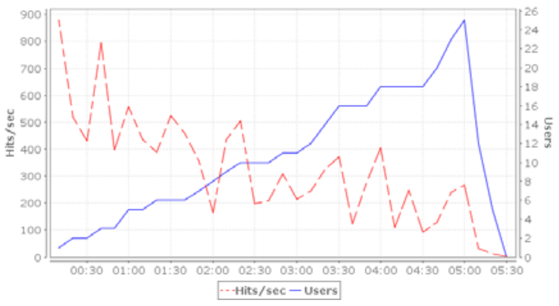


Fig. 9 Intrusion Tolerant Architecture - URL Completion Rate Graph

The dotted lines represent hits per second data and line represents users.

A failure analysis graph was also plotted which shows the total number of page failures and page failure rate changed through the test as shown in Figure 10.

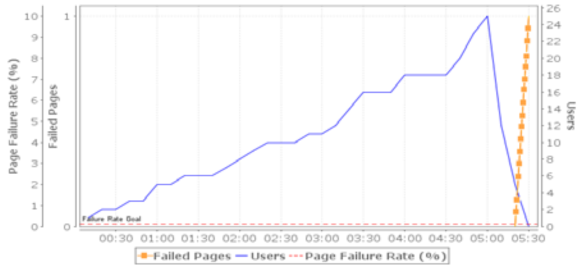


Fig. 10 Intrusion Tolerant Architecture - Page Failure Graph

A page failure could occur due to multiple reasons including network timeout issue, or server too busy. The graph shows only a single failure which showcases the robustness of the architecture. The lesser number of failures means good system performance.

Next analysis is for bandwidth consumption, the graph shows the total bandwidth consumed by the traffic generated.

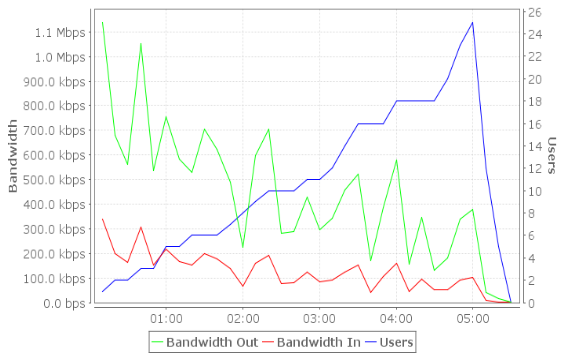


Fig. 11 Intrusion Tolerant Architecture - Bandwidth Consumption Graph

Figure 11 shows the bandwidth consumption graph. The bandwidth consumption is described in terms of the servers; i.e. outgoing bandwidth refers to data sent by the server to the client/browser. Notice that the outgoing bandwidth is scaling linearly with the number of users.

By observing the graphs plotted for standalone server and proposed intrusion tolerant architecture, it is obvious that proposed intrusion tolerant architecture for webserver is performing well and reliable. The architecture is able to withstand attacks and same time available for users with the service it is running whereas the standalone server fails to withstand attacks and fails to serve users under attacks situations and heavy load.

7. CONCLUSIONS

The proposed Intrusion tolerant architecture combines a variety of traditional security mechanisms and reputation based IP filtering for devastating internal attacks up to a certain level. The system is tolerant to attacks and acts upon intrusive attempts. Availability with security is achieved with this architecture. The tests yielded a qualitative evaluation of the deployed intrusion-tolerant mechanisms. The intrusion tolerant functionality is implemented and tested with two servers running windows and Linux operating systems, a sample file uploading application is developed in java with reputation based IP filtering. Web service is hosted using Apache Tomcat web servers. Snort is configured in both servers to detect and alert the attacks. Load balancer makes sure that the server is available and serves the users continuously. Performance testing was done for the developed architecture and compared the results with standalone server

- Implementing reputation based IP address filtering provided an effective solution as countermeasure for internal attacks.
- Implementing attack tolerant architecture by redundancy and diversification provides performance, reliability and availability with security.
- Comparison of the developed architecture was done with a standalone server and graphs were plotted, in which it was noticed that failure rate is much higher in standalone server.

8. REFERENCES

Akella, A., Esquivel, H., and Mori, T. (2010) ‘On the Effectiveness of IP Reputation for Spam Filtering’. *Second International Conference on Communication Systems and Networks*. held in 5-9 January 2010 at Bangalore. New Jersey: IEEE Press

Kassner, M. (2009) *Myth or Not: Most Security Breaches Originate Internally* [online] available from <http://www.techrepublic.com/blog/security/myth-or-not-most-security-breaches-originate-internally/1606> [21 October 2012]

Salchow, K. J. (2007) *Load Balancing 101: Nuts and Bolts*, White Paper, F5 Networks, Inc, [online] available from <http://www.f5.com/pdf/white-papers/load-balancing101-wp.pdf> [20 June 2012]

Valdes, A. (2002) ‘An Adaptive Intrusion-Tolerant Server Architecture’. *10th International Workshop on Security Protocols*. held 17-19 April 2002 at Cambridge. Berlin: Springer Heidelberg