IMPLEMENTATION OF INTRUSION TOLERANT ARCHITECTURE FOR WEB SERVER

E.Hima Bindu¹, Dr. R. V. Krishnaiah²

¹(MTech-CS, D.R.K.Institute of science and technology, Hyderabad, India)
²(Principal, Dept of CSE, D.R.K Institute of and technology, Hyd, India.)

ABSTRACT
This paper presents implementation of intrusion-tolerant architecture for Web servers. Now a day, more and more information systems are connected to the Internet and offer Web interfaces to the general public or to a restricted set of users. Sucopenness makes them likely targets for intruders, and conventional protection techniques have been shown insufficient to prevent all intrusions in such open systems. To prevent these attacks we use intrusion-tolerant architecture for Web servers. This is based on redundancy and diversification principles in order to increase the system resilience to attacks. Usually an attack targets particular software, running on a particular platform, and fails on others. it is composed of redundant proxies that mediate client requests to a redundant bank of diversified application server. And it provides same service by taking example of Tourist management system. And it also checks the server security using MD5 hash algorithm.

Keywords: Intrusion tolerance, MD5 security, web server.

INTRODUCTION
This paper presents the implementation of intrusion tolerance architecture for web server. This paper suggests that how to reduce the burden of overloaded sever to other available server. This is based on redundancy and diversification principle which increases system availability and integrity .it also provide correct service to the user even in the presence of intrusion. To make server secure we are using MD5 security algorithm [1].
LITERATURE REVIEW
Today in this world everyone agrees now that the Internet has become essential in everyday life. People use the Internet to work, to exchange information, to make purchases, etc. This growth of the Internet use has unfortunately been accompanied by a growth of malicious activity in the Internet. More vulnerability is discovered, and nearly every day [2] new security advisories are published. Almost all traditional approaches for building secure systems only focus on preventing attacks to be successful. Such approaches are becoming insufficient when used in the context of open networks like the Internet, which are characterized by frequent appearance of new attacks. For this purpose, we propose in this paper the implementation of intrusion tolerance architecture for web server. Here we address, in particular, servers publishing information that is not confidential (that is, public access Web server COTS [6])

To make the server intrusion tolerant, we propose to use diversification in redundancy. By diversification, we mean that the redundant components will be implemented with as much diversity as possible.

We use these techniques to give the

Intrusion Tolerant web Server
This section gives the details of intrusion tolerant web server architecture.

Firewall: It cannot stop all the attacks, but an attacker has no means of modifying the firewall configuration. At any given moment, all the correct servers return the same correct answers to the same request.

Proxy : To manage redundancy and make it transparent to the Clients, tolerance proxies are included in our architecture, these proxies have an essential function in the intrusion tolerance policy (ITP). They
mediate client requests, monitor the state of the Web servers. These proxies are also diversified.

**Leader**: The leader is responsible for receiving, filtering, and forwarding client requests to one or more Web servers according to the current regime. The leader impersonates all the Web servers from the client point of view. The IP address of the leader IPL is the only public IP address, and it is considered as the IP address of the virtual Web server. The leader is also responsible for balancing the load among the Web servers. All decisions taken by the leader to manage the redundant Web servers must be consistent with the current regime and security policy. The leader receives the replies from the Web servers and, according to an agreement protocol (AP), decides the response to return to the client. If the responses are not consistent, the leader is declared corrupted, then a proxy creates silicate process and redirect to other available server and provides same service to the user.

**WebServers: Redundancy and Diversification**

The architecture is based on the principles of redundancy and diversification. Redundancy is used to increase independence between redundant subsystems from the attacker’s point of view. Since most attacks take advantage of specific vulnerabilities in a particular OS, application software, or hardware platform, they are, in general, ineffective on others. So, the deployment of a redundant bank of servers (hardware/OS/software Diversification level) should allow the system to continue providing acceptable service to users, even if parts of the system are corrupted. The Web servers provide the same services (that is, they typically return exactly the same HTML pages) but are running different application software.

**Adaptive Redundancy Level**

In order to minimize the performance degradation of the system, the notion of adapted redundancy is introduced. The regime is the number of Web servers that process each client request. If the architecture includes N Web servers and if the regime is n, it means that only n among N servers process the request.

**Detection Mechanism**

The effectiveness of the intrusion-tolerant approach is strongly dependent on the efficiency of the detection mechanisms used in the architecture.
Agreement Protocol

The AP is one of the most reliable detection mechanism deployed in the architecture. It is used to validate server responses when the system is running in a non simplex regime and is executed by the leader. The assumption is that only a minority of servers can be corrupted at the same time, thanks to diversification. Thus, the AP should find a consensus between the servers.[7] If the leader receives different responses, it considers that the servers among the majority are correct and the others are corrupted. Actually, the servers do not send the whole reply (an html file for example) to the leader. Instead, they compute a cryptographic hash code (the MD5 algorithm) of the response (except the head part) and send this hash code to the leader. The AP is then executed on the hash codes. The complete response is sent by only one of the servers after a consensus has been reached.

In some particular situations [6], the html replies may differ, even in the body part http error codes, dynamic Web pages that include the name of the server, or the date for example. In such cases, either a preprocessing of the Web pages is necessary so that they are converted into a canonical format, independent of the Web server, or the corresponding http requests are forbidden and rejected.

The detection by the AP (comparison or majority voting) is executed by the leader only, as in a centralized scheme. This is sufficient to tolerate practically all intrusions affecting a minority of servers. The AP is also used to make a common decision between the proxies when an anomaly is detected in the system.

IMPLEMENTATION

In this section we will describe the implementation technique for redirecting the overloaded server and security analysis by using MD5 hash algorithm. It also describes the database access for connection between leader and the web server. Security analysis for MD5 is shown in below Figure2.
RESULT

The following graphs indicating the number of redirection Vs number of Requests. In fig 3 showing the redirection graph for number of requests 5. In fig 4 showing the redirection for number of requests 10.

CONCLUSION

From these entire graphs we conclude that numbers of redirection and request values are varied.[8] So that we have taken number of requests as 10. Because we have to show the reducing server burden and security of the server.
1. To build secure system using COTS components those are likely to include vulnerabilities.

2. To achieve high integrity and availability requirements.

3. To find a good trade-off between security and performance.

REFERENCES


Mrs. E.Hima Bindu is a research scholar in DRK Institute of Science & Technology.

Dr. R. V. Krishnaiah has received Ph.D from JNTU Ananthapur, M.Tech in CSE from JNTU Hyderabad, M.Tech in EIE from NIT (former REC) Warangal and B.Tech in ECE from Bapatla Engineering College. He has published more than 50 papers in various journals. His interest includes Data Mining, Software Engineering, Image Processing.