International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 08 | Aug-2016 www.irjet.net p-ISSN: 2395-0072

# Survey On Web Server Security and Web Application Security

**Vishnu Vardhana Rao [1] , S.V Anil [2] , Suraj Alamoni [3], Navya reddy[4]**

*Department of computer science engineering*
*Vignan's institute of technology and aeronautical engineering*
*Vignans hills, deshmukhi(V), nalgonda (Dist),Telangana(State)*

---***---

**Abstract -** A web server is a computer host configured and connected to Internet, for serving the web pages on request. Information on the public web server is accessed by anyone and anywhere on the Internet. Since web servers are open to public access they can be subjected to attempts by hackers to compromise the server's security. Hackers can deface websites and steal data valuable data from systems. This can translate into significant loss of revenue if it is a financial institution or e-commerce site. In the case of corporate or government systems, loss of important data means launch of information espionages or information warfare on their sites. Apart from data loss or theft, web defacement can also result in significant damage to the image of company [1]. The fact that an attacker can strike remotely makes a Web server an appealing target. Understanding threats to Web server and being able to identify appropriate countermeasures permits to anticipate many attacks and thwart the ever-growing numbers of attackers [3]. This work begins by reviewing the most common threats that affect Web servers. It then uses this perspective to find certain countermeasures. A key concept of this work focuses on the survey of most prevailing attacks that occurs due to certain vulnerabilities present in the web technology or programming which are exploited by attackers and also presents general countermeasures. In addition, various methods to detect and prevent those attacks are discussed and highlighted the summary and comparative analysis of the approaches on the basis of different attacks that shows you how to improve Web server's security.

***Key Words :*** SQLIA (SQL Injection Attack), XSS (Cross Site Scripting), CSRF (Cross Site Request Forgery), OWASP (Open Web Application Security Project), vulnerabilities, Countermeasures

## 1 .INTRODUCTION

A secure Web server provides a protected foundation for hosting Web applications, and Web server configuration plays a critical role in Web application's security. Badly configured virtual directories, a common mistake, can lead to unauthorized access. A forgotten share can provide a convenient back door, while an overlooked port can be an attacker's front door. Neglected user accounts can permit an attacker to slip by your defenses unnoticed. The fact that an attacker can strike remotely makes a Web server an appealing target. Understanding threats to Web server and being able to identify appropriate countermeasures permits to anticipate many attacks and thwart the ever-growing numbers of attackers [3]. A. Threats to Web Server and Countermeasures The main threats to a Web server are [3]:

1] Profiling
2] Denial of service
3] Unauthorized access
4] Arbitrary code execution
5] Elevation of privileges
6] Viruses, worms, and Trojan horses

### 1.1) Profiling

Profiling, or host enumeration, is an exploratory process used to gather information about your Web site. An attacker uses this information to attack known weak points.

Vulnerabilities :

Common vulnerabilities that make your server susceptible to profiling include:

- Unnecessary protocols
- Open ports
- Web servers providing configuration information in banners

Attacks :

Common attacks used for profiling include:

- Port scans
- Ping sweeps
- NetBIOS and server message block (SMB) enumeration

Countermeasures :

Countermeasures include blocking all unnecessary ports, blocking Internet Control Message Protocol (ICMP) traffic, and disabling unnecessary protocols such as NetBIOS and SMB

### 1.2) Denial of Service

Denial of service attacks occur when your server is overwhelmed by service requests. The threat is that Web server will be too overwhelmed to respond to legitimate client requests.

Vulnerabilities :
Vulnerabilities that increase the opportunities for denial of service include:

- Weak TCP/IP stack configuration
- Unpatched servers

Attacks :
Common denial of service attacks include:

- Network-level SYN floods
- Buffer overflows
- Flooding the Web server with requests from distributed locations

Countermeasures
Countermeasures include hardening the TCP/IP stack and consistently applying the latest software patches and updates to system software.

**1.3) Unauthorized Access:**
Unauthorized access occurs when a user without correct permissions gains access to restricted information or performs a restricted operation.
Vulnerabilities:
Common vulnerabilities that lead to unauthorized access include:

- Weak IIS Web access controls including Web permissions
- Weak NTFS permissions

Countermeasures
Countermeasures include using secure Web permissions, NTFS permissions, and .NET Framework access control mechanisms including URL authorization.

1.4) **Arbitrary Code Execution**:
Code execution attacks occur when an attacker runs malicious code on your server either to compromise server resources or to mount additional attacks against downstream systems.
Vulnerabilities :
Vulnerabilities that can lead to malicious code execution include:

- Weak IIS configuration
- Unpatched servers

Attacks
Common code execution attacks include:

- Path traversal
- Buffer overflow leading to code injection

Countermeasures
Countermeasures include configuring IIS to reject URLs with "../" to prevent path traversal, locking down system commands and utilities with restrictive access control lists (ACLs), and installing new patches and updates.

1.5) **Elevation of Privileges**:
Elevation of privilege attacks occur when an attacker runs code by using a privileged process account.
Vulnerabilities
Common vulnerabilities that make your Web server susceptible to elevation of privilege attacks include:

- Over-privileged process accounts
- Over-privileged service accounts

Countermeasures
Countermeasures include running processes using least privileged accounts and using least privileged service and user accounts.

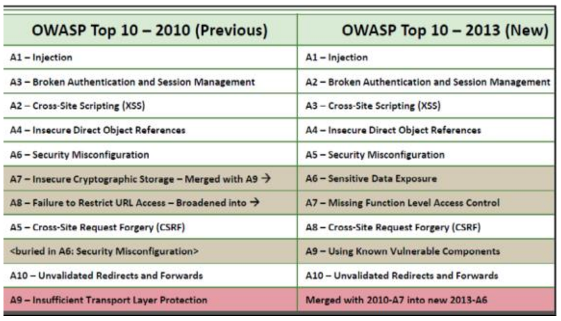
1.6) **Viruses, Worms, and Trojan Horses**:
Malicious code comes in several varieties, including:
Viruses:Programs that are designed to perform malicious acts and cause disruption to an operating system or applications.
Worms: Programs that are self-replicating and self-sustaining.

Trojan horses:Programs that appear to be useful but that actually do damage.
Vulnerabilities
Common vulnerabilities that make you susceptible to viruses, worms, and Trojan horses include:

- Unpatched servers
- Running unnecessary services

Countermeasures
Countermeasures include the prompt application of the latest software patches, disabling unused functionality such as unused ISAPI filters and extensions, and running processes with least privileged accounts to reduce the scope of damage in the event of a compromise [3].

## 2. OWASP TOP 10 WEB SECURITY THREATS



| OWASP Top 10 – 2010 (Previous) | OWASP Top 10 – 2013 (New) |
|---|---|
| A1 – Injection | A1 – Injection |
| A3 – Broken Authentication and Session Management | A2 – Broken Authentication and Session Management |
| A2 – Cross-Site Scripting (XSS) | A3 – Cross-Site Scripting (XSS) |
| A4 – Insecure Direct Object References | A4 – Insecure Direct Object References |
| A6 – Security Misconfiguration | A5 – Security Misconfiguration |
| A7 – Insecure Cryptographic Storage – Merged with A9 → | A6 – Sensitive Data Exposure |
| A8 – Failure to Restrict URL Access – Broadened into → | A7 – Missing Function Level Access Control |
| A5 – Cross-Site Request Forgery (CSRF) | A8 – Cross-Site Request Forgery (CSRF) |
| <buried in A6: Security Misconfiguration> | A9 – Using Known Vulnerable Components |
| A10 – Unvalidated Redirects and Forwards | A10 – Unvalidated Redirects and Forwards |
| A9 – Insufficient Transport Layer Protection | Merged with 2010-A7 into new 2013-A6 |

Fig. 1 OWASP Top 10 Web Application Security Threats
In this survey, we are focusing only on 3 most prevalent attacks which are occurring most frequently

## 3. SQLIA

Injection flaws, such as SQL, injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker‘s hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization [2].

An example of simple tautology type of SQLIA is shown in figure 1, which will result in displaying all the records in the database irrespective of wrong username and password because the condition 0=0 always evaluates to true.

Some of the most commonly followed prevention mechanism for SQLIA are as follows [4] :

1 **Use prepared statements**
2 **Perform Input Validation**
3 **Escape all user supplied input**
4 **Enforce least privilege**
5 **Use stored procedures**

## 4. XSS

XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim‘s browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites. [2]

**Example Attack Scenario**:

The application uses untrusted data in the construction of the following HTML snippet without validation or escaping:

(String) page += "<input name='creditcard' type='TEXT_ value='" + request.getParameter ("CC") + "'>";

The attacker modifies the ‗CC‘ parameter in his browser to:

'><script>document.location='http://www.attacker.com/cgi - bin/cookie.cgi? foo='+document.cookie</script>'.

This causes the victim‘s session ID to be sent to the attacker‘s website, allowing the attacker to hijack the user‘s current session. [2]

**Types of XSS**

1) **Stored XSS**: Stored attacks are those where the injected script is permanently stored on the target servers, such as in a database, in a message forum, visitor log, comment field, etc. The victim then retrieves the malicious script from the server when it requests the stored information. Stored XSS is also sometimes referred to as Persistent or Type-I XSS [5].

2) **Reflected XSS**:

Reflected attacks are those where the injected script is reflected off the web server, such as in an error message, search result, or any other response that includes some or all of the input sent to the server as part of the request. Reflected attacks are delivered to victims via another route, such as in an e-mail message, or on some other web site. Reflected XSS is also sometimes referred to as Non-Persistent or Type-II XSS. [5]

## 5. CSRF

A CSRF attack forces a logged-on victim‘s browser to send a forged HTTP request, including the victim‘s session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim‘s browser to generate requests to the vulnerable application thinks are legitimate requests from the victim. [2]

The differences between XSS and CSRF

Though CSRF seems similar to (XSS) at first, both are completely different attack vectors.Where XSS aims at inserting active code in an HTML document to either abuse client-side active scripting holes, or to send privileged information (e.g., authentication/session cookies) to an unknown evil website, CSRF aims to perform unwanted actions on a website where the victim has some prior relationship and authority.

Moreover, where XSS sought to steal your online trading cookies so an attacker could manipulate a victim‘s account, CSRF seeks to use the victims‘ cookies to force them to execute a trade without their knowledge or consent.

While XSS attacks exploits the trust that a user has on the website, CSRF attacks exploit the trust that the website has in its user. [6]

**Types of CSRF attacks**

1 Reflected CSRF attacks

In a reflected CSRF attack, the attacker uses a system outside the application to expose the victim to the exploit link or content. This can be done using a blog, an email message, an instant message, a message-board posting, [6].

2. Local/stored CSRF attacks

A stored/local CSRF attack is one where the attacker can use the application itself to provide the victim the exploit link, or other content which directs the victim‘s browser to perform attacker-controlled actions in the application.

## 6. CONCLUSIONS

To help organizations and developers reduce their application security risks in a cost effective manner, OWASP has produced numerous free and open resources that you can use to address application security in your organization. The following are some of the many resources OWASP has produced to help organizations produce secure web applications. [2]

**Guidelines for Developers:**

**Establish & Use Repeatable Security Processes and Standard Security Controls**

1) Application Security Requirements:

To produce a secure web application, developers must define what secure means for that application. OWASP recommends to use the OWASP Application Security Verification Standard (ASVS), as a guide for setting the security requirements for application(s).

2)Application Security Architecture**:**

Rather than retrofitting security into applications, it is far more cost effective to design the security in from the start. OWASP recommends the OWASP Developer‘s Guide, and the

OWASP Prevention Cheat Sheets as good starting points for guidance on how to design security in from the beginning
3) Standard Security Controls:
Building strong and usable security controls is exceptionally difficult. A set of standard security controls radically simplifies the development of secure applications. OWASP recommends the OWASP Enterprise Security API (ESAPI) project as a model for the security APIs needed to produce secure web applications. ESAPI provides reference implementations in Java, .NET, PHP, Classic ASP, Python, and Cold Fusion.
4) Application Security Education:
The OWASP Education Project provides training materials to help educate developers on web application security and has compiled a large list of OWASP Educational Presentations.

**Guidelines for Verifier's:**

1. Code Review:

OWASP has produced the OWASP Code Review Guide to help developers and application security specialists understand how to efficiently and effectively review a web application for security by reviewing the code.

2. Security and Penetration Testing:

OWASP produced the Testing Guide to help developers, testers, and application security specialists understand how to efficiently and effectively test the security of web applications. This enormous guide, which had dozens of contributors, provides wide coverage on many web application security testing topics. [2].

## REFERENCES

[1] —Web Server Security Guidelines,
Online: Available,
www.gswan.gov.in/PDF/CERT-In%20**Web**%20**Server**%20**Security**%20**Guideline**.pdf

[2] —The Ten Most Critical Web Application Security Risks, Online:available,
https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.

[3] —Improving Web Application Security: Threats and Countermeasures, online: available, http://msdn.microsoft.com/en-us/library/ff648653.aspx

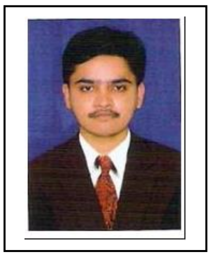
[4] SQLIA measures
Online: available,
www.iosrjournals.org/iosr-jce/papers/sicete-volume2/23.pdf.

[5] Type-I &Type-II XSS
Online : available,
https://mimmoo.wordpress.com/2011/06/19/xss-persistent-and-xss-non-persistent/

[6] Online:available ,
https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)

## BIOGRAPHIES

M.Vishnu Vardhana Rao working as
Asst .Professor in, Computer Science & Engg. Dept, in VIGNAN'S Institute of Technology & Aeronautical Engineering, Hyderabad , Telangana, India.



S.V Anil working as
Asst .Professor in, Computer Science & Engg. Dept, in VIGNAN'S Institute of Technology & Aeronautical Engineering, Hyderabad , Telangana, India.

SURAJ ALAMONI pursing Computer Science Engineering from VIGNAN'S Institute of Technology & Aeronautical Engineering,Hyderabad, Telangana , India.

Navya Reddy pursing Computer Science Engineering from VIGNAN'S Institute of Technology & Aeronautical Engineering, Hyderabad, Telangana , India