Evangelizing the need for Secure Web Development

Swapnil SINHA*

*Department of Systems (R&D), Syscom Corporation Ltd. (SAFRAN - Morpho), Noida, 201 301, India.

Abstract- As web attacks and security threats are increasing; security teams are now working together with web application development team to mitigate the causes of attacks at its initial stage of development. The paper describes the most critical web based threats that had defaced huge number of websites and also talks about various methodologies that can be followed by developers to build a secure web application. The paper describes certain practices that aids in tackling of web application security through secure web application development.

Index Terms-Cross Site Scripting Attack, Insecure codes, Malicious code, Secure Web Development, Security Breach, Web Attacks, Web Security Threats.

I. INTRODUCTION

When it comes to security, the developers had perception of considering it as a job not for them. Web applications get compromised due to insufficient security assessment at developer's end. This is because the team is chasing deadlines or had implemented a new feature or even due to insufficient skills [2]. The security team usually report similar set of issues to the development team. Many development teams aren't aware of what they do may hamper security and create loopholes. Increasing security attacks had forced the organizations to address application development to be secure application development.

It is important to understand that measures taken after the development process cannot be the solution to the web attacks. Attackers have understood that almost any web application can be exploited because of the mistakes made by developers when building them. Therefore it turns out to be developer's responsibility to focus on 'secure' web development. This article discusses about some of the biggest security threats that had shaken the organizations globally and guides developers about how they can possibly surpass these attacks at the early stages of development process.

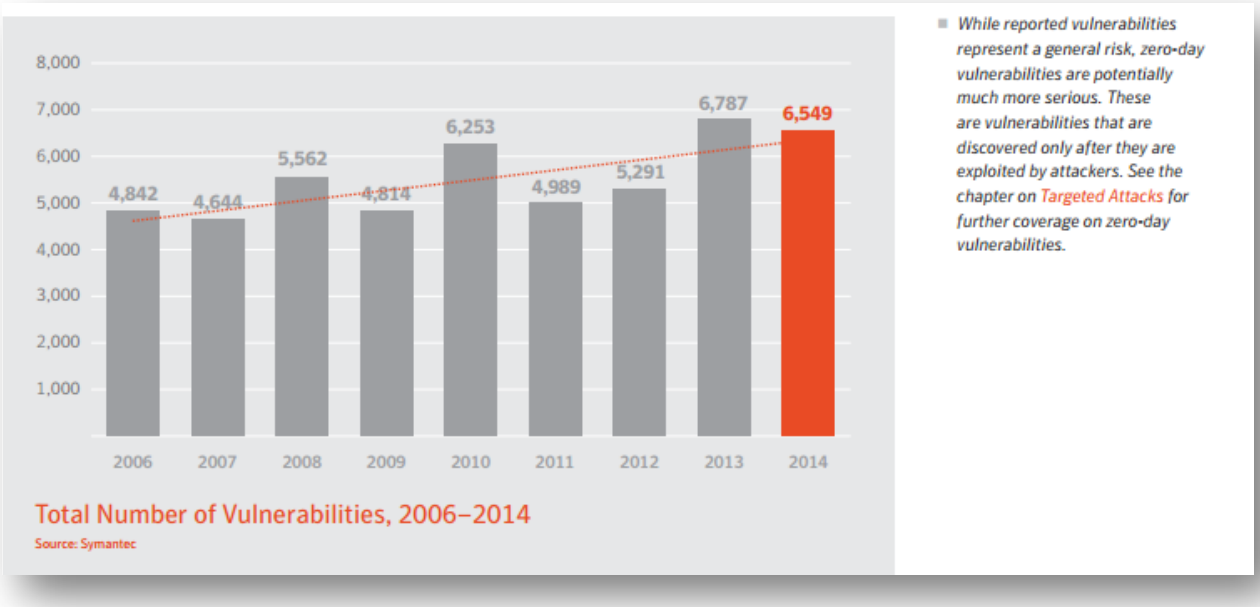

Figure 1: Symantec Internet Security Threat 2015 Report of total number of vulnerabilities from 2006-2014 [4]

WEB THREATS AND ITS HANDLING

Web application provides data to be accessed by the legitimate users. The attackers either steal or corrupt those data. The Open Web Application Security Project (OWASP) had reported some of the security threats to be the most powerful web application security threats used by the attackers. [1] These are categorized as follows-

1. Attacking through inputs

Attackers use this flaw to access backend data. Cross Site Scripting, Buffer Overflow and SQL injections are among the biggest threats that lie under this category. Attackers manipulate with the input and create malicious code to attack through end user's browser and perform illegal activities like hidden field data hack, session hack, database hack, query string hack etc. Certain measures can be taken by the developer while building the application to avoid these attacks from occurrence. Firstly, a rigorous input data validation should be done. Client side validation could be bypassed by the attacker so, server side validation must be included too. If possible a strict data format check should be included for inputs. Sometimes this cannot be done where inputs cannot be summarized for example in search section of web. Here a negative input validation can be done where data can be filtered from evil inputs that may do harm to the system thereby preventing insertion of scripts from producing executable form at client's machine.

Validation criteria for developers [3]

i. Allowed character checks
ii. Data type checks
iii. Regex checks
iv. Length check (Minimum and Maximum)
v. Range check (Minimum and Maximum)
vi. Null allowed check
vii. Data Annotation check
viii. Converting <, >, (,), # and & characters to HTML encoding to avoid running client side scripts.

2. Attacking sessions for authentication and unauthorized access

Sessions have always been the prime targets for hackers. Hacking the sessions, accessing the credentials of legitimate users to get authenticated and traverse unauthorized access to the web application by session hi-jacking, the attacker can even modify or delete the content from the web. The browser cannot identify whether the malicious script is coming from a trusted source or not. Therefore, developer should take high precautions by encrypting the session elements. XSS attack is also used for session hacking to impersonate as an authenticated user. The developer can follow the principle of least privilege so that even if the impersonator gets authenticated, not much of the system gets effected.

TOP 10 countries by number of attacked users

	Country	% of attacked users*
1	Russia	45.7%
2	India	6.8%
3	Kazakhstan	4.1%
4	Germany	4.0%
5	Ukraine	3.0%
6	Vietnam	2.7%
7	Iran	2.3%
8	UK	2.2%
9	Malaysia	1.8%
10	Brazil	1.6%

** Percentage of attacked users in the country from total number of attacked users*

Figure 2: Kaspersky Security Bulletin report 2014 representing top 10 most cyber attacked countries [5]

3. Tracing via exceptions and unsafe access control

In 2014, the researchers mentioned that 99.3% of malicious files re-used known URLs for Command & Control (CnC) [6]. Usually in web application, an authenticated user is not restricted on what not to do with the data (internal/external). Attackers use this as a privilege to access sensitive data with probably read/write and download permission. Moreover if there is an exception thrown by the application, the error traces the log which the attacker can use to gain system understanding like stack trace, database dumps, error codes and may cause security mechanism to fail or worse. These log messages gives implementation details which should never be exposed to attackers. Attackers can replicate these errors and may loop it on server thereby consuming a huge amount of resources end up in DoS attack.

The developer should disable path traversal of end users especially in case of external data source. The file permissions must be limited so that it only serves its purpose. The attacker may look into the source code for any external link available on the web page. In this case either block the source code display or put the external calls in API libraries. For database access, provide only those permission that are needed for performing the operation. Stored procedure is recommended instead of SQL commands for including validation at backend level too. The errors on the page should be handled in a way such that end users get a meaningful message and site maintainers receive the log information about the error. Thus the attacker cannot fetch any purposeful information from the errors. By default all the permissions should be denied and should be available on needful basis.

4. Configuration Security flaw

In 2014, HeartBleed made the headlines as a vulnerability found in the Open SSL library. In October 2014, Google discovered another vulnerability known as Poodle that let the attackers exploit servers compatible with older version of SSL i.e. SSL 3.0 [4]. Many web applications uses OS features like system date and time, browser cookie and cache management, memory management etc. Attacker may access these resources and consume it heavily thereby leading to DoS attack. The developers cannot assure these flaws while developing the software but can use certain tools to monitor it. Load testing tools such as JMeter creates virtual scenarios to monitor the site's performance on heavy traffic. It's always recommended to limit the resource allocation to any user by establishing quotas. For unauthenticated users, try avoiding unnecessary database request or other external resources by caching it.

CONCLUSION

The paper describes various web security threats and how these threats are hampering through the web vulnerabilities. It also tells about certain approaches for developers which if practiced can mitigate the exposure of security flaws to a minimal level. Securing HTTP request has always been a tedious task. While security team works on protecting the web application against various types of attacks, development team can also play a vital role in parallel. Practicing these methods and following security principles will end up with a greater user experience with web.

ACKNOWLEDGMENT

This work is supported by Syscom Corporation Ltd.

REFERENCES

[1] OWASP Top 10 - 2013 The ten most critical web application security risks.
[2] Sophos Trends and Prediction 2015 pp 7.
[3] https://en.wikipedia.org/wiki/Data_validation
[4] Symantec Internet Security Threat Report *April 2015 Volume 20*
[5] Kaspersky Security Bulletin 2014
[6] Websense 2015 Threat Report

AUTHOR

Swapnil SINHA – Former M.Tech (I.T.) student; C|EH v6; Software Engineer at Syscom Corporation Ltd., swapnil.sinha007@gmail.com