International Journal of Engineering and Techniques - Volume 3 Issue 6, 2017

RESEARCH ARTICLE OPEN ACCESS

A Study on Different Scanners and Their Limitations for Web Application Vulnerabilities

Mrs. M. Sridevi[1], Dr. K.V.N.Sunitha[2]

1(Research Scholar, Asst. Prof. Laqshya Institute of Technology and Sciences, Khammam
Email: sreetech99@gmail.com)
2 (Principal & Prof, BVRIT College of Engineering for Women, Hyderabad
Email: k.v.n.sunitha@gmail.com)

Abstract:

Web applications have become commodiously popular in the last two decades and have touched every walk of our daily lives. At the same time they have accrued large volumes of sensitive data. This has allured the attackers, who ever since been looking out for the vulnerabilities in the applications constantly. In this paper, the most common web attacks are deliberated. The artifice of how these attacks are effectuated is discussed. Some of the most common industries targeted by the attackers are Finance, Government, Healthcare and IT. The recent trends in the web attacks and their consequences on the organizations are analyzed. The common scanners or automated tools to scan for vulnerabilities and their limitations are deciphered.

Key words: Web Application Security, Web Application Vulnerabilities, Denial of Service, SQL Injection, Cross Site Scripting, Cross Site Request Forgery, Web Security Scanners

I. INTRODUCTION

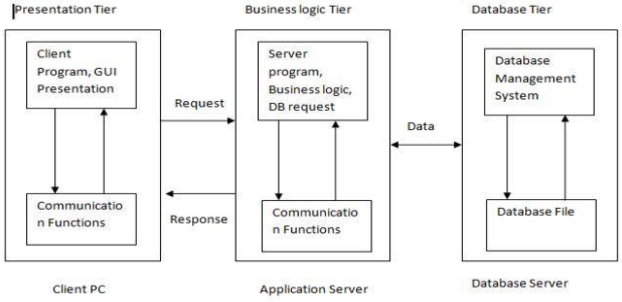
Over the last two decades, the use of internet has been growing exponentially which led to the witnessing of stupendous growth of Web applications. Web applications are increasingly used in almost, all the industries e.g., Financial, Healthcare, Government. Apart from these organizations, even individual people are using the web applications in their everyday life accessing and sharing data over the internet. As a result of this, a lot of data is accumulated in the databases of the Web applications which is attracting a lot of attackers. Nowadays, the information is becoming more and more valuable and the attackers understand this very well [1]. For that reason, they are constantly looking for vulnerabilities in web applications.

Background

Conventionally, the network level attacks are handled efficaciously by using firewalls but the security mechanism at the application level is grievously deficient [2]. Even though the firewall is used, it cannot stop all the requests because it has to keep the application open for business transactions. Most of the threats intrude camouflaged as business transaction requests, so the responsibility of the security from these kind of threats comes down to web application. Most of the web applications interact with databases frequently, any vulnerability left open to the threats, leads to monetary losses, information leakage, server disruption and could push organizations into ethical and legal issues [3].

Web Application Architecture:

A web application is a program stored on a remote server and is accessed through a browser interface over the internet and lets the user read or write data to the database. A webpage consists of HTML tags, CSS and JavaScript and these components are executed in the browser on the client side. The client makes HTTP requests to a web server using a Uniform Resource Locator (URL) over the internet.Here, HTTP is a transport protocol which defines data format and algorithms for making communication between client and server possible. On the server side, the HTTP request is processed according to the business logic defined in the server-side language(PHP, Java,etc.,), this process often involves writing or fetching data from the databases [4]. Finally, the web server responds to the client with an HTTP Response.



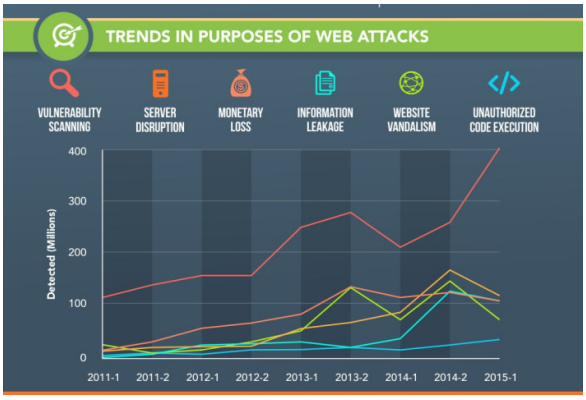
A 3-tier architecture contains presentation tier or client, business logic tier or web server and

database tier or database server. An isolated business logic layer would enhance the security as it adds an extra level between the client and database [5].

Web Attacks Statistics

According to the recent report by Positive Technologies [6], 1 out of 2 attacks aimed at accessing data and 1 out of 3 attacks aimed at users. About 61% of attacks have occurred during the daytime and evenings.The website attacks are mostly afflicted to Government, IT, Finance and Education sectors. SQL Injection and Cross-Site Scripting were the most common attacks, each representing almost one-third of the total number of attacks. The report also reveals that when government website is attacked, the main objective is to obtain access to important data. Since most of the data from government websites are personal data of application users, the attacks are directed towards database containing the information.

The attacks aimed at financial services websites are normally to steal money and they attempt to gain either access to sensitive information or to gain control over the server. The attacks on IT Industry websites are mostly dominated by SQL Injection and Cross-Site Scripting either to deface the websites or to infect the workstations with malware resulting in risking the company's reputation. Attackers against educational institutions more often are students trying to access data related to exams or modifying the data like exam results, grades or scholarship lists. In this case, the most common attack is Cross-Site Request Forgery which accounts for almost more than 90% of the attacks.



Denial of Service(DoS):

A Denial of Service attack would prevent any individual user from using a resource from the website. In this type of attack, the attacker would send millions or even billions of requests to keep a service or a resource of a website busy so that service would fail eventually. This attack makes the services inaccessible for the authorized users. There are two forms of Denial of Service attacks are Ping of Death and SYN Flood [7].

Distributed Denial of Service(DDoS):

In DDoS attack, the malignant actor uses other different computers to attack so that his identity will become difficult to trace. As more computing resources are needed to create a successful attack, this type would suit the attackers. DDoS attack is fundamentally a DoS attack on the target website using multitude of hosts. Coordination among the attacking hosts is the fundamental characteristic to the DDoS attack. Main characteristics of a DDoS attack [8].

1. It is a subset of DoS attack
2. More than one attacking hosts involved
3. Coordination among the attacking hosts

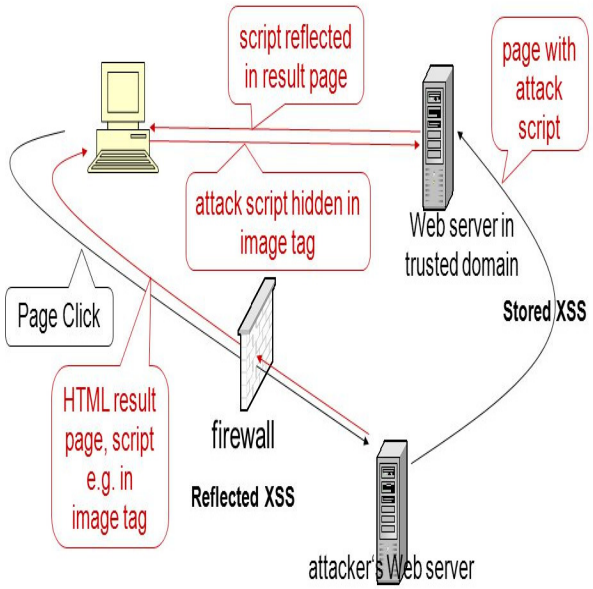
SQL Injection:

An SQL Injection is a code injection technique where the attacker injects malignant code into the strings retrieving imperative information from the web server database. The attackers using SQL related keywords in their code which results in manipulation of users data, authentication bypass, denial of access which can further lead to destruction of database.

Some of the proposed solutions includes such as avoid connecting to database with superuser access, avoid using dynamic SQL queries, use encryption techniques or hash format for sensitive data. It is also advised not to reveal more information in error messages instead use custom error messages for displaying minimal information.

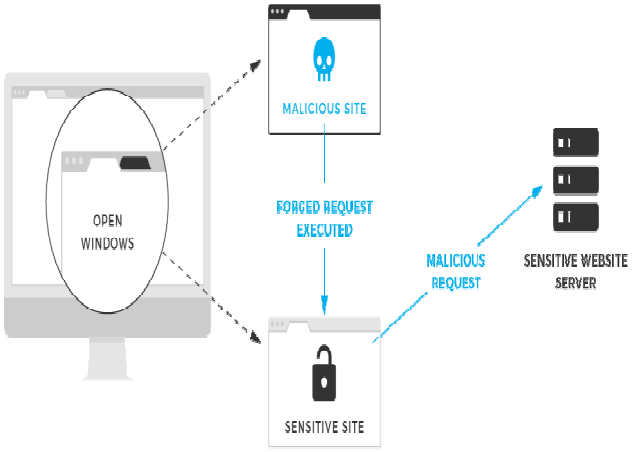
Cross-Site Scripting(XSS):

Cross-Site Scripting (XSS) is a type of attacks in which the attacker injects malicious scripts into web applications. An attacker use XSS to send malicious script to an unsuspecting user and the end users browsers has no way to know that the script cannot be trusted and will execute the script. Unlike most of the other attacks, which involve two parties, the attacker and the target website, XSS involves three parties, attacker, client and target website. The attacker steals the cookies from the client and impersonates the client while attacking the target website. This is achieved by malicious JavaScript code running from the web browser of the client and with the access privileges of the client [9].



Cross-Site Request Forgery:

Cross-Site Request Forgery is one of the most exploited security vulnerability likewise Cross-Site Scripting and SQL Injection [10]. CSRF differs from CSS whereas in CSS the attacker inputs malicious JavaScript onto the website (eg. comments area) while CSRF attack doesn't need it. As opposed to Cross-Site Scripting and SQL injection CSRF has received little attention. Comparatively CSRF is easy to exploit and easy to diagnose and easy to prevent. This threat is vastly underestimated by the developer community and some of the developers are under the wrong illusion that fortification against the Cross-Site Scripting also safeguards against CSRF attacks.



In CSRF attack, the user need to be logged in to the target site while visiting the attacker's site. If the target site accepts GET requests, the CSRF attack is possible without using javascript and if it only accepts POST requests, the attack needs javascript [11].

We can protect the individual websites by making taking the below safety measures.

1. By not letting the GET requests modifying any data.
2. By requiring a cryptographically generated random value for all POST requests.

Scanners

Web applications have always played an important role in an organization being a gateway to organization's imperative information. Nowadays, hackers are always active and look for attacking data which led to developing security testing applications or web security scanners. A web application security scanner is a software program which performs automatic black box testing on the web application and identifies security vulnerabilities. These scanners do not access source code instead they go for functional testing and find the vulnerabilities in web application. These are widely used by security auditors and web application administrators because of their easy usability by performing tests and identifying

vulnerabilities. Altogether, there are more than 30 scanner tools according to OWASP, some of them being commercial and some are open source tools. Some of the top web security scanners include Burp Suite, Netsparker, Arachni, W3af [12] and vega. Grabber, vega, wapiti, W3af and skipfish are among the open source web security scanners which are widely used in the market.

Vulnerability scanner	XSS Accuracy / False positive	SQL injection Accuracy / False positive	DDOS Accuracy / False positive
W3AF	35/30	37/0	19/3.5
IronWASP	100/1.8	96/0	31/0
ZAP	100/0	100/30	18/100
Syhunt Dynamic	100/0	100/50	95/0
QualysGuard WAS	50/0	63.24/0	69.5/0
Wapiti	66.6/42.8	100/20	100/0
Vega	100/0	100/20	100/0
Scrawlr	X	13/0	X
SQID(SQL Injection Digger)	X	0/0	X
Crawlfish	14/28	X	X
XSSS	33/71	X	X
ScreamingCSS	61/0	X	X

Table 1: vulnerability scanner accuracy and false positive comparison.

In table 1 displays each scanner accuracy and false positive against XSS attack, SQL injection, DDOS attack. Above table conclude that accuracy and false positive of each scanner and some of scanners doesn't prevent some kind of attacks.

LIMITATIONS:

Most of the scanners detect vulnerabilities like SQL Injection, XSS, File inclusion, Path traversal, hidden pages & files and web server vulnerabilities. But sometimes, the web security scanner won't work in some cases. These are the few reasons why web security scanner fails to detect vulnerabilities.

[1] Custom-built authentication mechanism:

When a web application uses proprietary approaches to authenticate the users, sometimes the scanner may fail to login and only scan unauthenticated parts of web application.

[2] Web applications with heavy use of AJAX:

Most of the scanners can't handle dynamic ajax content properly

[3] Websites with lot of content or huge number of dynamic generated pages:

E-commerce websites like Amazon and social networking site Facebook generates huge number of dynamic web pages because of user actions makes the scanner unable to perform tests properly.

Most of the time, the scanners are not able to test for authentication vulnerabilities, Session management flaws, Vulnerabilities in access controls, application logic flaws, shared hosting vulnerabilities and leakage of sensitive information [13].

CONCLUSION

The use of the web applications has become conspicuously popular in recent years. This has led to the accumulation of valuable and sensitive data. This phenomena has led to attackers constantly looking for vulnerabilities, has opened up new battle front for the developers.

This paper surveys the most common threats to the web applications. However much the defense is updated, the new attacks are always surfacing. This paper equips the web developers with better planning of web security and encourages future work on threats.

FUTURE ENHANCEMENT

After understanding limitations of scanner, there is no efficient scanner to defense web application

vulnerabilities, and although each scanner might limited to few attacks so in order to overcome scanners limitation and to prevent web application vulnerability an efficient mechanism must be introduced, overcome scanner limitations is future enhancement in future proposing Hybrid framework , which will integrate existed scanner functionality and overcoming above mentioned limitations.

REFERENCES

[1] Lepofsky, Ron. "Web Application Vulnerabilities and Countermeasures." The Manager's Guide to Web Application Security: 2014, pp. 47–79., doi:10.1007/978-1-4842-0148-0_4.

[2] J. Pescatore, Web Services: Application-Level Firewalls Required,
report no. SPA-15-5542, Gartner, Stamford, Conn, 7 Mar. 2002.

[3] Gopal R. Chaudhari, Prof. Madhav V. Vaidya, A Survey on Security and Vulnerabilities of Web
Application, International Journal of Computer Science and Information Technology, Vol. 5 (2), 2014.

[4] Dwivedi, Vandana, et al. "Web Application Vulnerabilities: A Survey." International Journal of Computer Applications, vol. 108, no. 1, 2014, pp. 25–31., doi:10.5120/18877-0144.

[5] Simon Brown, Architecture, Coding the. "When Do You Need a 3-Tier Architecture?" Coding the Architecture, www.codingthearchitecture.com/2012/07/20/when_do_you_need_a_3_tier_architecture.html.

[6] Positive Technologies, Web Application Attack Statistics Q1 2017,
https://www.ptsecurity.com/upload/corporate/ww-en/analytics/WebApp-Attacks-2017-eng.pdf.

[7] Khaled M. Elleithy, Drazen Blagovic, Wang Cheng, and Paul Sideleau, Denial of Service Attack Techniques: Analysis, Implementation and Comparison, SYSTEMICS, CYBERNETICS AND INFORMATICS VOLUME 3 - NUMBER 1, 2005.

[8] Tuomo Penttinen, Distributed Denial-of-Service Attacks in the Internet, Department of Computer Science and Information Systems, University of Jyväskylä, 2005.

[9] AmitKein, Cross Site Scripting Explained, Sanctum Security Group, 2002.

[10] Barth, Adam, et al. "Robust Defenses for Cross-Site Request Forgery." Proceedings of the 15th ACM Conference on Computer and Communications Security - CCS '08, 2008, doi:10.1145/1455770.1455782.

[11] William Zeller and Edward W. Felten, "Cross-Site Request Forgeries: Exploitation and Prevention", Department of Computer Science, Woodrow Wilson School of Public and International Affairs, Princeton University, 2008.

[12] Mcquade, Kinnaird. (2014). Open Source Web Vulnerability Scanners: The Cost Effective Choice?. . 10.13140/2.1.3360.0005.

[13] "Scalar IT Solutions." Scalar IT Solutions Limitations of Automated Web Application Vulnerability Scanners Comments, www.scalar.ca/en/blog/limitations-of-automated-web-application-vulnerability-scanners/.