ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 2, February 2017

A Survey on Web Application: Vulnerabilities, Attack and Detection Techniques

[1] Saba Khan [2] Prof. Dilip Motwani
[1] Student of Master of Engineering [2] Associate Professor
Department of Computer Engineering, Vidyalankar Institute of Technology,
Mumbai University, India

Abstract: **-- A web application is generally known as a client-server software application where the client uses a user interface within a web browser. Web applications today provide a universal way to access information over the internet. Since the use of web application are increasing it became a popular target for security attacks. Security vulnerabilities in web applications may result in stealing of confidential data, breaking of data integrity or affecting web application availability. Thus, the task of securing web applications is not only important but also needs immediate attention. Some of well-known web application vulnerabilities are SQL Injection, Buffer Overflow, Cross Site Scripting etc. In order to overcome these vulnerabilities, it is important to detect first the problem before preventing it. In this paper, different types of web application attack which are vulnerable to the web application have been discussed. Furthermore, several techniques for detection of web application related attacks are presented.**

***Keywords*— Cross-site scripting, SQL Injection, Cross-site Request Forgery, Remote File Inclusion, Intrusion Detection System.**

I. INTRODUCTION

Web application run in a remote web server and uses HTTP protocol [1] for responding to the request over the internet. Web applications are computer program which allow website visitors to submit and retrieve data to database using the internet over their preferred web browser. There are many web browsers like Internet Explorer, Apples's safari, Firefox browser etc [2]. Web based applications are used by the user to perform various operation such as to search data, exchange message, interact with other, to perform financial operation etc [3]. Thousands of attacks are made daily by some anti-social elements to make our experience with web application disastrous. These attacks are put in place to access the private information of the user and to malign the name of the particular web application. Due to the presence of security weaknesses in web application it can steal private information and can perform malicious operations which will limit the use of applications. For example, banking services handle sensitive and private data which if becomes accessible to attacker can lead to havoc [3]. If an attacker is able to execute an attack on an application, there will be possibilities that the user data or application source code will also be compromised. Many measures have been taken to decrease the web application attack. In this paper, we have presented various web application attack and their detection technique.

In section 2, different types of web application attack have been described. In section 3 web application vulnerability detection is presented. In Section 4 we have described types of IDS technique and previous work done in developing intrusion detection system. In section 5 conclusion of the paper is presented. Section 6 consist of references that helps in the completion of the paper.

II. TYPES OF WEB ATTACKS

A. Cross-Site Scripting (XSS)

Cross-site scripting is the most common security problem affecting web applications. This type of attack allows malicious script to be injected into trusted web sites causing harmful operation affecting user or site [5]. This occur when web application fails to validate input from the web site users. It allows a malicious user to steal sensitive information and performs other harmful operations [8] such as transfer private information, manipulate the web content, etc [6]. Cross-Site Scripting are of three types: Stored, Reflected, and Document Object Model (DOM)-based.

1. The stored XSS

This type of attack occurs when the input from the user is stored in the database and then it is used in the response page. This type of vulnerability stores the malicious script. It is stored in databases, message forums etc. of the attacked server [5]. The user which visits and execute those scripts will pass their privileges

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 2, February 2017

to the attacker. The stored XSS is more harmful than other XSS

2. The reflected XSS

This type of attack occurs when user input is not properly validated and referenced in an immediate response web page. This occurs mostly in greeting or error messages. Reflected XSS is executed by the victim's browser and occurs when the victim provides input to the web site [5]. In this attack, the attacker creates their URL which includes malicious code and makes the victim to believe that the URL is trusted. The malicious links will be delivered to victims through the link which is embedded into the web page located on some different server [6].

3. The Dom-based XSS

This type of attack occurs on the client side. DOM allows dynamic scripts to reference the document's components – for example, a session cookie. This happened when invalidated user input is dynamically obtained from the document object model structure [6].

B. Cross-Site Request Forgery (CSRF) This type of attack forces end user to execute unwanted actions on a web application on which legitimate user are currently authenticated [9]. This can take place through email. e.g attacker. can force the legitimate users of a web application to execute actions of his choice. In case of normal user CSRF exploits can compromise end user and operation but if the targeted user is the administrator account it can compromise the entire web application. This type of attack can trick the legitimate user into submitting a malicious request. It can take over the privileges and identity of the victim and can perform undesired function on behalf of victim [8]. Due to this, if the user is currently authenticated to the site, the site will not be able to distinguish between the legitimate request and the forged request sent on behalf of the victim. CSRF attacks target state-changing request such as changing the victim's password or email address etc. It can force the victim to retrieve the data that doesn't benefit an attacker because the response will be received by the victim not the attacker.

C. Buffer Overflow

Buffer overflow attack occur when the excessive data is written to a memory block than it is capable of holding [3]. In this attack, while writing data in the buffer, if the buffer size is limited, it overruns the boundaries of the buffer and it will overwrites the memory in the adjacent location. Buffer overflow attack can lead's to DoS attack. For example, crashing of the application [2]. Techniques for exploiting Buffer overflow vulnerability[10]:

1. Stack Buffer Overflow

Attacker exploits stack overflow to manipulate the program for their use.This type of buffer overflow occur when a program writes extra data on stack's buffer more than the capacity of stack buffer.

2. Heap Buffer Overflow

An application allocates memory to the heap dynamically at run time .when a buffer overflow occur inside the heap data area then heap buffer overflow is caused.

D. Brute Force

This attack is an automated method that determines an unknown value by using a large set of possible values. If the attacker arrives at the correct value, private and sensitive information can be accessed[13].The web applications which are password based are more vulnerable to this attacks. This attack mainly works for short passwords. There are various types of this attack which includes the following[12].

1. Blind Attack: In this type, the attacker tries to find out account member name as well as password.
2. Targeted Attack: In targeted attack, the attacker tries to predict the password.
3. Trawling Attack: In Trawling attack, the attacker tries to find out the account holder for a given password.

E. Remote File Inclusion (RFI) RFI process includes remote files by exploiting vulnerable inclusion procedures which is implemented in the application. The web application may trick to include remote files with malicious code at a time of taking user input. For example: parameter value, URL, etc. An attacker use RFI to:

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 2, February 2017

1. Run malicious code on the server:
The malicious file that contain any code will be run by the server [14]. Code which is in includes file will be executed in the context of server user, if that file is not executed using wrapper. Due to this complete system can compromise.

2. Run malicious code on clients:
The malicious code by the attacker can lead to manipulation of the content of response sent to the client. The malicious code can be embedded by the attacker in the response that will be run by the client. Due to extensive use of "file includes" and default server configuration, PHP is more vulnerable to RFI attacks.

F.SQL Injection (SQLI)

SQL Injection is an attack where attackers can inject SQL queries through input of a web page. Such SQL commands can make changes to the database and modify the contents. They can retrieve important information which harms the integrity of the database. The target is to bypass the server level and gain access to the backend. SQL injection occurs due to insufficient input validation, Such as if the data provided by the user is not properly validated and directly included in an SQL query. Because of these vulnerabilities, the attacker can submit SQL commands directly to the database[15].

G. Classification Of SQL Injection
SQL Injection can be classified broadly as [16]:
1. Tautologies:

This type of attack make use of conditional queries and inserts SQL tokens into them, which proves to be always true. This attack used to identify injectable parameters to extract data.

2. Illegal/Logically Incorrect Queries:

The database error messages can be used by attacker to find vulnerabilities in the applications. This attack can be used to extract information and data about the backend database.

3. Union Query:
In this type of attack, the attacker uses infected queries to inject in the existing legitimate queries by using UNION operator that return the dataset result of both the legitimate and infected queries and thus retrieve information from the database.

4. Piggy-backed Queries:
In this type of attack, the attackers attach delimiters such as ";" to the original query which can cause multiple queries to be executed in which the first query is legitimate followed by infected and illegitimate queries. Generally, the initial query is trusted one, whereas following queries could be malicious. The attacker can modify or extract data, execute remote command, etc.

5. Stored Procedure:
This type of attack execute the stored procedure which is present in the database and is typically used to validate data and access control mechanism. In stored procedure, the malicious user find the database which is in use and the vulnerable parameters. The attack is then executed which includes the stored procedure from the database as well as the procedures that communicate with the operating system.

6. Blind SQL:
In this type of attack, the attacker inserts queries in the form of question such as ''yes or no''. Based on the application response it determines the answer. This technique enables the malicious user to check for vulnerabilities even when there is no result to returned from the database to the end user. It can identify injectable parameter and can extract data.

7. Time-Based Attacks:
In this type of attack, the attacker used timing delays in the database response to observe the amount of time taken to execute query. It extract information by manipulating the response time of the database.

8. Alternate Encodings:
This technique is used to bypass various detection method. It enables the attacker to encode the input string using ASCII, hexadecimal or Unicode character to hide the attack, due to this it will pass undetected and execute. It can cause denial of service attack.

III. WEB APPLICATION VULNERABILITIES DETECTION

There are different methods and techniques which are used to deal with threats against web application. Some of them are firewalls, DMZ, IDS, etc. Intrusion detection system is used to find actions and attempts which are trying to bypass the security policy, and alerts the

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 2, February 2017

administrator in case of threat detection [11]. Intrusion Detection Systems (IDSs) are one of the most useful tools for identifying malicious attempts over the network and protecting the systems without modifying the end-user software.

The efficiency of IDS can be evaluated using following metrics: Table 1 shows the IDS metrics and their definition

IDS Metrics	Metrics Definition
True Positive(TP)	IDS identifies an activity as an attack and the activity is actually an attack
False Positive(FP)	IDS identifies an activity as an attack but the activity is acceptable behaviour
True Negative(TN)	IDS identifies an activity as acceptable behavior and the activity is actually acceptable
False Negative(FN)	IDS identifies an activity as acceptable when the activity is actually an attack

Detection rate = (TP) / (TP + TN)
False alarm rate = (FP) / (FP + TN)

IV. WEB APPLICATION INTRUSION DETECTION SYSTEM

Intrusion detection systems are classified as network intrusion detection systems (NIDS) and host intrusion detection systems (HIDS) based on whether an intrusion detection system monitors a network, or a host. Currently, there are two common types of IDS: signature-based intrusion detection systems and anomaly-based intrusion detection systems.

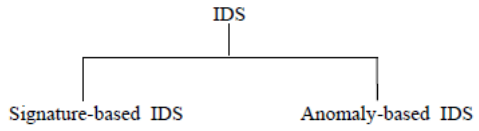

Fig.1 Types of IDS

1. Signature-based Intrusion Detection Systems

Signature-based IDS uses the signatures of already known attacks or vulnerabilities to detect security threats. Such detection mechanisms are applied at the both the network level, by analyzing network traffic and at application level, by monitoring server logs [11]. As long as signatures are specified ahead of time this IDS would ideally identify 100% of the attacks with no false alarms.

2. Anomaly-based Intrusion Detection Systems

Anomaly-based IDS focuses on normal behaviors of the system's instead of focusing on attack behaviors. It detect intrusions by analyzing deviation from expected behavior in the captured data. The data is said to be anomalous if the deviation crosses a certain threshold. It has the capability to detect unknown intrusions, but the major difficulty with anomaly based approach is to understand what constitutes normal behavior and abnormal behavior [11].This approach can lead to several false negative alarms.

There are two types of attack detection technique which are classified into static and dynamic detection. Static detection techniques includes the post occurrence of events whereas dynamic detection includes detecting attack in real time.

The Table 2 shows the related work done in the field of detecting web-application attack. The table shows the name of the author, title of the work its advantage ,limitation and the attack class

Table 2: Analysis Of Various Intrusion Detection Techniques

Serial No.	Author(s)	Title	Advantage	Limitations	Attack class
1	Vigna et al.[18]	Testing network-based intrusion detection signatures using mutant exploits	Examined Apache logs to collect data on string length and sequence and exploited mutations to detect Number of signature based attacks either by SHORT or IIS RealSecure before and after mutating data	Did not consider FP/FN results because the goal was to provide useful indication about the average quality of signatures under testing	Directory traversals, Buffer Overflows, DoS, Stack Overflow, Double decoding.
2	C.Kruegel and G.Vigna [19]	Anomaly Detection of Web based attacks	It allow the system to detect new type of attacks and cannot be evaded by attempting to hide malicious code inside a string	The header data of GET request are not taken into account its rely on web access logs, attacks that compromise the security of the web server	Buffer overflow, Directory traversal, cross site scripting, Input validation & Code Red
3	M. Auxilia and D.Tamilselvan[20]	Anomaly detection using Negative security model in web Application	Provided a rule set for detecting the attacks, HTTP traffic can be monitored in real time in order to detect and prevent attacks from reaching web applications	Does not provide a unique rule for detection of all attacks	HTTP attacks, SQL injection, Cross site scripting
4	G. Vigna et al.[21]	Reducing errors in the anomaly based detection of web based attacks through the combined analysis of web requests and SQL queries.	When the attack is detected it does not block anomalous request immediately but its attempt to serve them through a web server with restricted access to sensitive information thereby reducing the false positive.	Vulnerable to distributed DOS attack	SQL injection, Command injections, Information tampering, Cross site scripting.
5	Le et al. [22]	Double Guard : Detecting Intrusions in multi-tier web application	It examines both web server and database server logs to precisely detect attacks leaking confidential information. The casual mapping can identify the attack even in normal network traffic, 100 % detection accuracy with 0 % false positive rate for static web pages, and 0.6% false positive rate for dynamic web pages.	Cross site scripting attack is possible, Vulnerable to mimicry attack, Not designed to mitigate DDOS attacks.	Privilege escalation, Hijack future session attack, Injection attack, Direct DB attack
6	Juan Jose Garcia Adeva and Juan Manuel Pikatza Atxa[23]	Intrusion detection in web application using text mining	Log information generated by the system does not need any particular format, Does not require any explicit programming for machine learning.	The system cannot detect the new attack, Focus only on access control, The false positive is high.	Access control.
7	C. Kruegal et al[24]	A multi model approach to the detection of web based attacks	The reduced number of false positives, able to detect a high percentage of attacks with a very limited number of false positives.	Relies on web access logs, The direct instrumentation of web servers introduces unwanted delay	Buffer overflow, Directory traversal, cross site scripting, Code Red, Input validation.
8	Chou[25]	Security Threats on Cloud Computing Vulnerabilities	Increased frequency of identified SQL Injection attacks in SaaS, PaaS, and IaaS Cloud settings measured in %	Not discussed on false positive and false negative as focus was more on cloud security	SQL injection, DDos, brute force, session hijacking, XSS,etc.
9	R.Sekar [26]	An Efficient Black box technique for defeating web application Attacks	Effective in detecting a broad range of attacks on application written in multiple languages, Low overheads	A false negative may occur due to parsing error, policy error or incompleteness.	SQL injection, Command injection, Path traversals, cross site scripting.
10	Ludinard et al[27]	Detecting Attacks against Data in Web Applications	This technique rely on the application normal behaviour using invariants. The violation of an invariant is then considered as an indication of an attack against the application.	Attacks such as Cross Site Scripting or Session Hijacking are not detected by this approach.	SQL injection

V. CONCLUSION

There are many challenges for web application services which need to be addressed such as security ,scalability and availability. In this paper we have discussed different types of web application attack and their detection techniques. By using different techniques the attacker can steal sensitive data and may harm the web application. It is not possible to built a secure web application until we know about possible attack and threats. In this paper, we came across through the XSS attack which is the major problems in web application. It can lead to stealing of user account and cookies and can also transfer private data if the input is not validated. We have also discussed, CSRF attack which forces end user to execute unwanted actions on web application. SQL injection followed by its classification, Buffer overflow, Brute Force and RFI attack is also presented in this paper. As there is an increase in new types of web attacks there is a need for security professional to react quickly without putting web application services at risk. There is a need to get aware of different types of attack which attacker use to threaten web application.

In future, the IT professional and developer should work to increase the necessary security requirement so that breaches can be reduced and attacks can be avoided.

REFERENCES

[1] M. khari, P. sangwan and vaishali, "Web-application attacks: A survey," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi,2016,pp.2187-2191

[2] H. Atashzar, A. Torkaman, M. Bahrololum and M. H. Tadayon, "A survey on web application vulnerabilities and countermeasures,"2011 6th International Conference on Computer Sciences and Convergence Information Technology(ICCIT), Seogwipo, 2011, pp.647-652

[3] The Open Web Application Security Project. Top Ten 2013

[4] M. K. Gupta, M. C. Govil and G.Singh, "Predicting Cross-Site Scripting (XSS) Security vulnerabilities in web applications,"2015 12th International Joint Conferences on Computer Science and Software Engineering (JCSSE),Songkhla,2015,pp.162-167.doi:10.1109/JCSSE.2015.7219789

[5] Abdalla Wasef Marashdih, Zarul Fitri Zaaba, "Cross Site Scripting: Detection Approaches in Web Application,"(IJACSA) International Journal of Advanced Computer Science and Applications,Vol.7, No. 10,2016

[6] Gupta.M, Govil.M, Singh.G and Sharma.P 2015. XSSDM: Towards detection and mitigation of cross-site scripting vulnerabilities in web applications. In the 2015 International Conferences on Advances in Computing, Communications and Informatics(ICACCI). Doi: 10.1109/ICACCI.2015.7275912

[7] OWASP-XSS,https://www.owasp.org/index.php/Crossite_Scripting_(XSS)

[8] Siddiqui,M and Verma, D. 2011. "Cross site request forgery:A common web application weakness,"Proceedings of the 3rd IEEE International Conference on Communication Software and Networks (ICCSN),Xi'an, China.pp.538-543.Doi:10.1109/ICCSN.2011.6014783

[9] OWASP-CSRF,https://www.owasp.org/index.php/Cross Site_Request_Forgery_(CSRF)

[10] OWASP-BUFFER OVERFLOW ATTACK, Accessed from https://www.owasp.org/index.php/Buffer overflow attack

[11] N.Sakthipriya and K.Palanivel "Intrusion Detection for Web Application:An Analysis" International Journal of Scientific & Engineering Research, Volume 4. Issue 5, May-2013 1824 ISSN 2229-5518

[12] Jaspreet Kaur, Rupinder singh and Pawandeep kaur "Prevention of DDoS and Brute Force Attacks on Web Log Files using Combination of Genetic Algorithm and Feed forward Back propagation Neural Network," Intenational pp Journal of Computer Application (0975-8887)Volume 120-No.23,June 2015

[13] 19 Adams, C, Jourdan, G.V., Levac, J. P., & Prevost, F. (2010,August), "Lightweight protection against brute force login attacks on Web applications", In PST (pp.181-188).

[14] Robledo, H. 2008. Types of Hosts on a Remote File Inclusion (RFI) Botnet. Electronics, Robotics and Automotive Mechanics Conference, (CERMA '08), Morelos, Mexico, pp.105-109. Doi: 10.1109/CERMA.2008.60

[15] Buja, G., Jalil, K., Ali, F. and Rahman, T. 2014. Detection model for SQL injection attack:An approach for preventing a web application from the SQL injection attack. Proceedings of the IEEE Symposium of Computer Applications and Industrial Electronics (ISCAIE),pp.60-64. Doi: 10.1109/ISCAIE.2014.701021

[16] Swayam Charania and Vidhi Vyas "SQL Injection Attack :Detection and Prevention" International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 03 Issue: 04 | Apr-2016 www.irjet.net p-ISSN: 2395-0072

[17] Nadya ElBachir El Moussaid and Ahmed Toumanari, "Web Application Attacks Detection: A Survey and Classification" International Journal of Computer Applications (0975 – 8887) Volume 103 – No.12, October 2014

[18] Vigna, G., Robertson, W. and Balzarotti, D. 2004. Testing network-based intrusion detection signatures using mutant exploits. In Proceedings of the 11th ACM conference on Computer and communications security (CCS '04). ACM, New York, NY, USA, 21-30.Doi: 10.1145/1030083.1030088

[19] Kruegel, C. and Vigna, G. 2003. Anomaly detection of web-based attacks. In Proceedings of the 10th ACM conference on Computer and communications security (CCS '03). ACM,New York, NY, USA, pp. 251-261. Doi:10.1145/948109.948144

[20] M. Auxilia, D.Tamilselvan, "Anomaly Detection Using Negative Secu-rity Model in Web Application,"IEEE 2010.

[21] Vigna, G., Valeur, F., Balzarotti, D., Robertson, W., Kruegel, C. and Kirda, E. 2009. Reducing Errors in The Anomaly-based Detection of Web-based Attacks Through the Combined Analysis of Web Requests and SQL Queries. Journal of Computer Security, 17,pp. 205-329, IOS Press.

[22] Le, M. and Stavrou, A. 2012. DoubleGuard: Detecting Intrusions in Multitier Web Applications. IEEE Transactions of Dependable and Secure Computing, 9, 4, pp. 512-525

[23] Juan Jose Garcia Adeva, Juan Manuel Pikatza Atxa," Intrusion Detec-tion in web applications using text mining," Journal of Artificial Intelli-gence - Elsevier 2006.

[24] Christopher Kruegel, Giovanni Vigna, William Robertson," A multi model approach to the detection of web based attacks", Journal of Com-puter Networks - Elsevier 2005.

[25] Chou, T. 2013. Security Threats on Cloud Computing Vulnerabilities. International Journal of Computer Science & Information Technology, 5, 3, pp. 79–88. Doi:10.5121/ijcsit.2013.5306

[26] R. Sekar," An Efficient Black box Technique for Defeating Web Appli-cation Attacks", Proc. Network and Distributed system security sump.(NDSS),2009.

[27] Ludinard, R., Totel, E., F. Tronel, V. Nicomettee, and Kaaniche, M. 2012. Detecting Attacks Against Data in Web Applications. Proceedings of the 7th International Conference on Risks and Security of Internet and Systems, Cork, Ireland, pp. 1-8.