A NEW APPROACH OF WEB ATTACKS CLASSIFICATION FOR TESTING SECURITY TOOLS AT THE APPLICATION LEVEL

[1] KARIM ABOUELMEHDI, [2]AHMED BENTAJER, [3]LOUBNA DALI, [4]NACER SEFIANI
[1,4]The Mohammadia Engineering School laboratory l'ASTIMI, Rabat, Morocco
[2]University Cadi Ayyad, High School of Technology Safi, Morocco
[3]Research Lab D.CS, Bowie State University.Maryland, USA
E-mail: [1]karim.abouelmehdi1@gmail.com

ABSTRACT

Classification of web attacks is the focus of cyber security nowadays. Therefore, it is still infancy, but like the cyber security, is evolving constantly. Therefore, classification needs to be properly organized in order to choose the best tests, to better choose the fixtures, for web security systems. The goal of this paper is to classify the web attacks using the classification of OWASP, the Open Web Application Security Project, which deals with the top ten security web attacks.

We present and describe a classification method that assists to figure out the growth of web attacks. The particularity of our approach is not only it gives relevant information about the current web attacks but also can predict and estimate the likely future attacks. Our method is tree based classification.

Keywords: *Web Attacks, Attacks Categories, Web Security Tools, Classification Tree*

1. INTRODUCTION

Even though there are a lot of security tools to use in web attacks, the greatest difficulty in using them is to study their effectiveness and make sure that they are working properly. To solve this issue, we have proposed a tree based classification method along with the top ten vulnerabilities in web application identified by OWASP. More specifically, classifying these attacks so that a test will take only one element of each class.

Several studies has been done in order to easily identify and analyze the common features of known venerability [1,2, 3].As a result several non profit organizations have been created. For example, the Web Application Security Consortium (WASC) which is made up of an international group of experts, industry practitioners, and organizational representatives who produce open source to standardize the web security. In addition to WASC, there is The Open Web Application Security Project (OWASP) which makes software security visible. so that decision about true software security risks can be made. For the latter, OWASP publishes OWASP10 which ranks the top 10 dangerous attacks in the web application. We will use this ranking to establish a new ranking.

The aim of our approach is to find a suitable classification algorithm which can test systems for web application attacks. To do that, first we grouped attacks into several classes where each class has a set of attacks. Then, test one attack and generalize to the test to all attacks that belongs to the same group or class.

2. PRESENTATION OF WEB ATTACKS

Web applications are targeted by a wide variety of attacks, some of which are known and dangerous and others unknown. For this reason, several databases are dedicated to save the attackers vulnerabilities, such as CVE (Common Vulnerabilities and Exposures) NVD (national vulnerability database) and VUPEN (vulnerability penetration testing). Because of numerous web attacks on application web, many organizations are forced and merged to asses and improve the security level of their web applications.

3. SECURITY TOOLS AGAINST WEB ATTACKS

3.1 Firewall-WAF

Firewalls are filtering tools that block unwanted addresses (Zone of unauthorized IP addresses+ port number) and allow access to others (zone of authorized IP addresses + port number). [12]

In web applications, firewall has a role as an agent ("inverse proxy") [13] to check the exchanged requests between the client and the web application, it is also called WAF (web application firewall). The customer sees the WAF as his/her web server in inverse-proxy mode, this latest hide the infrastructure that host the web application from network and users.

The server side is protected by WAF firewall, through attacks signature database [14], which does verify the client requests as well.

Being a barrier against attempts, The WAF is accordingly used to prevent attacks, detect and block almost all known attacks. Otherwise, with all its features, using the WAF by itself is not enough.[15]

3.2 Intrusion Detection Systems (IDS)

3.2.1 Definition of Intrusion Detection Systems (IDS)

An intrusion detection system (IDS) is a mechanism that detects abnormal or suspicious activity on a target data. And, several studies have been done over the past 20 years to increase their effectiveness. There are two types of IDS: the Network IDS (NIDS) and Host IDS (HIDS). In this paper, and since they belong to the application layer, we will focus on (HIDS). There are two detection approaches used by IDS:

- Scenario approach

It is based on comparing the observed interaction with a corresponding reference signatures or known attack scenarios, if such a signature is found in the database of signatures, in this case it will be considered as an attack. [16, 17]

- behavioral approach

IDS generate an alert if there is a deviation from the normal behavior and the observed behavior [18].

3-2-2 The IDS of application layer

In this section, we will present three intrusion detection tools of application layer whose source is free on the internet:

- w3af

Created by ANDRES RANCHO in 2006 and is one of the most efficient [19]. It is also free and allow users to access and modify its different modules. One of these modules is the SQLi Module that detects SQL injections in authentication forms [20]

- Skip Fish

Developed by Google and it is based on [21]. It detects a large number of vulnerabilities on web servers. In addition it checks the site out and collects only all the appearing stable pages. To check if a page is stable, Skip Fish applications sends 15 identical queries and compare answers. If the answers are similar then the page is considered stable.

- Wapiti

It is an open source and based on [22]. It detects SQL, XSS and LDAP injections, control operating system injection from an URL and mishandled files. Furthermore, it sends queries exploiting vulnerabilities, researching error messages in the resulted answers.

4. EXESTING CLASSIFICATION OF WEB ATTACKS

4.1 WASC Classification

"WASC threat Classification" reports an exhaustive list of web applications security threats .Furthermore, they are grouped into the following six categories [10].

- ✓ Insufficient authentication: includes websites attacks whose target is the validation system of the users ID, service, or application.
- ✓ Insufficient authorization: covers all websites attacks whose target is the

verification system of user privileges, service or application trying to perform an application interaction.

- ✓ Client-side attacks: includes user’s attacks during the application performance.
- ✓ Command execution: contains all attacks that execute commands on any of the website architectural component.
- ✓ Information leakage: defines all the attacks discovering hidden features or any kind of information.
- ✓ Logical attack: characterizes attacks that use processes (change password system, account system...).

4.2. OWASP Classification

Unlike WASC who define all possible attacks on a web application, OWASP focus on the processing level, in fact, it covers the ten greatest security risks designed. The "OWASP Top 10" report help focusing on protecting the Web application from dangerous threats, as a consequence the protection will be cheaper and easier to implement rather than trying to shelter from all dangers.

- Injection: occurs when sending a unreliable data that might be either command or query to an interpreter [4]. So the attacker's non respective data can easily trick the interpreter and let him/her execute unusual commands or access to unauthorized data [5].

- Cross-Site Scripting (XSS) : occurs when unreliable data is sent it to a web browser without validation [6], whenever an application has data causing diversion of user sessions ,as result the user is redirected to another undersigned websites depending also on XSS, whether it allows attackers to execute scripts in the victim browser[9].

- Broken authentication and session management: occurs if the functions are incorrectly implemented, so the attackers can easily compromise keys, session's tokens, in addition to appropriating the user’s identities thanks to vulnerabilities [7].

- Insecure direct object reference: occurs when a developer has a reference to an internal variable, as a record of database, for example a file name, file, or a key database. However, an access control or protection is necessary to avoid the attacker’s use of these references and access to unauthorized data.

- Cross-site request Forgery (CSRF): forces the browser of an authenticated victim to send an HTTP request including its information and cookie session to vulnerable web application. Legitimating the victim [6]. The attacker generates requests through victim's browser.

- Security Misconfiguration: occurs when the application servers, Web servers, database server, and the platform are not safe or properly configured [7].

- Sensitive data Exposure: occurs when Web application does not properly protect sensitive and confidential data, such as credit card numbers [6].

- URL Restricted access: happens Whenever URL access is unsecure. So, it is necessary for the applications to control their access, hence, the attackers cannot force the URLs to access to hidden pages [8].

- Missing function level access control: occurs when applications fail to encrypt and secure the confidentiality and integrity of sensitive network traffic. Even if they do, they carry less powerful algorithms, using expired or invalid certificates, so the deployment remains incorrect [9].

- Invalidated redirects and forwards: occurs when a user is redirected to another web page, determining the destination pages using unreliable data. In fact, the lock of validation is good advantage for the attackers, they redirect victims to phishing sites or malware, also access to unauthorized pages using footnotes[8]

5. OUR CLASSIFICATION APPROACH

It should be noted that the decision tree is a testing tool that is used in software development as well as the artificial intelligence. Our method is based on OWASP and decision tree. Using the top ten risks cited by OWASP10, we presented each risk in form of decision tree. This tree is reduced so that will contain all the attacks categories. It will also predict potential attacks that are not yet created. Furthermore, the tree created will generates test cases which can be used as a tool to test attacks assessment.

While there is study by GADELRAB [11] that classify attacks using decision tree, our work in particular focus on the top ten web attacks that OWASP published and it was the last quote made our days.

6. OUTILNE OF OUR APPROACH

Step 1

First, we begin extracting properties of each attack of the top ten listed by OWASP. The following properties are found:

-Attack Type: viruses, Trojan horses, denied service etc..

-Average: method or algorithm used by the attacker to achieve his/her goal.

-Target: data or functions that wish to reach the attacker.

- Objective of attacker: financial purpose, terrorism, personal purpose, etc........).

-Privilege: files (right to read or write, etc. ....).

-Vulnerability Exploited by the attacker: which are the weak points.

Step 2

Rank Keep the most important properties and take the least important ones off, which are:

-Attack source: Our classification aim is to test the performance of attacks detection tools. That is preventing the attack regardless of its source.

-Attack Type: the detection system must detect all types of known attacks, which is random in our case.

-Attack objective: Regardless of the attack target, the goal is to prevent it before achieve its goal.

Step 3

Draw a diagram of each attack cited by OWASP 10. Then, save only the five most important properties so that we can reduce the final diagram into all possible attacks.

7. RESULTS

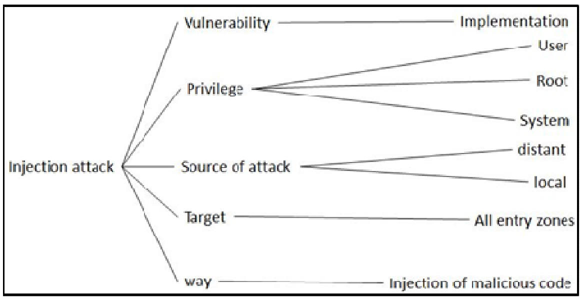

Figure1:Injection

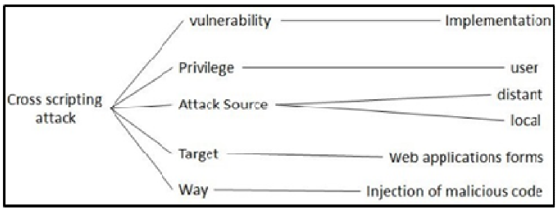

Figure 2:Cross Site Scripting

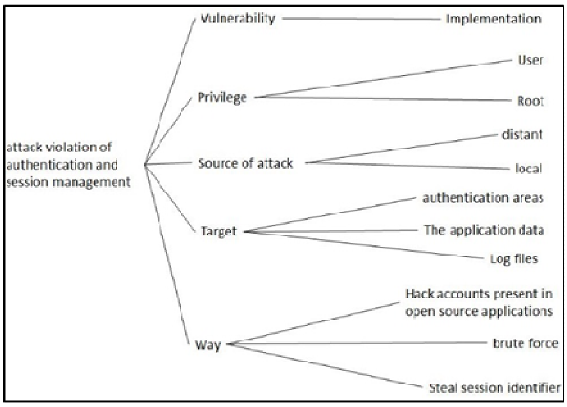

Figure 3: Broken authentication and session management

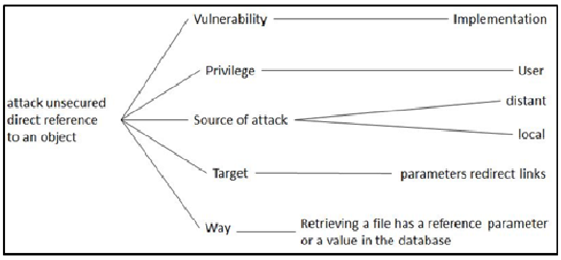
attack unsecured direct reference to an object
Vulnerability
Implementation
Privilege
User
Source of attack
distant
local
Target
parameters redirect links
Way
Retrieving a file has a reference parameter or a value in the database

Figure 4: Insecure direct object reference

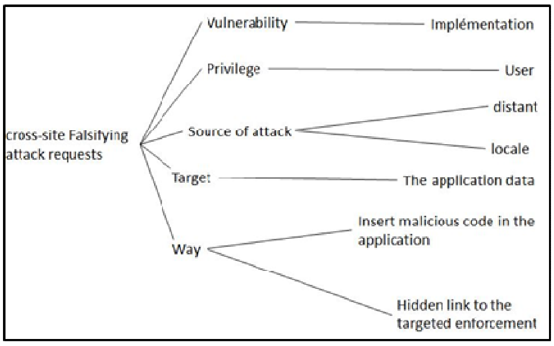
cross-site Falsifying attack requests
Vulnerability
Implémentation
Privilege
User
Source of attack
distant
locale
Target
The application data
Way
Insert malicious code in the application
Hidden link to the targeted enforcement

Figure 5: Cross-site request Forgery

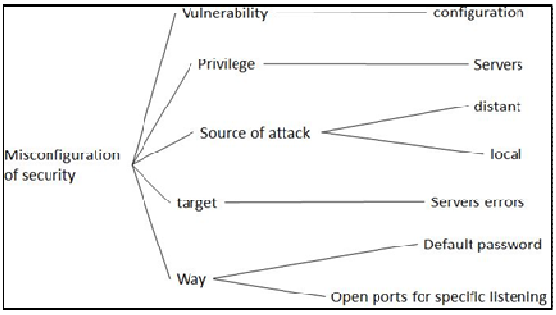

Figure 6: Security Misconfiguration

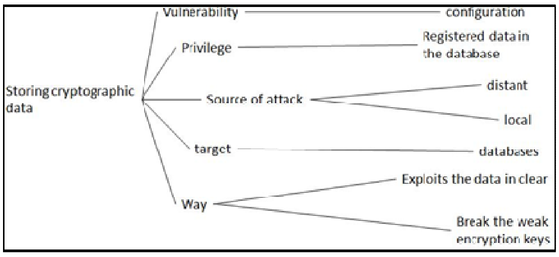
Storing cryptographic data
Vulnerability
configuration
Privilege
Registered data in the database
Source of attack
distant
local
target
databases
Way
Exploits the data in clear
Break the weak encryption keys

Figure 7:Sensitive data exposure

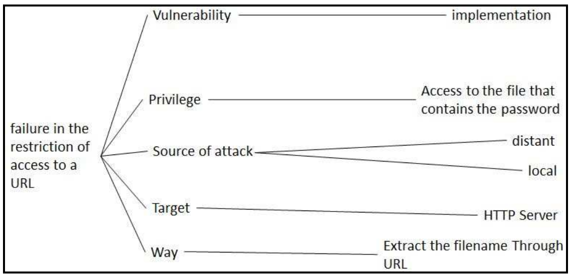
failure in the restriction of access to a URL
Vulnerability
implementation
Privilege
Access to the file that contains the password
Source of attack
distant
local
Target
HTTP Server
Way
Extract the filename Through URL

Figure 8: URL restricted access

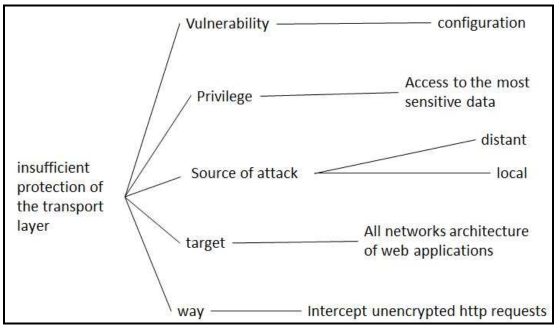
insufficient protection of the transport layer
Vulnerability
configuration
Privilege
Access to the most sensitive data
Source of attack
distant
local
target
All networks architecture of web applications
way
Intercept unencrypted http requests

Figure 9: Missing function level access control

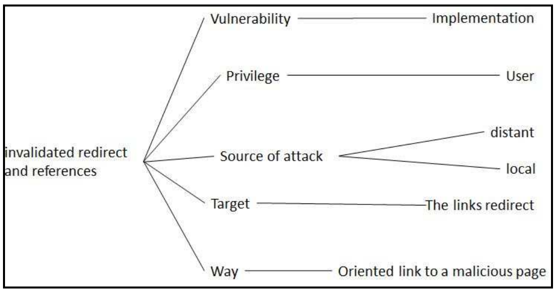

Figure 10: Invalidated redirect and forwards

To obtain the final diagram of our approach, we have eliminated the redundancy of properties so that the final attack pattern will not have any common properties. In addition, the source property of attack is also removed because our main objective is to test the performance of attack detection tools whatever the source is. (Figure 11)

8. DECISION OF OUR APPROACH

In our approach, we have classified attacks into categories, and each category has sub categories. This was based on our ranking that selects the most important attributes (e.g., source privilege obtained, medium, and target). In addition, we believe that the dimensions "type" and "detection technique" does not establish a clear categorization.

On the other hand, combinations of different suspected (considering the fine classification) results in the test cases 480. For example: (vulnerability = implementation, privilege = Root, target = logfiles Average = pirate open source accounts) is an attack that should test among the 480 test cases.

9. CONCLUSION AND PERSPECTIVE

It is becoming increasingly necessary to evaluate the robustness of each system web security regardless of the type of attack .This will help the administrator to check whether their system is secure or not. Another immediate research that we will pursue is that, we will test this approach on web attacks after classifying and saving them.

REFERENCES:

[1] M. S. Gad El Rab, A. Abou El Kalam, “Testing Intrusion Detection Systems: An Engineered Approach”, *IASTED International Conference on Software Engineering and Applications* (SEA 2006), Nov. 2006.

[2] Common Vulnerabilities an Exposures "CVE»:*http://cve.mitre.org/*

[3] Open Source Vulnerability Data Base “OSVDB”: *http:// www.osvdb.org/*

[4] A. Kiezun, P. J. Guo, K. Jayaraman, M. D. Ernst. “Automatic Creation of SQL Injection and Cross-Site Scripting Attacks”. *ICSE*, p 199-209, 2009

[5] M. Contensin. “Web Application Security”, *in training PHP/MySQL* chapter 6. CNRS, 2007

[6] D. Gollmann. “Securing Web applications”. *in Information Security Technical Report*, chapter 1-9, Elsevier, 2008

[7] J. Scambray, V. Liu et C. Sima. “Hacking Exposed Web Applications: Web Application Security Secrets and Solutions”, *Osborne/McGraw-Hill*, 482p, 2010

[8] Z. Su et G. Wassermann. “The essence of command injection attacks in Web applications”. *in POPL'06 Conference*, ACM SIGPLAN Notices, p372-384, 2006

[9] Y.-W. Huang, C.-H. Tsai, T.-P. Lin, S.-K. H., D.T. Lee et S.-Y. Kuo. “A testing framework for Web application security assessment”.*in Computer Networks*, pages 739-761, 2005

[10] *http://projects.webappsec.org/f/WASC-TC-v2_0.pdf*

[11]M.Gadelrab, “Evaluation of intrusion detection system “*a doctoral thesis* pages35-40, 2008

[12] W. R. Cheswik, S. M. Bellovin, “Firewalls and Internet Security”, *Addison-Wesley*, 1994.

[13] R.Barnett ET B.Rectanus “WAF Virtual Patching” *Workshop: Securing WebGoat with ModSecurity*, Breach Security, 2009

[14] Mod Security:” Open Source Web Application Firewall “:*http://www.modsecurity.org/*

[15] Bee Ware - Web Security : *http://www.bee-ware.net/fr/*

[16] H.Debar, M.Dacier, A.Wespi, “A revised taxonomy for intrusion detection systems”, *Annales des Telecommunications*, vol. 55, pp. 361-378, 2000.

[17] Y.Deswarte, “Chapitre 1 : La sécurité des systèmes d’information et de communication
“, In Sécurité des réseaux et des systèmes répartis, dir. Yves Deswarte, Ludovic Mé, Traité IC2, Hermès, *ISBN 2-7462-0770-2*, pp 15-65, octobre 2003.

[18] A. Kartit, A. Saidi, F. Bezzazi, M. El Marraki & A. Radi, "A new approach to intrusion

detection system", *Journal of Theoretical and Applied Information Technology*, Vol. 36, No. 2, 2012, pp. 284-289

[19] Top 10 vulnerability scanners: http*://Sectools.org/web-scanners.html*[accessed on 02/22/10]

[20] W3af - Source Forge: http*://w3af.sourceforge.net/*

[21] Skipfish - Google Code: *https://code.google.com/p/skipfish/*

[22] Wapiti - *Source Forge: wapiti.sourceforge.net/*

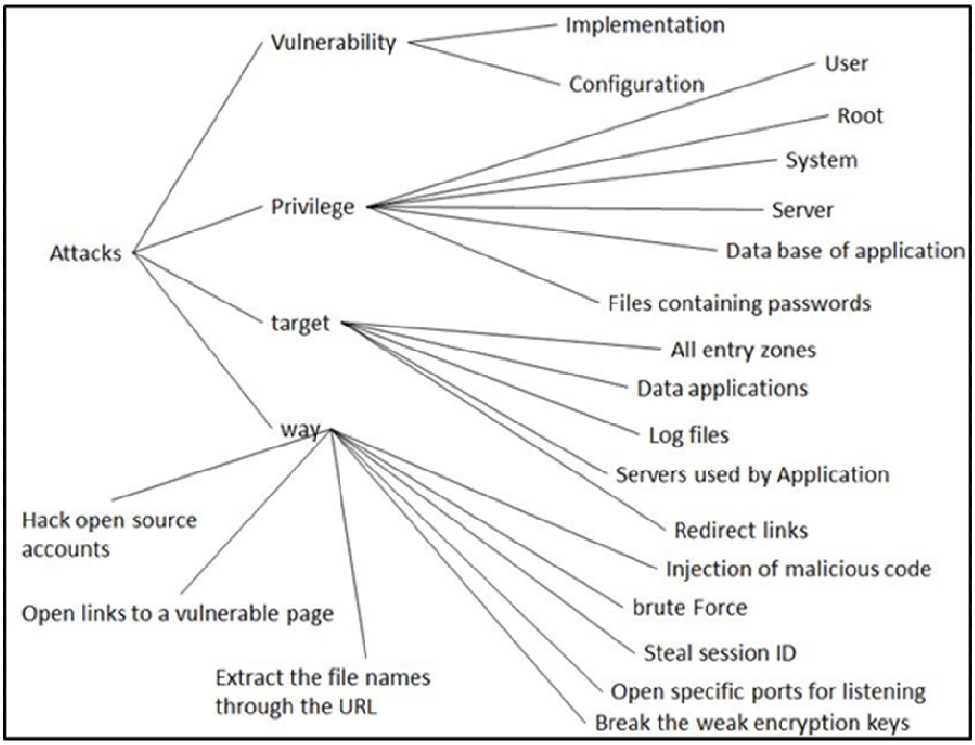
Attacks
Vulnerability
Implementation
Configuration
Privilege
User
Root
System
Server
Data base of application
Files containing passwords
target
All entry zones
Data applications
Log files
Servers used by Application
Redirect links
way
Hack open source accounts
Open links to a vulnerable page
Extract the file names through the URL
Injection of malicious code
brute Force
Steal session ID
Open specific ports for listening
Break the weak encryption keys

Figure 11: A new approach of web attacks classification