ISSN: 0976-3104
SPECIAL ISSUE: COMPUTER SCIENCE

ARTICLE

STUDYING OPEN SOURCE VULNERABILITY SCANNERS FOR VULNERABILITIES IN WEB APPLICATIONS

Deepika Sagar[1], Sahil Kukreja[2], Jwngfu Brahma[3], Shobha Tyagi[4], Prateek Jain[5*]

[1-4] *Department of Computer Science and Engineering, Manav Rachna International Institute of Research and Studies, Faridabad, INDIA*

[5]*Accendere KMS Services Pvt. Ltd, New Delhi, INDIA*

ABSTRACT

During the past few decades the digital market has seen an enormous growth in terms of cyber technologies and web applications. With the growth in digitalization the amount of risk is also mounting. A small mistake is capable of making the whole web application vulnerable to the attackers seeking it. Therefore, to save the developer's time, web application scanners are well placed to check for a group of known vulnerabilities all together. In our work, we have evaluated OWASP top 10 threats with three vulnerability scanners w3af, Skipfish and OWASP Zed Attack Proxy on vulnerable applications like DVWA. Scanning process starts with the insertion of the targeted vulnerable web application URL. A complete analyzed report is formed in each scenario that is further analyzed with the reports of other vulnerable web applications gathered through the application of the same process. At last the resultant running time of each scanner is compared to obtain the final tool that work efficiently with minimal time consumption. From three different dataset gathered from the tested scanning tools we conclude that OWASP ZAP performed better than the other scanning tools mentioned in this paper.

KEY WORDS
web applications, vulnerability scanner, vulnerability assessment tools, vulnerable web applications, OWASP top 10 threats.

INTRODUCTION

A Web application or a software application is a program that is used to run applications over internet to perform specific tasks. Such programs (applications) are stored on the web servers that can only be accessed by the web browsers. Some of the common web applications includes Google Docs, sheets, selenium and many others.

Vulnerable web applications mention to those applications that are vulnerable or exposed. Vulnerability here refers to the weakness that on encountering by an attacker can be well exploited. Such vulnerability can risk a small company to large organizations. Exploitation of any vulnerability by any unauthorized person does not only demand a huge recovery amount but also risk the reputation of the organization in the market. There are numerous threats that surrounds these applications such as Broken Authentication, Session Management, Cross-Site Scripting (XSS) and many others out of which SQL injection is the mostly used and is highly vulnerable. To prevent such threats from happening we use web scanners to find vulnerabilities in the web applications and the possible attacks that can be used by an attacker.

In this paper we try to test all the OWASP top 10 threats [1] on different vulnerable applications and analyze the outputs obtained. OWASP also known as Open Web Application Security project is an organization that focuses on improving software security and provide information to individuals, organizations, community, corporations, government agencies and universities. It is a non-for-profit organization that provides free materials that are under open software license.

The **OWASP** top 10 includes:

Injection: Attack in which the security is compromised by placing SQL commands or strings into the code. It is one of the most common hacking techniques in which SQL commands are manipulated into the input fields of the web application.

Broken authentication and session management: Security is compromised by exploiting leaks in the authentication process system or any flaws in the session management.

Cross site scripting: XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user supplied data using a browser API that can create JavaScript.

Broken Access Control: Attack that occurs when the restrictions on user's activity is not properly enforced that gives the attacker an opportunity to exploit these flaws and hence achieving the access to the authorized functionality of one's account or personal information of an organization or of the people authorized.

Security Misconfiguration: Attack that occurs sue to the flaws in the security configuration of an application/ server/ website or an organization. A small misconfiguration can put the data of the people at a stake.

Received: 8 Jan 2018
Accepted: 20 Feb 2018
Published: 4 March 2018

***Corresponding Author**
Email:
Prateek.jain@accendere.co.in
Tel.:9810245840

IIOAB JOURNAL
ISSN: 0976-3104

Sensitive Data Exposure: Sensitive data exposure is a type of security vulnerability where the web application fails to protect confidential data of an organization and hence exposes it to attackers for attacks. Sensitive data includes personal information, healthcare information, financial information that can be well used in attacks such as phishing, card fraud's, email spoofing and many more.

Insufficient Attack Protection: It denoted to the inefficiency of a web application to incorporate necessary tools and protecting elements for strong security. A majority of APIs is incapable of detecting, preventing and responding basic manual as well as automated attacks. This contains weak input validation, improper auditing and logging, captcha bypass.

Cross-Site Request Forgery: Cross Site Scripting Forgery attack includes malicious site that sends requests to the web application and hence take over the control of the whole functionality of the target website that the user is authenticated to. In this attack basically, the user's browser is fooled to perform unintended actions without the knowledge of the victim.

Using Components with Known Vulnerabilities: In this type of attack the vulnerable components such as libraries, software modules that run with the same privileges as the application could be used to compromise the security.

Under protected APIs: Most of the 3rd party APIs present in the market are unprotected and contains numerous vulnerabilities that the users are mainly unaware of. Such APIs take over the control once the user give the potential chance of it and hence compromise user's sensitive information that can be then well exploited [Table-1].

Table 1: Top 10 Vulnerability (2013)

VULNERABILITIES	THREAT AGENTS	EXPLOITABILITY	PREVALENCE	DETECTIBILITY	IMPACT	BUSINESS IMPACTS
1. INJECTION	APP SPECIFIC	EASY	COMMON	AVERAGE	SEVERE	APP SPECIFIC
2. AUTHENTICATION AND SESSION MGT	APP SPECIFIC	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	APP SPECIFIC
3. CROSS SITE SCRIPTING	APP SPECIFIC	AVERAGE	VERY WIDESPREAD	EASY	MODERATE	APP SPECIFIC
4. INSECURE DIRECT OBJECT REFERENCES	APP SPECIFIC	EASY	COMMON	EASY	MODERATE	APP SPECIFIC
5.SECURITY MISCONFIGURATION	APP SPECIFIC	EASY	COMMON	EASY	MODERATE	APP SPECIFIC
6.DATA EXPOSURE	APP SPECIFIC	DIFFICULT	UNCOMMON	AVERAGE	SEVERE	APP SPECIFIC
7. MISSING FUNC. LEVEL ACCESS CONTROLS	APP SPECIFIC	EASY	COMMON	AVERAGE	MODERATE	APP SPECIFIC
8. CROSS-SITE REQUEST FORGERY	APP SPECIFIC	AVERAGE	COMMON	EASY	MODERATE	APP SPECIFIC
9.USING COMPONENTS WITH KNOWN VULNERABILITY	APP SPECIFIC	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	APP SPECIFIC
10.UNVALIDATED REDIRECTS	APP SPECIFIC	AVERAGE	UNCOMMON	EASY	MODERATE	APP SPECIFIC

EXPERIMENTAL ENVIRONMENT

In this experimental research we used vulnerable web applications and the vulnerability assessment tools to carry out different attacks any generate a report on the basis of the output we received.

Vulnerable web applications [2]

For testing and evaluating the web vulnerability scanners, a vulnerable test environment is needed, this need for environment is fulfilled by Vulnerable Web Applications that are specially designed to provide users, the environment to test their attacks without causing any intended harm to the organization. For our experiments we ran the apps on windows, Linux and Finally on OWASP Virtual Machines.

DVWA: Damn Vulnerable Web Application [3] or shortly known as DVWA is a PHP/MySQL based vulnerable web application [4] that aims to be an aid to the security professionals and students alike in learning and testing their skills in a safe and legal environment and to help web developers better understand the process of securing web application.

ISSN: 0976-3104

Evaluated web vulnerability scanner

We performed the Evaluation of the following vulnerability scanners in Windows 10 creator's update and Kali Linux machines with i5 Intel processors.

OWASP ZAP: The OWASP Zed Attack Proxy (ZAP) [5] is an easy to use and open source intrigrated web application penetration testing tool designed to be used by beginners and professionals alike and also for developers and functional testers with low experience of security penetration testing [6]. Written in Java, ZAP is available across all the major operating systems including windows, OS X and almost all the destros of Linux.

Skipfish: Skipfish [7] is an active web application security reconnaissance tool by Google that prepares an interactive sitemap for the targeted site by carrying out a recursive crawl and dictionary-based probes. The resulting map is then annotated with the output from a number of active but mostly non-disruptive security checks. The final report generated by the tool is meant to serve as a foundation for professional web application security assessments [8]. Skipfish come handy in determining if the code is vulnerable [9] to scripting and injection attacks.

w3af: w3af [10] is a web application attack [9] and audit framework that aims at creating a framework to help people secure their web applications by finding and exploiting the vulnerabilities in the web application. w3af provides an easy to use GUI for its framework for the general users. Both the w3af core [11] and plugins are fully written in Python, more than 130 plugins in the framework makes it easy to identify most of the known vulnerabilities.

Table-2 and Table-3 show the general characteristics of vulnerable web applications and web application scanners displaying their version in web application and version and operating system in application scanners.

Table 2: General characteristics of vulnerable web applications

WEB Applications	DVWA
VERSION	1.10

Table 3: General characteristics of web scanners

COMPANY	OWASP ZAP	SKIPFISH	W3af
VERSION	2.6.0	2.10b	1.1
OPERATING SYSTEM	Windows Linux	Windows Linux	Windows Linux

Tables-4-6 display the input vector support of the vulnerability assessment scanners taking different parameters under consideration.

Table 4: Input vector support of tools

	OWASP ZAP	Skipfish	W3af
GET	✓	✓	✓
POST	✓	✓	✓
COOKIE	✓	✓	✓
HEADER	✓	✓	✓
SECRET	-	-	✓
PName	-	-	✓
XML	✓	-	-
XML Attributes	✓	-	-
XML Tag	✓	-	-
JSON	✓	-	-
DIR	✓	-	✓
FIILE	✓	-	✓
PATH	✓	-	✓
CMDExec	✓	✓	✓

IIOAB JOURNAL
ISSN: 0976-3104

Table 5: Glossary of the input support vector parameters

	General Feature	Description
GET	HTTP Query String Parameters	Input parameters sent in the URL
POST	HTTP Body Parameters	Input parameters sent in the HTTP body
COOKIE	HTTP Cookie Parameters	Input parameters sent in the HTTP cookie
HEADER	HTTP Headers	HTTP request headers used by the application
SECRET	Secret HTTP Parameters	Non-visible valid HTTP parameters (such as GET to POST, etc)
PName	HTTP Parameter Names	HTTP parameter names used by the application
XML	XML Element Content	The content of XML elements
XmlATT	XML Attributes	XML attributes
XmlTAG	XML Tags	The names of XML tags
JSON	JSON Parameters	Parameters sent in JSON format
DIR	Directory Name Input Vector	Support for scanning the directory section in the HTTP URL
FILE	File Name Input Vector	Support for scanning the file name section (without extension) in the HTTP URL
Path	HTTP Path Input Vector	Support for appending to and scanning the HTTP path

Table 6: Audit Feature of the Evaluated Scanners

	OWASP ZAP	Skipfish	w3af
SQLi	✓	✓	✓
BSQLi	✓	✓	✓
SSJSi	-	-	-
RXSS	✓	✓	✓
PXSS	✓	✓	✓
DXSS	-	-	✓
JSONh	-	-	-
LFI	✓	✓	✓
RFI	✓	✓	✓
CMDExec	✓	✓	✓
UPLOAD	-	-	✓
REDIRECT	✓	✓	✓
CRLFi	✓	-	✓
LDAPi	✓	-	✓
XPAPHi	✓	✓	✓
MXi	-	-	✓
SSi	✓	-	✓
FORMATi	-	✓	✓
CODEi	✓	✓	-
XMLi	-	✓	-
ELi	-	-	-
BUFFERo	-	-	✓
INTERGERo	-	✓	-
CODEDisc	-	✓	✓
BACKUPf	✓	-	✓
PADDING	-	-	-
AUTHb	✓	-	✓
PRIVe	-	-	-
XXE	-	-	-
SESSION	-	-	✓
FIXATION	✓	-	-
CSRF	✓	✓	✓
ADOS	-	-	✓
COUNT	17	15	23

METHODS

The scanning process starts with the insertion of the URL into the input URL field of scanners mentioning the application to scan for vulnerability. Generally, Application Scanners consists of three main components that helps in completing the scanning process successfully that includes

- Crawling Component: after the insertion of the target URL the scanning process starts where the crawling components identifies all the reachable web pages as well as all the input points in the target application.
- Attacker Component: the analysis of the discovered data is done by the attacker component. For each input fields, for every form and for every test vectors of application scanners an attacker module is generated that triggers a vulnerability.

This data is then sent to the server to get the appropriate response.

- Analysis Component: the server response is analyzed and interpret it as per desired.

IIOAB JOURNAL
ISSN: 0976-3104

Scanners basically scan for two scanning mode Log and No_Log Mode. In the Log mode a proper set of result is maintained with proper logging of every results generated whereas in No_Log mode the scanners are redirected to the initial page and requested to scan for all the vulnerabilities. In the following tables we have shown total number of vulnerability count build into DVWA [Table-7] and then checked the result through the vulnerability scanners that we are using [Table-8].

On DVWA

Table 7: The total count of vulnerabilities (intentional) in DVWA

Vulnerability	Count
RXXS (Reflected Cross Site Scripting)	1
SXSS (Stored Cross Site Scripting)	1
SQLi	2
BSQLi (Blind SQL Injection)	1
CSRF (Cross Site Request Forgery)	1
LFI (Local File Inclusion)	1
CMDExec	1

Table 8: The total count of true positive detection in DVWA

VULNERABILITY	TOOLS		
	OWASP ZAP	Skipfish	W3af
RXXS	1	1	-
SXSS	1	1	-
SQLi	1	1	-
BSQLi	-	-	-
CSRF	1	1	-
LFI	1	-	-
CMD Exec	1	-	-

OBSERVATION AND RESULTS

On testing the application scanners for the vulnerabilities in web application we plotted some resultset on the basis of our experience that is shown in [Table-9 and 10].

Table 9: comparison

SCANNER	GENERAL FEATURES						
	GUI	CONFIGURATION	REPORT	STABILITY	PERFORMANCE	USAGE	SCANLOG
ZAP	YES	VERY SIMPLE	YES	VERY STABLE	FAST	VERY SIMPLE	YES
W3af	YES	COMPLEXC	YES	UNSTABLE	FAST	COMPLEX	YES
SKIPFISH	NO	SIMPLE	YES	STABLE	VERY FAST	SIMPLE	YES

Table 10: Glossary of the comparison table

SIMPLE: Easy to understand and performed.
COMPLEX: Difficult to understand and perform.
STABLE: stays fixed without any interruption or do not terminate in between the process.
UNSTABLE: fluctuate during processing and sometimes do not respond.

The result datasets of the scanners include input vector support of the tool, supported audit features and the total vulnerability count calculated by each scanner over different platforms. The running time of each scanner is gathered and transformed into a tabular format as shown in [Table-11]. [Fig.1] and [Fig.2] shows the time taken by each scanner. Furthermore, the table data is converted into a graph format to show and compare the efficiency of each tool in terms of time taken by them to complete the scanning of

IIOAB JOURNAL
ISSN: 0976-3104

vulnerable web application. The paper also presents true positive results collected by each tool that is obtained by checking as well as comparing resulted datasets with each other and with the documented specification of the tool published by their manufacturers. From all the datasets collected, the final result showed OWASP ZAP to be the best whereas w3af hold the last position after Skipfish that has an intermediate working performance.

Table 11: Running time of application scanners

SCANNER	RUNNING TIME ON DVWA
ZAP	2 min 50 sec (Fig 1)
w3af	5 hours 20 min (Fig 2)
Skipfish	1 min 48 sec (Fig 3)

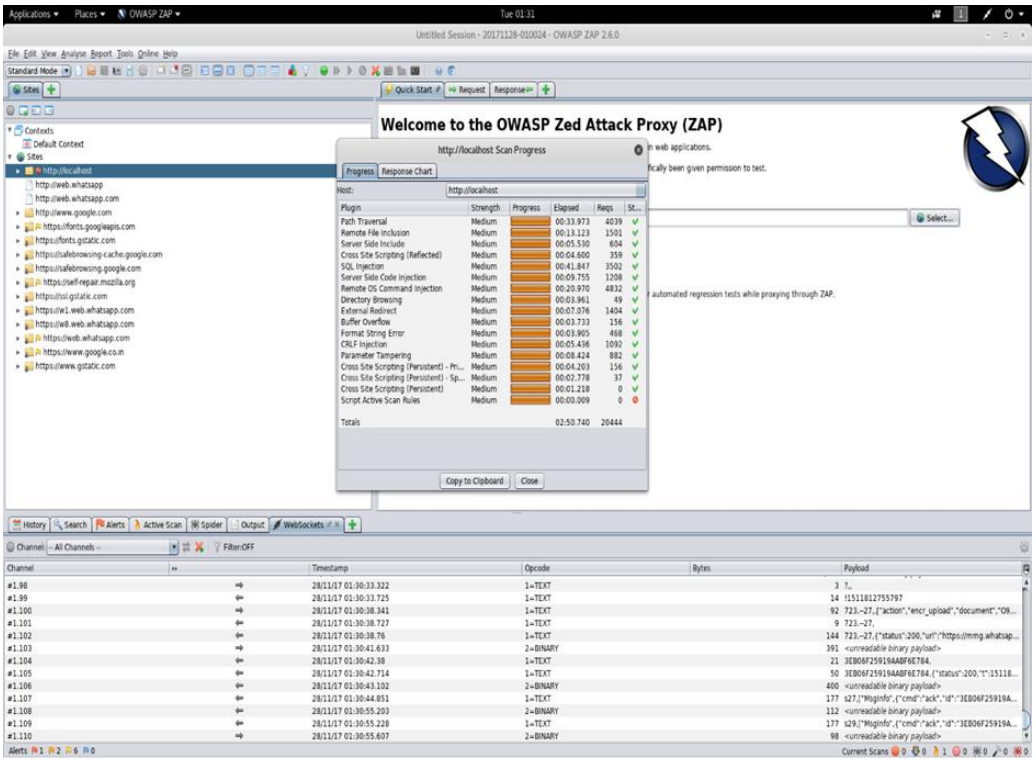

Fig. 1: Zap running time

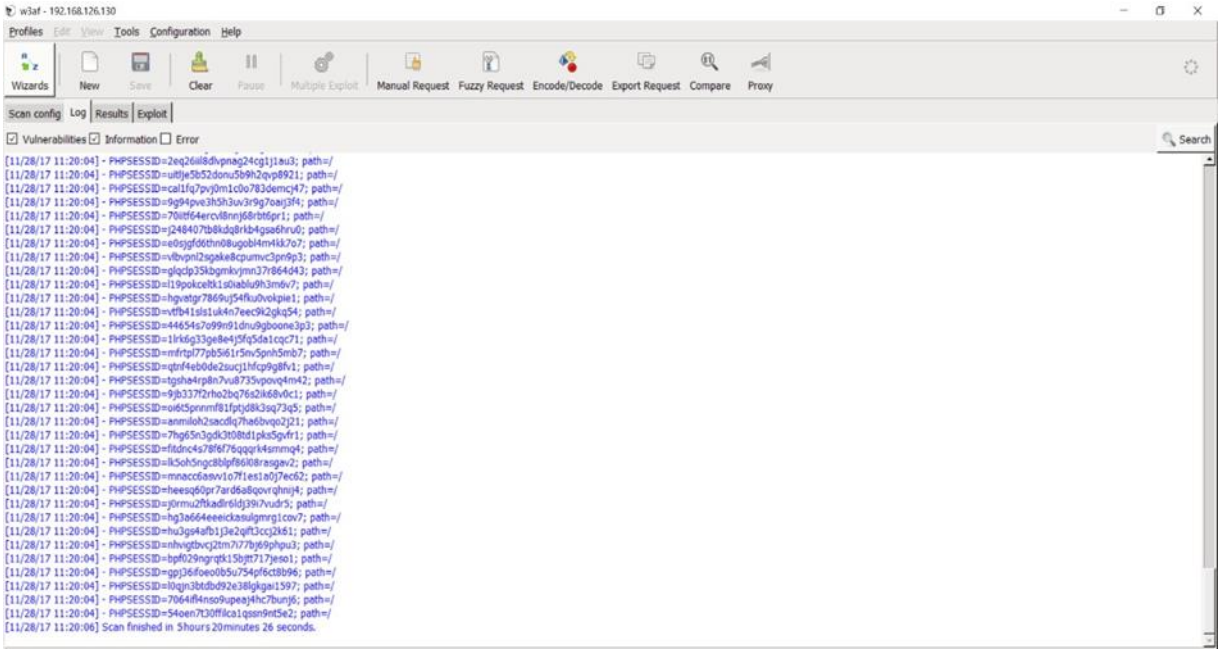

Fig. 2: w3af running time

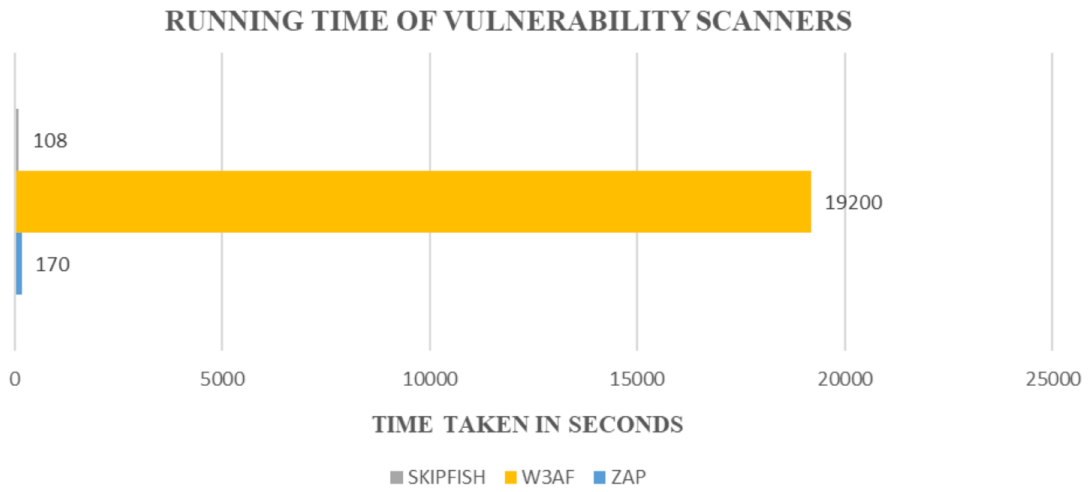

Fig. 3: Comparison of running time of application

CONCLUSION

After testing and analyzing the scanning tools w3af, ZAP and Skipfish on different parameters we conclude that OWASP ZAP has better results as compared to Skipfish and w3af. This is finalized by carefully examining the overall features contained by a scanner to the quality of result produced by each scanner Moreover, we learned that there doesn't yet exit a vulnerability scanner that can detect all of the OWASP Top 10 vulnerabilities all together.

CONFLICT OF INTEREST
None

ACKNOWLEDGEMENTS
We would like to sincerely bring our kind gratitude to Dr. Prateek Jain, Accendere Knowledge Management Services Pvt. Ltd for helping and guiding us in this paper formation.

FINANCIAL DISCLOSURE
None

REFERENCES

[1] Dewhurst R.[2012] Damn Vulnerable Web Application (DVWA).
[2] Makino Y, Klyuev V. [2015] Evaluation of web vulnerability scanners. In Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2015 IEEE 8th International Conference on 1:399-402
[3] Jovanovic N, Kruegel C, Kirda E. [2006] Pixy: A static analysis tool for detecting web application vulnerabilities. In Security and Privacy, 2006 IEEE Symposium on (pp. 6-pp). IEEE.
[4] Kindy DA, Pathan ASK. [2012] A detailed survey on various aspects of sql injection in web applications: Vulnerabilities, innovative attacks, and remedies. arXiv preprint arXiv:1203.3324.
[5] Evans SC. [2008]Securing WebGoat using ModSecurity, summer of code 2008. OWASP beta level, OWASP Foundation.
[6] Bennetts S. [2013] Owasp zed attack proxy. In AppSec USA 2013
[7] Mohammed R. [2016] Assessment of Web Scanner Tools. International Journal of Computer Applications (0975-8887), 133(5).
[8] Lecoeuche D. [2015] Tools for Computer Security (No. CERN-STUDENTS-Note-2015-082
[9] Muniz J. [2013] Web Penetration Testing with Kali Linux. Packt Publishing Ltd.
[10] Riancho A. [2011] w3af-web application attack and audit framework. World Wide Web electronic publication, 21
[11] Munadi R, Fajri TS, Meutia ED, Mustafa E. [2013] Analysis of SQL injection attack in web service (a case study of website in Aceh province). In Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), 2013 3rd International Conference on (pp. 431-435). IEEE.