From Security Perspective, the Quickest Way to Assess Your Web Application

GIAC (GWAPT) Gold Certification

Author: Mohammed Alduhaymi, m.s.aldeheme@gmail.com
Advisor: Chris Walker, chriswwalker@hotmail.com

Accepted: 1/24/2017

Abstract

The aim of this paper is to explain how to assess web applications with a fast, easy and effective method. A framework has been created as a Chrome Extension to solve two problems. 1. The first problem is when the IT team wants to know the security posture of their web application, but they did not have the budget/time to hire a penetration tester. Therefore, they can use this framework "WPSecAnalyzer Chrome Extension" to check their web application scores from a security perspective without having a deep knowledge of penetration testing. 2. The second problem is when the penetration tester wants to do the reconnaissance phase, he will use many tools, which will consume his time/effort. Consequently, to reduce the time/effort consumed he can use "WPSecAnalyzer Extension" to check many issues/vulnerabilities from one place with an efficient and effective method. The Chrome Extension which is called "WPSecAnalyzer" checks and verifies eleven issues/vulnerabilities on any website the end user visits, and provides him with a report based on the findings. The report will have the score of the website, as well as a list of the findings based on eleven issues/vulnerabilities.

1. Introduction

The Chrome Extension "WPSecAnalyzer" works as a framework to simplify assessing your web application. "WPSecAnalyzer" checks eleven issues/vulnerabilities on any website the end user visits. For example, once the user accesses google.com, WPSecAnalyzer will work and assess the web application by checking whether the website web application is vulnerable to one of the eleven issues, then it will display a report to the end user. WPSecAnalyzer has been built as client/server extension. In the client side it is used to display the landing page and the report to the end user. In addition, the server side is used to "handle the HTTP response, checking HTTP header fields, allowing HTTP methods and integration with shodan.io. It is also parsing the output and giving a score for each issue/vulnerability, then generating the report".

The score/grade of any website will change based on the findings. WPSecAnalyzer works in two modes. The first mode for WPSecAnalyzer when it retrieves the information successfully by integrating with shodan.io. The second mode is when WPSecAnalyzer could not retrieve all the eleven data points because the failure to integrate with shodan.io. In the first mode, the maximum score/grade will be 11 out 11. In the second mode, the score/grade will be 7 out 7.

1.1. What are the eleven issues/vulnerabilities that WPSecAnalyzer will check?

1. Does the website implement HTTPS?
2. Does the server implement one of these HTTP methods (TRACE, CONNECT, OPTIONS, DELETE, PUT)?
3. Does the server have unneeded open ports except 80 and 443 ports?
4. Is FTP port (21) open?
5. Are there three and more open ports?
6. Does the server implement X-XSS-Protection field in the HTTP response?
7. Does the server implement X-Content-Type-Options: nosniff field in the HTTP response?
8. Does the server implement X-Frame-Options field in the HTTP response?
9. Does the server implement HttpOnly field in the HTTP response?

10. Does the web server flag the cookie values to be secure; only sending the cookie values via HTTPS?
11. Is robots.txt file available on the server?

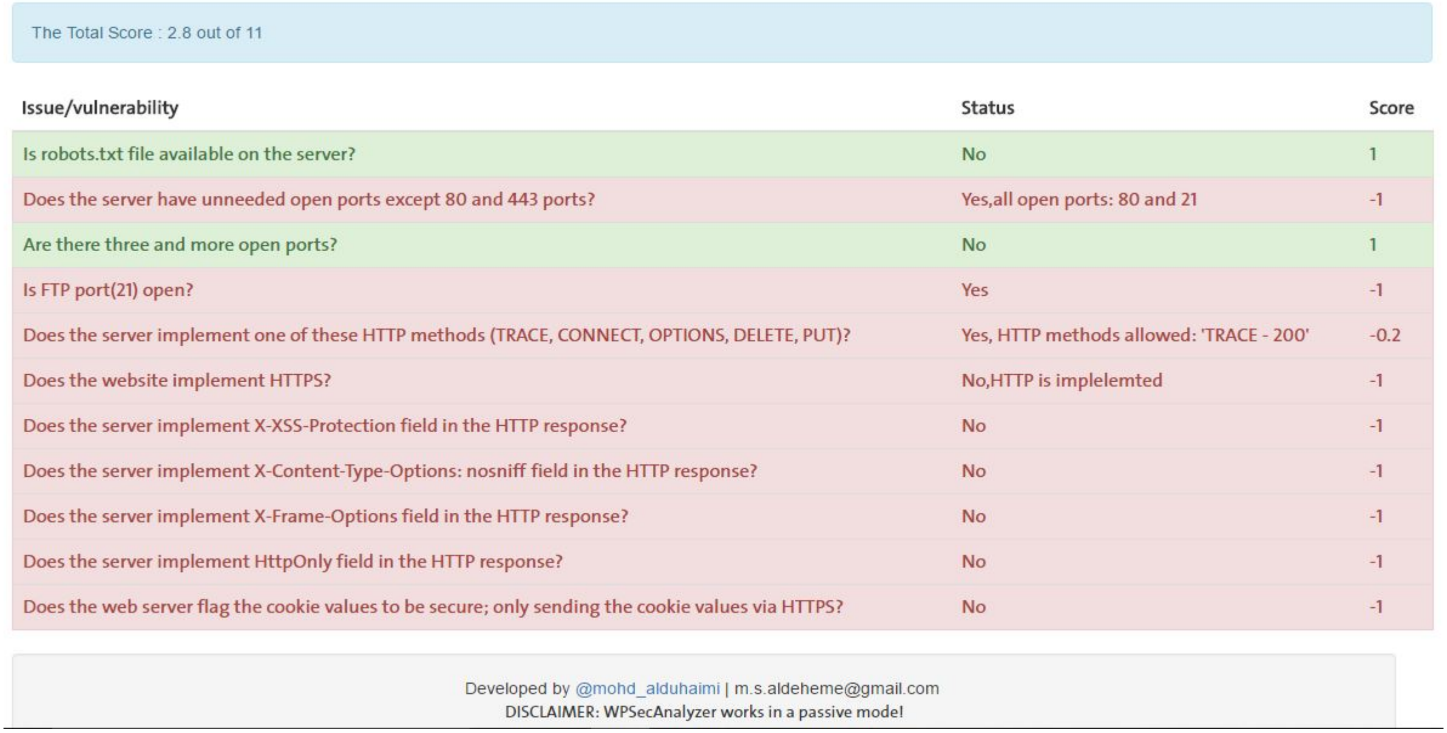
1.2. The research methodology

Why the eleven issues/vulnerabilities have been chosen, and what are the criteria in choosing them?

I have chosen the eleven issues/vulnerabilities because I wanted to create a tool to do passive testing not active testing, since active testing is more aggressive and could be blocked by the Intrusion Prevention System (IPS). In addition, it could blacklist WPSecAnalyzer's IP as a malicious IP because all the requests will be sent from the IP of WPSecAnalyzer server. Moreover I used the OWASP Testing Guide v4 (https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents) to check how OWSAP to understand how I could create a passive checklist. Therefore, I have created a WPSecAnalyzer tool which is using the eleven issues/vulnerabilities checking list. WPSecAnalyzer does not do "fuzzing/active testing" to figure out the issues/vulnerabilities, It works as a passive testing tool.

1.3. WPSecAnalyzer report

The generated report will be displayed, as below:



The Total Score : 2.8 out of 11

Issue/vulnerability	Status	Score
Is robots.txt file available on the server?	No	1
Does the server have unneeded open ports except 80 and 443 ports?	Yes,all open ports: 80 and 21	-1
Are there three and more open ports?	No	1
Is FTP port(21) open?	Yes	-1
Does the server implement one of these HTTP methods (TRACE, CONNECT, OPTIONS, DELETE, PUT)?	Yes, HTTP methods allowed: 'TRACE - 200'	-0.2
Does the website implement HTTPS?	No,HTTP is implelemted	-1
Does the server implement X-XSS-Protection field in the HTTP response?	No	-1
Does the server implement X-Content-Type-Options: nosniff field in the HTTP response?	No	-1
Does the server implement X-Frame-Options field in the HTTP response?	No	-1
Does the server implement HttpOnly field in the HTTP response?	No	-1
Does the web server flag the cookie values to be secure; only sending the cookie values via HTTPS?	No	-1

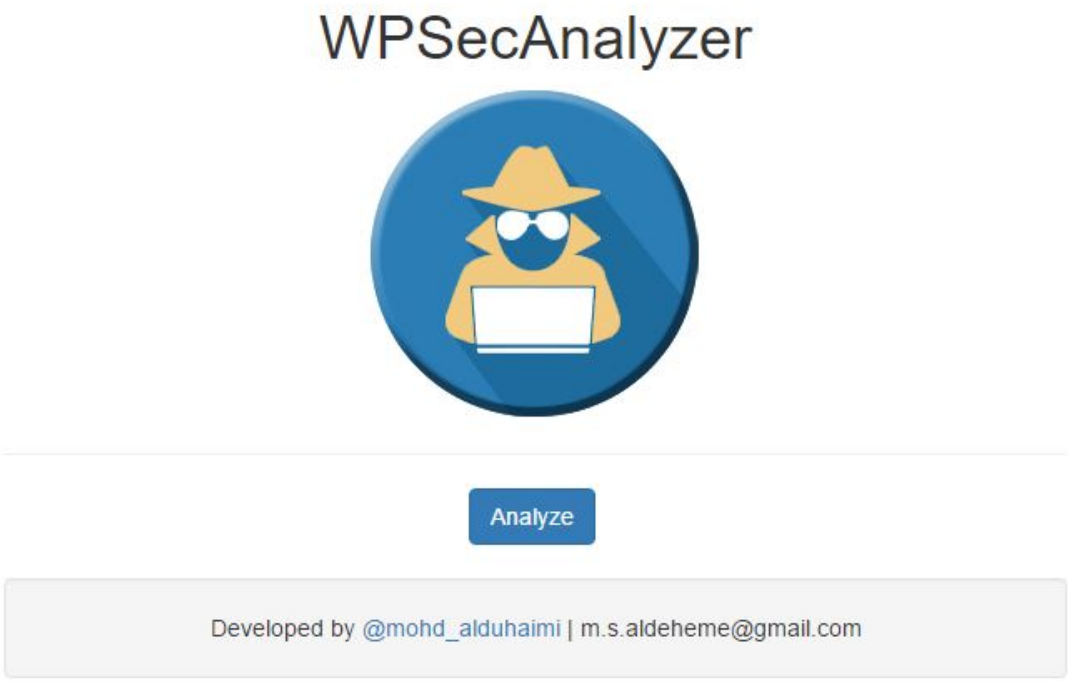
Developed by @mohd_alduhaimi | m.s.aldeheme@gmail.com
DISCLAIMER: WPSecAnalyzer works in a passive mode!

For more information and description about each issue/vulnerability, please visit this website: http://www.ratemywebsite.org/URLs/list.html

2. WPSecAnalyzer Client/Server Side

2.1. Client side code

In the client side you will find these files "background.html, and background.js, manifest.json".background.html is the home page for WPSecAnalyzer as below:



Developed by @mohd_alduhaimi | m.s.aldeheme@gmail.com

Background.js is a JavaScript used to fetch the URL that the end user wants to assess then send a URL as a POST request to the server side "scanPage.php", as below:

```
d = document;
fullurl=tab.url
var f = d.createElement('form');
f.action = 'http://localhost:81/serverSide/backend/scanPage.php?fullUrl='+fullurl;
f.method = 'post';
d.body.appendChild(f);
f.submit();
```

In the line highlighted in red, the PHP file, scanPage.php, will receive the URL then forward it by using Ajax to the main PHP file display.php.

Manifest.json is the configuration file for WPSecAnalyzer, as below:

```
{
 "manifest_version": 2,
 "name": "WPSecAnalyzer Plugin",
 "description": "From Security Perspective, the Quickest Way to Assess Your Web
Application",
 "version": "1.0",

 "browser_action": {
  "default_icon": "icon.png",
  "default_popup": "background.html"
 },
 "permissions": [
  "tabs"
  ]
}
```

The variable, “default_popup” value is "background.html" which is the home page, so when the user clicks of the extension's Icon, background.html will be shown. The value “default_icon” is the icon of the WPSecAnalyzer extension.

2.2. Server side code

WPSecAnalyzer server-side represents the business logic layer to do the assessment and calculating the score. The most important function is "process" located in the display.php file. This process function will receive only one parameter which is the full URL that will be assessed. After receiving the full URL, it will check the 11 issues/vulnerabilities. The first test is the process function which will check is if the HTTP/S is implemented, as coded below.

```
if($splitedUrl[0]=="https"){
$httpsFound=true;
}
else{
$httpFound=true;
}
```

In the code above, it will check if the website implements HTTPS or HTTP. In case it implemented HTTPS, it will set this parameter "$httpsFound" to true. However if it finds HTTP, it will set this parameter "$httpFound" to true. The idea from setting the parameters to true is to use them in the scoring. For more information about the scoring see section 3.

The process function checks if the web server implements specific HTTP header fields (X-XSS-Protection, X-Content-Type-Options: nosniff, X-Frame-Options, HttpOnly, secure string). To check if the previous headers are implemented, a HTTP/S request is made to get the HTTP header response fields as the code below:

```
$HttpHeaders=get_headers($fullUrl);
```

After getting all the HTTP header response fields, a "For Loop" will go through all the received fields and check if these fields " X-XSS-Protection, X-Content-Type-Options: nosniff, X-Frame-Options, HttpOnly, secure string" are present in the HTTP/S header response or not. Furthermore, and each time one of the fields in the HTTP header response is found, a "true" value for the parameter that represent the HTTP header is set. For example if the received HTTP Header has added this "X-XSS-Protection" field, true is assigned to this parameter $xssFound, then $xssFound value is added to the scoring. The full code is in the appendix section.

One of the issues WPSecAnalyzer checks is the robots.txt file, since robots.txt file is important for the attacker to find out the sensitive pages/files to use them to hack the website. Therefore, the process function will make a HTTP/S request for this URL path "**/robots.txt**" to check if robots.txt file available or not in the server, for example, if WPSecAnalyzer wants to check if robots.txt file available in google.com , WPSecAnalyzer will make a HTTPS request to this URL https://www.google.com/robots.txt .

Check the code below:

```
$robots=get_headers($prot.$urlWithoutProtocol."/robots.txt");

if (strpos($robots[0], '200')  !== false) {
        $score=$score-1;  }
```

In the code above, If the web server response status code is 200, that means is robots.txt file available on the server, so the score is decreased by one: $score=$score-1.

One of the important issues/vulnerabilities WPSecAnalyzer checks is how many open ports are there in the web server? Is the FTP port (21) open? Does the server have unnecessary open ports except 80 and 443 ports?

- WPSecAnalyzer integrated with shodan.io

  WPSecAnalyzer is integrated with shodan.io API to get the open ports and HTTP methods that are implemented in the server, shodan.io is a search engine which provides information about particular IP such as open ports, HTTP methods, HTTP banners and more, below you can see how WPSecAnalyzer is integrated with shodan by using python script.

```
API_KEY = "bEeEwKzhuL49CvhLLHChFOwB2THaVGeA"
IP=socket.gethostbyname(url)
api = shodan.Shodan(API_KEY)
host = api.host(IP)
allPorts=host['ports']
```

socket.gethostbyname(url) function will retrieve the IP address for a particular URL because Shodan accepts only IP string as input. Therefore, I need to convert the host name to IP address then sent it to Shodan. host['ports'] will extract the open ports from the array list "host" then assign it to this parameter "allPorts", the output will be like this : " [443, 80] "

- WPSecAnalyzer retrieving the HTTP methods from shodan.io

  The function httplib.HTTPConnection is used to check if these HTTP methods (TRACE','CONNECT','OPTIONS','DELETE','PUT) are implemented/allowed or not in the web server, as coded below.

```
for x in range(0, 5):
                conn = httplib.HTTPConnection(url)
                conn.request(notAllowedHttpMethods[x], '/')
                response = conn.getresponse()
```

```
                    httpMethods.append(notAllowedHttpMethods[x]+" /
"+str(response.status))

print allPorts,"-",httpMethods
```

After call.py executed, the result "list of the open ports, and the allowed HTTP methods in the web server" will returned to the process function for the scoring. Furthermore, to calculate the score for the HTTP methods, each time one of the above five HTTP methods are allowed and implemented in the web server, the score will decrease by 0.2, so in case the web server allowed all the five HTTP methods the score decreases by 1=.2*5. As another example, let's suppose the web server allowed PUT and TRACE methods, the score will be deceased by (2*0.2) =0.4, as coded below.

```
for($i=0;$i<count($httpOptions);$i++){
  $HttpStatus=split(' / ' , $httpOptions[$i]);
/* $HttpStatus[1] is the status of the HTTP method which is returned from shodan.io,
so in case the status was 200 for the TRACE method that means it's opened/allowed */
  if($HttpStatus[1]==200) {
    $score=$score-0.2;
  }
..
```

In the code above, $httpOptions parameter has a list of the HTTP methods that are returned from "call.py" file then submitted to the process function in display.php file.

WPSecAnalyzer will check if the web server has unneeded open ports except 80 and 443 ports, then it will decrease the score by 1 for every unneeded open port, as shown in below:

```
for($i=0;$i<count($openPorts);$i++){
if($openPorts[$i]!=80 && $openPorts[$i]!=443){
$score=$score-1;
break;
    }
}
```

WPSecAnalyzer will check if this port "21" is opened then it will decrease the score by 1, as coded below.

```
    if($openPorts[$i]==21){
          $score=$score-1;
    }
```

3. The Scoring Matrix

3.1. WPSecAnalyzer retrieves all the information

If WPSecAnalyzer retrieves all the eleven issues/vulnerabilities that are needed to assess the website the maximum score will be 11 out of 11 points. The scoring matrix will be as illustrated below:

Issue/vulnerability	Score equation/description	Example
Does the website implement HTTPS?	If the website implemented HTTP , the total score decreases by 1	
Does the server implement one of these HTTP methods (TRACE, CONNECT, OPTIONS, DELETE, PUT)?	(Number of the HTTP methods that are implemented/allowed in the server /5) Hint: HTTP methods are one of these (TRACE, CONNECT, OPTIONS, DELETE, PUT).	There are "TRACE and PUT" methods are implemented/allowed , so the total score decreased by (2/5) =0.4
Does the server have unneeded open ports except 80 and 443 ports?	If yes the total score decreases by 1.	
Is FTP port (21) open?	If the FTP port (21) open, the total score decreases by 1.	
Are there three and more open ports?	If yes the total score decreases by 1.	
Does the server implement X-XSS-Protection field in the HTTP response or not?	If yes the total score decreases by 1.	
Does the server implement X-Content-Type-Options: nosniff field in the HTTP response or not?	If yes the total score decreases by 1.	

Does the server implement X-Frame-Options field in the HTTP response or not?	If yes the total score decreases by 1.	
Does the server implement HttpOnly field in the HTTP response or not?	If yes the total score decreases by 1.	
Does the web server flag the cookie values to be secure or not?	If yes the total score decreases by 1.	
Is robots.txt file available in the server?	If yes the total score decreases by 1.	

3.2. WPSecAnalyzer could not retrieve all the information

If WPSecAnalyzer retrieved all eleven issues/vulnerabilities except (the open ports, and the HTTP methods implemented) from shodan.io, the maximum score will be 7 out of 7 points, so WPSecAnalyzer will ignore these Issue/vulnerability to be assessed because WPSecAnalyzer could not retrieve them from shodan.io website. The scoring matrix will be as illustrated as below:

Issue/vulnerability	**Score equation/description**	**Example**
Does the website implement HTTPS?	If the website implemented HTTP , the total score decreases by 1	
Does the server implement X-XSS-Protection field in the HTTP response or not?	If yes the total score decreases by 1.	
Does the server implement X-Content-Type-Options: nosniff field in the HTTP response or not?	If yes the total score decreases by 1.	
Does the server implement X-Frame-Options field in the HTTP response or not?	If yes the total score decreases by 1.	
Does the server implement HttpOnly field in the HTTP response or not?	If yes the total score decreases by 1.	
Does the web server flag the cookie values to be secure or not?	If yes the total score decreases by 1.	

Is robots.txt file available in the server?	If yes the total score decreases by 1.	

How does the website get the full mark "11 out of 11"?

By implementing the below controls/configuration below:

controls/configuration	**Score**	**Impact**
Disabling/closing FTP port(21)	The total score increased by 1	To prevent non-secure connection.
Implementing HTTPS	The total score increased by 1	
Disabling these HTTP methods(TRAC, OPTIONS,PUT,CONNECT,DELETE)	The total score increased by 1	
Only opening these ports 80 and/or 443, and close other ports	The total score increased by 1	
implementing X-XSS-Protection field in the HTTP response	The total score increased by 1	
implementing X-Content-Type-Options: nosniff field in the HTTP response	The total score increased by 1	
implementing X-Frame-Options field in the HTTP response	The total score increased by 1	
implementing HttpOnly field in the HTTP response	The total score increased by 1	
Secure the cookie values by adding "secure" string in the cookie to prevent sending it in HTTP connection.	The total score increased by 1	
Delete/disable robots.txt file	The total score increased by 1	
Only opening two ports as maximum	The total score increased by 1	
The Total Score Is 11		

How does the website get the full mark "7 out of 7 "?

By implementing the below controls/configuration:

controls/configuration	**Score**	**Impact**
Implementing HTTPS	The total score increases by 1	

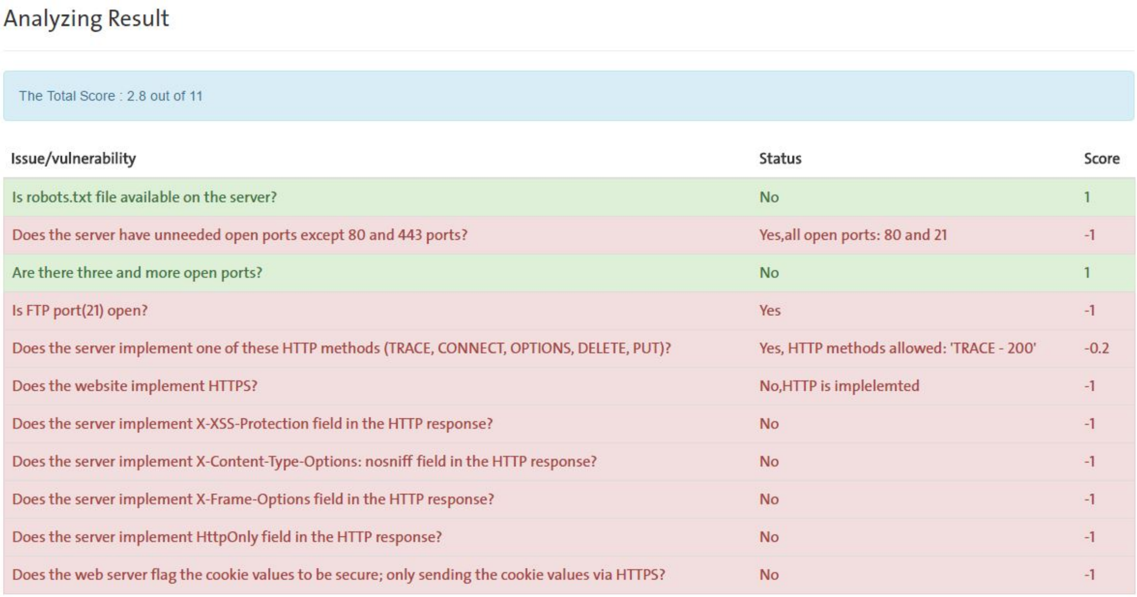
mohammed alduhaymi

implementing X-XSS-Protection field in the HTTP response	The total score increases by 1	
implementing X-Content-Type-Options: nosniff field in the HTTP response	The total score increases by 1	
implementing X-Frame-Options field in the HTTP response	The total score increases by 1	
implementing HttpOnly field in the HTTP response	The total score increases by 1	
Secure the cookie values by adding "secure" string in the cookie to prevent sending it in HTTP connection.	The total score increases by 1	
Delete/disable robots.txt file	The total score increases by 1	
The Total Score Is 7		

4. Case Studies

I ran WPSecAnalyzer on two different websites and explain the result and the remediation to fix discovered security issues.

The first case study was applied on this URL (http://www.ratemywebsite.org).After running WPSecAnalyzer on it. Here is the output:



Analyzing Result

The Total Score : 2.8 out of 11

Issue/vulnerability	Status	Score
Is robots.txt file available on the server?	No	1
Does the server have unneeded open ports except 80 and 443 ports?	Yes,all open ports: 80 and 21	-1
Are there three and more open ports?	No	1
Is FTP port(21) open?	Yes	-1
Does the server implement one of these HTTP methods (TRACE, CONNECT, OPTIONS, DELETE, PUT)?	Yes, HTTP methods allowed: 'TRACE - 200'	-0.2
Does the website implement HTTPS?	No,HTTP is implelemted	-1
Does the server implement X-XSS-Protection field in the HTTP response?	No	-1
Does the server implement X-Content-Type-Options: nosniff field in the HTTP response?	No	-1
Does the server implement X-Frame-Options field in the HTTP response?	No	-1
Does the server implement HttpOnly field in the HTTP response?	No	-1
Does the web server flag the cookie values to be secure; only sending the cookie values via HTTPS?	No	-1

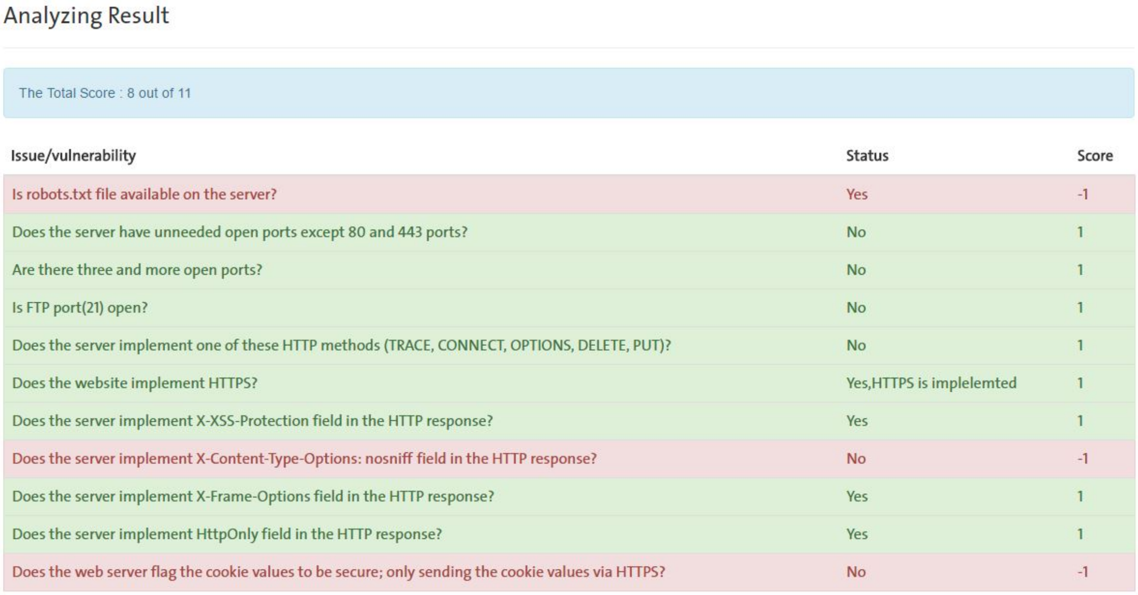
The grade of the website is "2.8 out of 11". Nine Security issues were not solved and colored in red, for example, there is a non-encrypted open port which is the FTP port (21), also the HTTPS protocol not implemented. The good thing they do not use the robots.txt file in their server and they does not have more than three open ports.

To solve the Nine Security issues. I have created a table to explain the recommended solutions, as below:

Issue/vulnerability	Status	Score	Recommendations	References
Is robots.txt file available on the server?	No	1		
Does the server have unneeded open ports except 80 and 443 ports?	Yes,all open ports: 80 and 21	-1	Only open these ports 80 and/or 443, and close other ports	
Are there three and more open ports?	No	1		
Is FTP port (21) open?	Yes	-1	Disable/close FTP port (21) because it's not secure protocol and can be sniffed.	
Does the server implement one of these HTTP methods (TRACE, CONNECT, OPTIONS, DELETE, PUT)?	Yes, HTTP methods allowed: 'TRACE - 200'	-0.2	Disable TRAC method.	
Does the website implement HTTPS?	No,HTTP is implelemted	-1	Implement HTTPS	
Does the server implement X-XSS-Protection field in the HTTP response?	No	-1	Implement X-XSS-Protection field in the HTTP response to prevent Cross Site Scripting (XSS) attack.	https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet#Bonus_Rule_.234:_Use_the_X-XSS-Protection_Response_Header

Does the server implement X-Content-Type-Options: nosniff field in the HTTP response?	No	-1	Implement X-Content-Type-Options: nosniff field in the HTTP response to prevent MIME content-sniffing attacks.	http://security.stackexchange.com/questions/12896/does-x-content-type-options-really-prevent-content-sniffing-attacks
Does the server implement X-Frame-Options field in the HTTP response?	No	-1	Implement X-Frame-Options field in the HTTP response to prevent Clickjacking attack.	https://www.owasp.org/index.php/Clickjacking
Does the server implement HttpOnly field in the HTTP response?	No	-1	Implement HttpOnly field in the HTTP response to prevent reading the cookie values via XSS attack.	https://www.owasp.org/index.php/HttpOnly
Does the web server flag the cookie values to be secure; only sending the cookie values via HTTPS?	No	-1	Secure the cookie values by adding "secure" string in the cookie to prevent sending it in HTTP connection.	https://www.owasp.org/index.php/SecureFlag

The second case study is Google, (https://www.google.com). After running WPSecAnalyzer on it and getting the output as below, Google took 8 out of 11.

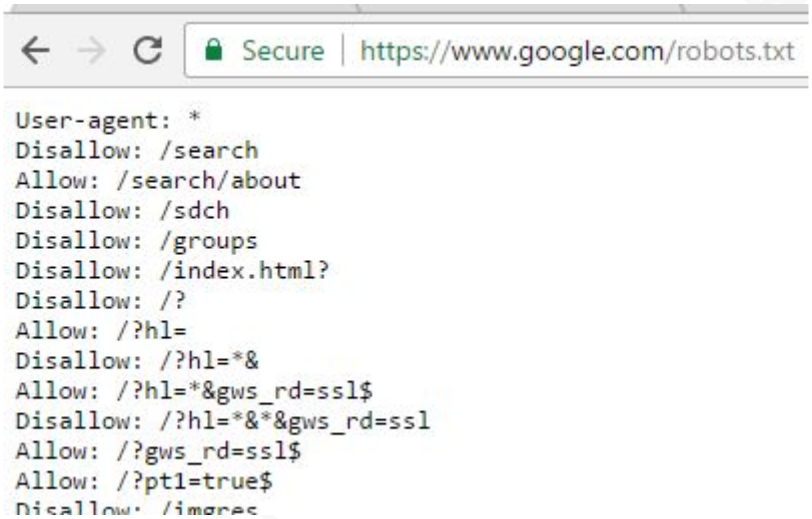


Analyzing Result

The Total Score : 8 out of 11

Issue/vulnerability	Status	Score
Is robots.txt file available on the server?	Yes	-1
Does the server have unneeded open ports except 80 and 443 ports?	No	1
Are there three and more open ports?	No	1
Is FTP port(21) open?	No	1
Does the server implement one of these HTTP methods (TRACE, CONNECT, OPTIONS, DELETE, PUT)?	No	1
Does the website implement HTTPS?	Yes,HTTPS is implelemted	1
Does the server implement X-XSS-Protection field in the HTTP response?	Yes	1
Does the server implement X-Content-Type-Options: nosniff field in the HTTP response?	No	-1
Does the server implement X-Frame-Options field in the HTTP response?	Yes	1
Does the server implement HttpOnly field in the HTTP response?	Yes	1
Does the web server flag the cookie values to be secure; only sending the cookie values via HTTPS?	No	-1

As you can see from the above image, Google took 8 out 11 which is mean they implemented 8 security controls. It seems Google using robots.txt file and to verify this issue you can go this URL (https://www.google.com/robots.txt) then you will see the robots.txt already there, as below:



```
User-agent: *
Disallow: /search
Allow: /search/about
Disallow: /sdch
Disallow: /groups
Disallow: /index.html?
Disallow: /?
Allow: /?hl=
Disallow: /?hl=*&
Allow: /?hl=*&gws_rd=ssl$
Disallow: /?hl=*&*&gws_rd=ssl
Allow: /?gws_rd=ssl$
Allow: /?pt1=true$
Disallow: /imgres
```

5. WPSecAnalyzer Download

You can download WPSecAnalyzer by visiting this website: https://chrome.google.com/webstore/detail/wpsecanalyzer-plugin/mkmomlbnjjjcnekjbedmnpeoknbegbdc. WPSecAnalyzer source code can be found in this URL: http://www.ratemywebsite.org/WPSecAnalyzerCode

6. Conclusion

WPSecAnalyzer has been built to help the security specialist to analyze the websites in a passive mode. WPSecAnalyzer will assess any website by checking the eleven issues/vulnerabilities then it will give a grade to the website based on the finding. Doing a passive testing will allow to the tester to avoid disturbing the system and the security devices such as Intrusion Detection/Prevention System (IDS/IPS), also the IP of the tester will not be blocked since it does not send a huge and malicious network traffic, moreover no need to make the IP address of the tester as a whitelisted IP. I believe WPSecAnalyzer will help assess your website (from a security perspective) in a passive mode and an effective way.

References

OWASP website. (2015). *Test HTTP Methods (OTG-CONFIG-006). Retrieved from* https://www.owasp.org/index.php/Test_HTTP_Methods_(OTG-CONFIG-006)

Stackoverflow website. (2011). *How to read a secure cookie using JavaScript. Retrieved from* http://stackoverflow.com/questions/8064318/how-to-read-a-secure-cookie-using-javascript

Securitywing website. (2014). *15 Penetration Testing Tools-Open Source. Retrieved from* http://securitywing.com/15-penetration-testing-tools-open-source/

Shodan website. (2014). *Looking up a host. Retrieved from* http://shodan.readthedocs.io/en/latest/tutorial.html#looking-up-a-host

Thoughtbot website. (2014). *How to Make a Chrome Extension. Retrieved from* https://robots.thoughtbot.com/how-to-make-a-chrome-extension

OWASP website. (2014). *Testing for cookies attributes (OTG-SESS-002). Retrieved from* https://www.owasp.org/index.php/Testing_for_cookies_attributes_(OTG-SESS-002)

Github website. (2016). *the official Python library for Shodan. Retrieved from* https://github.com/achillean/shodan-python

The SANS Institute*. (2016). SEC542: Web App Penetration Testing and Ethical Hacking. The SANS Institute.*

Appendix
The process function in Display.php

```
function process($fullUrl){
$outPut=new TemplatePower("temp/output.html" );
$outPut->prepare();
isset($httpsFound);
isset($httpFound);
isset($xssFound);
isset($ContentTypeOptionsFound);
isset($xFrameOptFound);
isset($cookie);
isset($cookieSecure);
isset($score);
isset($portcount);
isset($portsFlag);
$score=11;

//Does the website implement HTTPS?
$splitedUrl=split('://', $fullUrl);
if($splitedUrl[0]=="https"){
$httpsFound=true;
$prot="https://";
$p="https";
}
else{
$httpFound=true;
$prot="http://";
$p="http";
}

$urlWithoutProtocol=split('/', $fullUrl);
$urlWithoutProtocol=$urlWithoutProtocol[2];
//sending a HTTP request for the URL that the end-user wants to assess to get the
HTTP header response fields.
$HttpHeaders=get_headers($fullUrl);
$arrayCount=count($HttpHeaders);

for($i=0;$i<$arrayCount;$i++){
//Does the server implement X-XSS-Protection field in the HTTP response?
if (strpos($HttpHeaders[$i], 'X-XSS-Protection')  !== false) {
$xssFound=true;
}
//Does the server implement X-Content-Type-Options: nosniff field in the HTTP
response?
if (strpos($HttpHeaders[$i], 'X-Content-Type-Options: nosniff')  !== false) {
```

```
$ContentTypeOptionsFound=true;
}

//Does the server implement X-Frame-Options field in the HTTP response?
if (strpos($HttpHeaders[$i], 'X-Frame-Options')  !== false || strpos($HttpHeaders[$i],
'x-Frame-Options')  !== false) {
$xFrameOptFound=true;
}

//Does the server implement HttpOnly field in the HTTP response?
if (strpos($HttpHeaders[$i], 'HttpOnly')  !== false) {
$cookie=true;
}

//Does the web server flag the cookie values to be secure; only sending the cookie
values via HTTPS?
if (strpos($HttpHeaders[$i], ';secure')  !== false) {
$cookieSecure=true;
}

}

//Is robots.txt file available on the server?
$robots=get_headers($prot.$urlWithoutProtocol."/robots.txt");
if (strpos($robots[0], '200')  !== false) {
$score=$score-1;
$outPut->newBlock("Item");
$outPut->assign("Item","Is robots.txt file available on the server?");
$outPut->assign("Status","Yes");
$outPut->assign("alert","alert-danger");
$outPut->assign("Score","-1");
}
else{
$outPut->newBlock("Item");
$outPut->assign("Item","Is robots.txt file available on the server?");
$outPut->assign("Status","No");
$outPut->assign("alert","alert-success");
$outPut->assign("Score","1");
}

// run call.py file
$cmd2='C:\\xampp\\htdocs\\serverSide\\tools\\shodan\\shodan-python-master\\shodan-
python-master\call.py '.$urlWithoutProtocol;
$pythonResult=exec($cmd2, $output2);

if(isset($pythonResult) && $pythonResult!="" && $pythonResult!="error"){
```

```
$pythonResult=str_replace("[","",$pythonResult);
$pythonResult=str_replace("]","",$pythonResult);

$fullPythonResult=split('-', $pythonResult);
$openPorts=split(',' , $fullPythonResult[0]);
$flag=false;
 if(isset($openPorts) && $openPorts!=""){

//Does the server have unneeded open ports except 80 and 443 ports?
for($i=0;$i<count($openPorts);$i++){
   if($openPorts[$i]!=80 && $openPorts[$i]!=443){
   $score=$score-1;
   $outPut->newBlock("Item");
   $openPortsString=implode(" and ",$openPorts);
   $outPut->assign("Item","Does the server have unneeded open ports except 80 and
443 ports?");
   $outPut->assign("Status","Yes,all open ports: $openPortsString");
   $outPut->assign("alert","alert-danger");
   $outPut->assign("Score","-1");
   $flag=true;
   break;
   }
}
if($flag==false){
$outPut->newBlock("Item");
$outPut->assign("Item","Does the server have unneeded open ports except 80 and 443
ports?");
$outPut->assign("Status","No");
$outPut->assign("alert","alert-success");
$outPut->assign("Score","1");
}

//Are there three and more open ports?
if(count($openPorts)>3){
$score=$score-1;
$outPut->newBlock("Item");
$outPut->assign("Item","Are there three and more open ports?");
$outPut->assign("Status","Yes");
$outPut->assign("alert","alert-danger");
$outPut->assign("Score","-1");
}
else{
$outPut->newBlock("Item");
$outPut->assign("Item","Are there three and more open ports?");
$outPut->assign("Status","No");
$outPut->assign("alert","alert-success");
```

```
$outPut->assign("Score","1");
}

//Are FTP port (21) open?
$flag=false;
isset($countloop);
for($i=0;$i<count($openPorts);$i++){
   if($openPorts[$i]==21){
          $score=$score-1;
          $flag=true;
          $port21="21";
   }
}
if($flag==false){
$outPut->newBlock("Item");
$outPut->assign("Item","Is FTP port(21) open?");
$outPut->assign("alert","alert-success");
$outPut->assign("Status","No");
$outPut->assign("Score","1");
}
else{
$outPut->newBlock("Item");
$outPut->assign("Item","Is FTP port(21) open?");
$outPut->assign("Status","Yes");
$outPut->assign("alert","alert-danger");
$outPut->assign("Score","-1");
}
}
else{
echo "shodan  failed!";
}

//Does the server implement one of these HTTP methods (TRACE, CONNECT,
OPTIONS, DELETE, PUT)?
if(isset($fullPythonResult[1])){
$httpOptions=split(',', $fullPythonResult[1]);
if(isset($httpOptions) && $httpOptions!=""){
$flag=false;
isset($thereIsHttpMethods);
$thereIsHttpMethods="";
isset($countloop);
$countloop=0;

for($i=0;$i<count($httpOptions);$i++){
  $HttpStatus=split(' / ' , $httpOptions[$i]);
  if($HttpStatus[1]==200) {
    $flag=true;
    $thereIsHttpMethods=$thereIsHttpMethods." ".$HttpStatus[0]." - ".$HttpStatus[1];
    $score=$score-0.2;
    $countloop++;
  }

}
if($flag==false){
$outPut->newBlock("Item");
$outPut->assign("Item","Does the server implement one of these HTTP methods
(TRACE, CONNECT, OPTIONS, DELETE, PUT)?");
$outPut->assign("Status","No");
$outPut->assign("alert","alert-success");
$outPut->assign("Score","1");
}
else{
$outPut->newBlock("Item");
$outPut->assign("Item","Does the server implement one of these HTTP methods
(TRACE, CONNECT, OPTIONS, DELETE, PUT)?");
$outPut->assign("Status","Yes, HTTP methods allowed: ".$thereIsHttpMethods);
$outPut->assign("alert","alert-danger");
$outPut->assign("Score",($countloop/5)*-1);
}

}////
}
else{
$callpyfaild=true;
echo "<span class='label label-default'>WPSecAnalyzer could not retrieve all the
information, so the total score will change from 11 to 7 </span>";
}


}
else{

$callpyfaild=true;
echo "<span class='label label-default'>WPSecAnalyzer could not retrieve all the
information, so the total score will change from 11 to 7 </span>";

}


if(isset($httpsFound)){
$outPut->newBlock("Item");
$outPut->assign("Item","Does the website implement HTTPS?");
```

```
$outPut->assign("Status","Yes,HTTPS is implelemted");
$outPut->assign("alert","alert-success");
$outPut->assign("Score","1");
}
if(isset($httpFound)){
$outPut->newBlock("Item");
$outPut->assign("Item","Does the website implement HTTPS?");
$outPut->assign("Status","No,HTTP is implelemted");
$outPut->assign("alert","alert-danger");
$outPut->assign("Score","-1");
$score=$score-1;

}
if(isset($xssFound)){
$outPut->newBlock("Item");
$outPut->assign("Item","Does the server implement X-XSS-Protection field in the HTTP response?");
$outPut->assign("Status","Yes");
$outPut->assign("alert","alert-success");
$outPut->assign("Score","1");
}
else{
$score=$score-1;
$outPut->newBlock("Item");
$outPut->assign("Item","Does the server implement X-XSS-Protection field in the HTTP response?");
$outPut->assign("Status","No");
$outPut->assign("alert","alert-danger");
$outPut->assign("Score","-1");
}
if(isset($ContentTypeOptionsFound)){
$outPut->newBlock("Item");
$outPut->assign("Item","Does the server implement X-Content-Type-Options: nosniff field in the HTTP response?");
$outPut->assign("Status","Yes");
$outPut->assign("alert","alert-success");
$outPut->assign("Score","1");
}
else{
$score=$score-1;
$outPut->newBlock("Item");
$outPut->assign("Item","Does the server implement X-Content-Type-Options: nosniff field in the HTTP response?");
$outPut->assign("Status","No");
$outPut->assign("alert","alert-danger");
```

```
$outPut->assign("Score","-1");
}
if(isset($xFrameOptFound)){
$outPut->newBlock("Item");
$outPut->assign("Item","Does the server implement X-Frame-Options field in the
HTTP response?");
$outPut->assign("Status","Yes");
$outPut->assign("alert","alert-success");
$outPut->assign("Score","1");
}
else{
$score=$score-1;
$outPut->newBlock("Item");
$outPut->assign("Item","Does the server implement X-Frame-Options field in the
HTTP response?");
$outPut->assign("Status","No");
$outPut->assign("alert","alert-danger");
$outPut->assign("Score","-1");
}

if(isset($cookie)){
$outPut->newBlock("Item");
$outPut->assign("Item","Does the server implement HttpOnly field in the HTTP
response?");
$outPut->assign("Status","Yes");
$outPut->assign("alert","alert-success");
$outPut->assign("Score","1");

}
else{
$score=$score-1;
$outPut->newBlock("Item");
$outPut->assign("Item","Does the server implement HttpOnly field in the HTTP
response?");
$outPut->assign("Status","No");
$outPut->assign("alert","alert-danger");
$outPut->assign("Score","-1");
}



if(isset($cookieSecure)){
$outPut->newBlock("Item");
$outPut->assign("Item","Does the web server flag the cookie values to be secure; only
sending the cookie values via HTTPS?");
$outPut->assign("Status","Yes");
```

```
$outPut->assign("alert","alert-success");
$outPut->assign("Score","1");

}
else{
$score=$score-1;
$outPut->newBlock("Item");
$outPut->assign("Item","Does the web server flag the cookie values to be secure; only
sending the cookie values via HTTPS?");
$outPut->assign("Status","No");
$outPut->assign("alert","alert-danger");

$outPut->assign("Score","-1");
}
if(isset($callpyfaild)){

$score=$score-4;
$outPut->newBlock("Score");
$outPut->assign("totalScore","The Total Score : ".$score ." out of 7");
echo $outPut->getOutputContent();
}
else{
$outPut->newBlock("Score");
$outPut->assign("totalScore","The Total Score : ".$score." out of 11");
echo $outPut->getOutputContent();
}
}
```

call.py

```
import shodan
import sys
import os
import socket
import httplib

url=sys.argv[1]
print "error"
API_KEY = "Here put your key"
IP=socket.gethostbyname(url)
api = shodan.Shodan(API_KEY)
host = api.host(IP)
allPorts=host['ports']

httpMethods = []

#check for TRACE,CONNECT,OPTIONS,DELETE,PUT
notAllowedHttpMethods=['TRACE','CONNECT','OPTIONS','DELETE','PUT']

for x in range(0, 5):
                                        conn = httplib.HTTPConnection(url)
                                        conn.request(notAllowedHttpMethods[x], '/')
                                        response = conn.getresponse()
                                        httpMethods.append(notAllowedHttpMethods[x]+" /
"+str(response.status))

print allPorts,"-",httpMethods
```