CYBER SECURITY AND ETHICAL HACKING

PROJECT TITLE

Scanning using OWASP ZAP

Submitted by: -

Diksha Sharma

Nirma University

B Tech. CSE 2nd year

Abstract

Today, new web applications are made every single day with increasingly more sensitive data to manage. To ensure that no security vulnerabilities such as data leakage in web applications exist, developers are using tools such as a web vulnerability scanner. This type of tool can detect vulnerabilities by automatically finding input fields where data can be injected and performing different attacks on these fields.

One of the most common web vulnerability scanners is *OWASP ZAP*. Web vulnerability scanners were first developed during a time when traditional multi-page applications were prominent. Nowadays, when modern single-page applications have become the de facto standard, new challenges for web vulnerability scanners have arisen. These problems include identifying dynamically updated web pages.

This project aims to evaluate the use of the technique of OWASP ZAP for identifying the vulnerabilities present in a web application. This issue is approached by testing the selected web vulnerability scanner, OWASP ZAP on deliberately vulnerable web applications, to assess the performance and techniques used, and to determine the performance of OWASP ZAP.

In the test scan conducted, it shows the vulnerabilities with their full description consisting of the risks, solutions, the method(get or post) or the urls where the following vulnerability is discovered etc:-.

Objective

1) Scan one web application for testing.
2) Scan using OWASP ZAP Tool (quick/automated).
3) Set of vulnerabilities – make a report with mitigations.
4) Do the following operations:
- Injection
- Broken authentication and session management
- XSS
- IDOR
- Security misconfiguration
- Sensitive data exposure
- CSRF
- Missing functional level access controls
- Using components with known vulnerabilities
- Unvalidated redirects and forwards

Introduction

Zed Attack Proxy (ZAP) is a free, open-source penetration testing tool being maintained under the umbrella of the Open Web Application Security Project (OWASP). ZAP is designed specifically for testing web applications and is both flexible and extensible.

At its core, ZAP is what is known as a “man-in-the-middle proxy.” It stands between the tester’s browser and the web application so that it can intercept and inspect messages sent between browser and web application, modify the contents if needed, and then forward those packets on to the destination. It can be used as a stand-alone application, and as a daemon process.

If there is another network proxy already in use, as in many corporate environments, ZAP can be configured to connect to that proxy.

ZAP provides functionality for a range of skill levels – from developers, to testers new to security testing, to security testing specialists. ZAP has versions for each major OS and Docker, so you are not tied to a single OS. Additional functionality is freely available from a variety of add-ons in the ZAP Marketplace, accessible from within the ZAP client.

Because ZAP is open-source, the source code can be examined to see exactly how the functionality is implemented. Anyone can volunteer to work on ZAP, fix bugs, add features, create pull requests to pull fixes into the project, and author add-ons to support specialized situations.

OWASP ZAP is an ideal tool to use in automation (security testing). It can be run in headless mode and has a powerful API. The *OWASP Zed Attack Proxy (OWASP ZAP)* is an easy-to-use integrated penetration testing tool for finding vulnerabilities in web applications. ZAP passively scans all the requests and responses made during your exploration for vulnerabilities, continues to build the site tree, and records alert for potential vulnerabilities found during the exploration. OWASP ZAP will proceed to crawl the web application with its spider and passively scan each page it finds. Then it will use the active scanner to attack all of the discovered pages, functionality, and parameters.

OWASP ZAP key advantages

- Safe and Secured data handling
- Safeguard our files & folders from external Vulnerabilities and Hacking
- Restrict Malicious File Upload
- Potential ways to mitigate or additional testing that should be done to reduce identified threats
- Focus on areas where your application is most at risk, report back any issues that are found, and provide detailed remediation advice.
- Actively scans the top 14 vulnerability

ZAP can scan through the web application and detect issues related to:

- SQL injection
- Broken Authentication
- Sensitive data exposure
- Broken Access control
- Security misconfiguration
- Cross Site Scripting (XSS)
- Insecure Deserialization
- Components with known vulnerabilities
- Missing security headers

How ZAP attacks works: -

- Once you click the attack button, ZAP starts crawling the web application with its spider, passively scanning each page it finds. Next ZAP uses the active scanner to attack all discovered pages, parameters, and functionality.
- ZAP includes two spiders that can crawl web applications.
- It discovers the risks and shows it accordingly.

Risk Scoring: -

All identified vulnerabilities are classified according to CWE Common Weakness Enumeration (CWE database) Threat Classification (( ) list of common software security weaknesses and WASC database) and likelihood factors between Informational, Low, Medium and High risk scores, and then ranked based on the potential impact.

The assigned risk scoring, whilst provided for informative purposes, could be used to determine the severity and urgency of the reported vulnerabilities.

Risk Score	Description
Informational (INFO)	Informational level issues provide additional insights about your application security features and configuration.
Low (LOW)	Low level issues might not not individually constitute a critical risk these can be used for more advanced attacks when used with each other.
Medium (MEDIUM)	Medium level issues can lead to a very high risk when combined together, these should be resolved with a priority.
High (HIGH)	High level issues individually pose a very high risk and can lead to a very serious impact on data integrity and confidentiality or the availability of the overall application.

Alert Definition: -

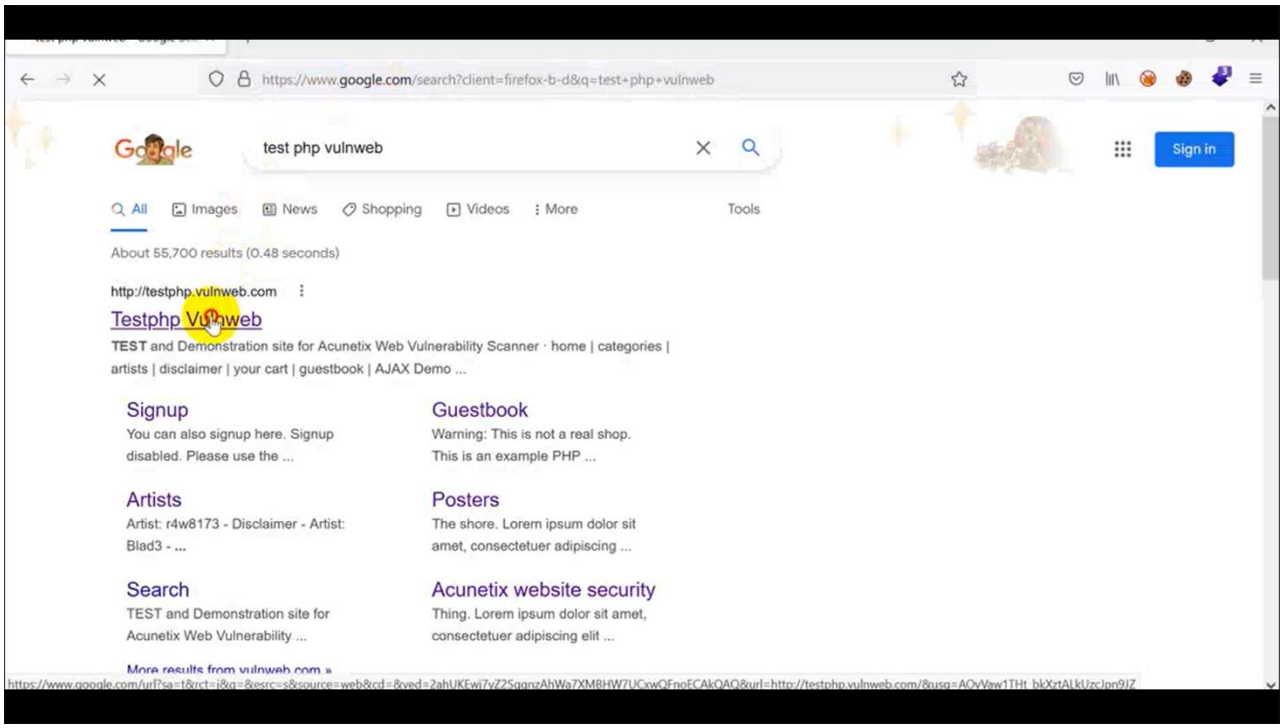
The security issues discovered in the scanned web application are reported in detail as security alerts with the following attributes:

Risk Score (XX)	Estimated risk score of the identified issue with the number of occurrences
Alert Name	Vulnerability name
Description	Vulnerability description
Instances	Number of occurrences found
URL / Method	URL and request method (GET/POST) of the vulnerable location
Solution	Possible solution / mitigation
Other Information	Any supplementary information documenting the issue discovery
Reference	External references describing the problem

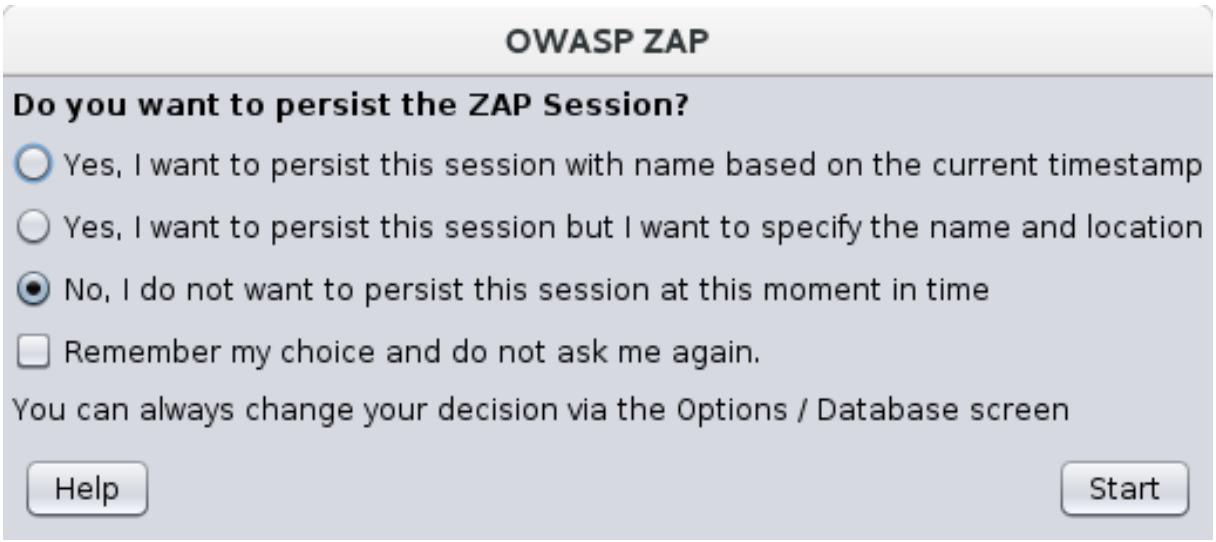
Methodology

1. *Selecting an application*

First step is selecting any one particular application for doing these particular attacks mentioned in the objectives sections. I selected testphp.vulnweb.com.

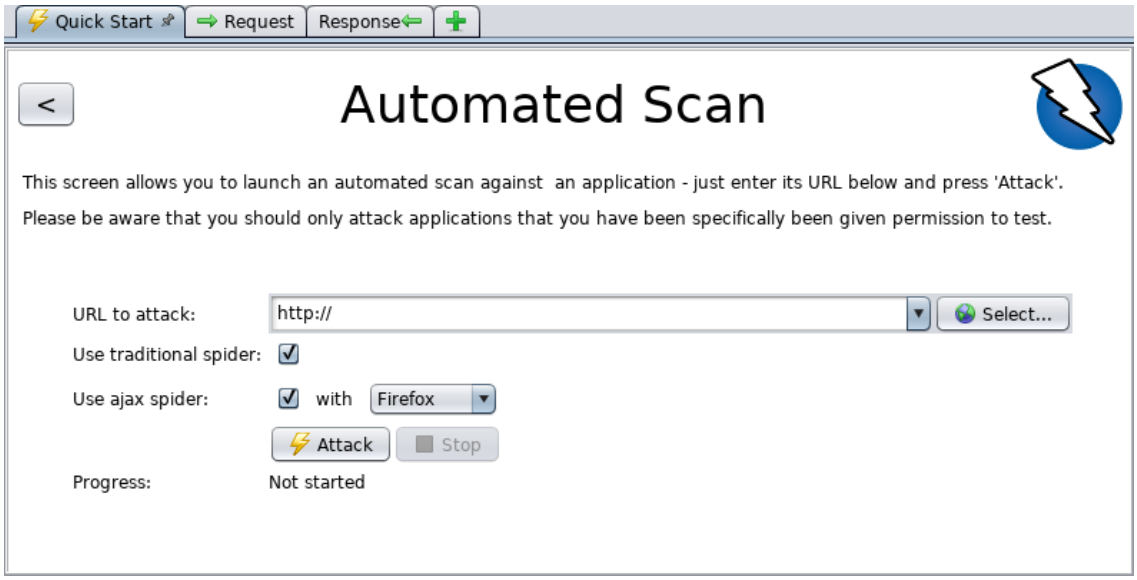


2. *Open the ZAP app*



3. *Run an automated scan*

Select the url of web application and paste it in url to attack box and click on url. Then Click on automated Scan and it will check all the vulnerabilities present in the web application.

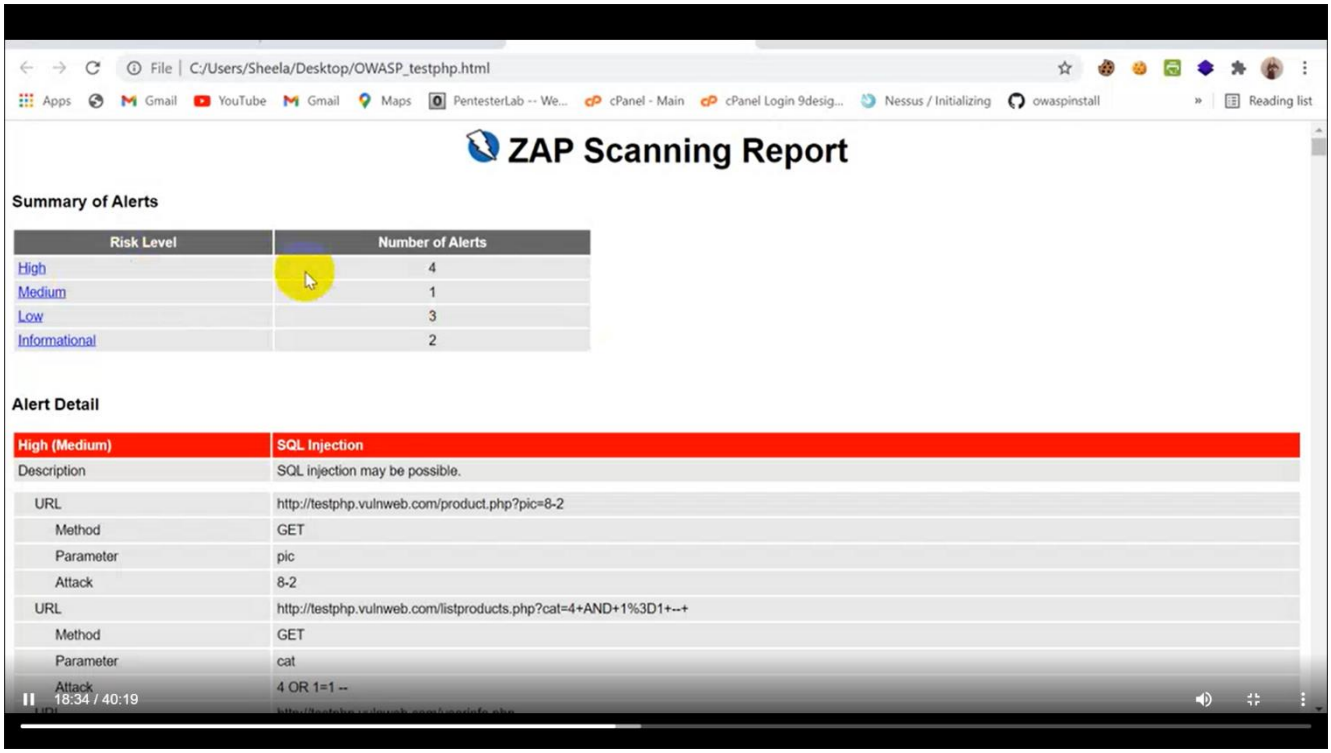


In the alert section, it's gonna generate a very clear detailed information about the vulnerabilities including where they are present and their risks and solution.

In the report, each alert is explained in detail- it's risks, the no. of alerts etc:-

4. *Generate the report*

After the full scanning, we generate a report in the format we require. In this project, I have generated the report in the html format.



5. *Consolidate the report.*

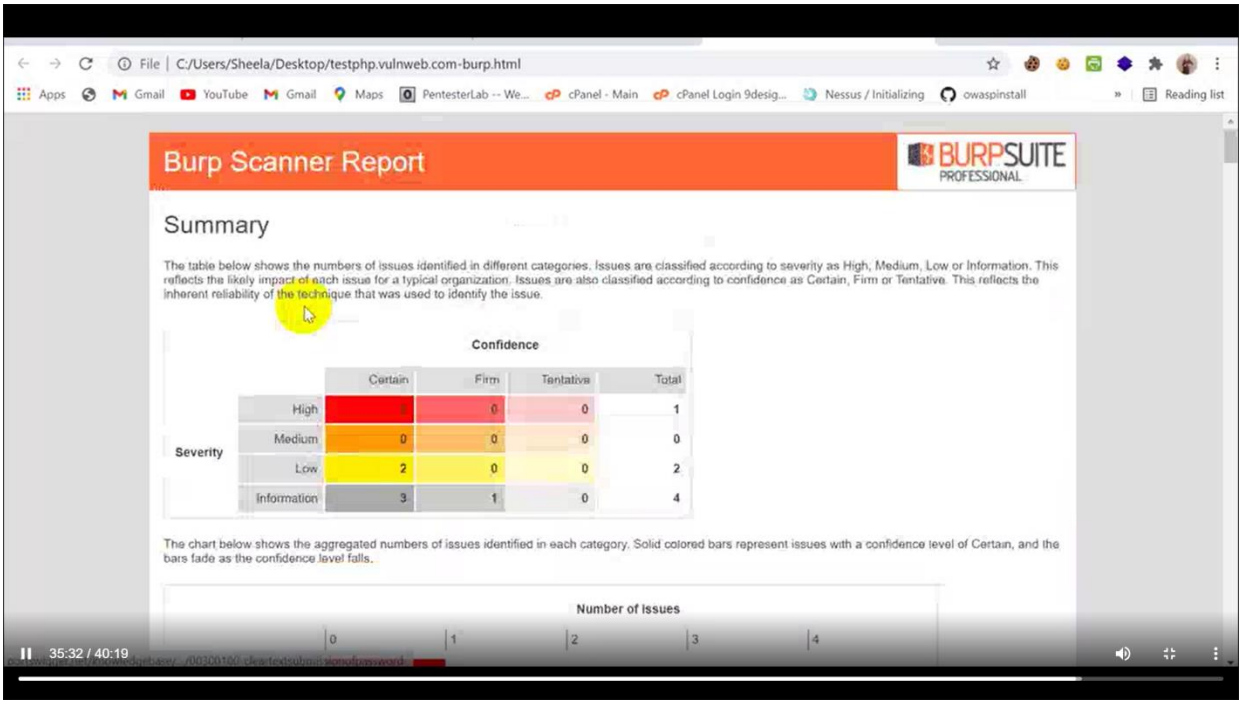
To consolidate this report, we will check the vulnerabilities manually to see whether these vulnerabilities exist or not.

6. *Manual Scanning*

For manual scanning, I used burpsite and checked all the vulnerabilities required manually in the selected website. Make sure to check for all the injections like sql etc: -.

7. *Generate the report*

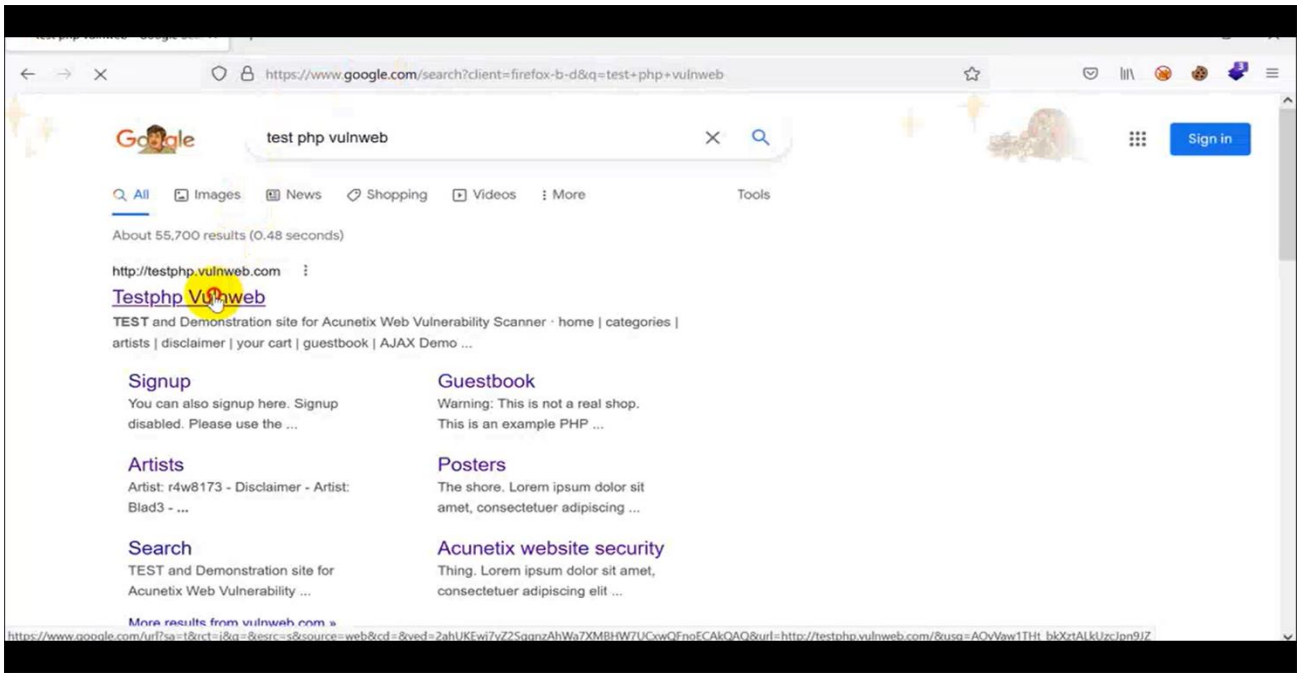
Then after complete scanning, generate the report from burpsite. It will also contain the detailed information regarding each and every vulnerability.

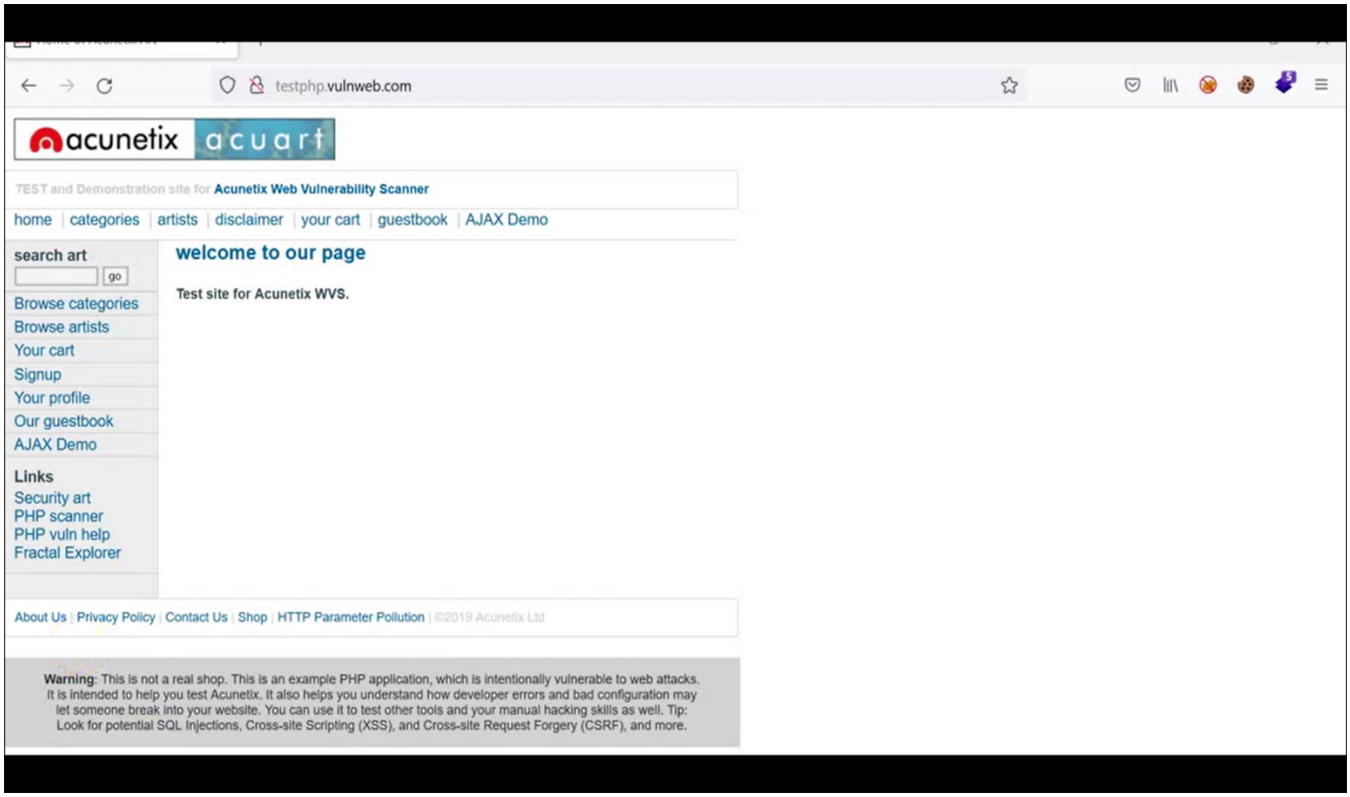


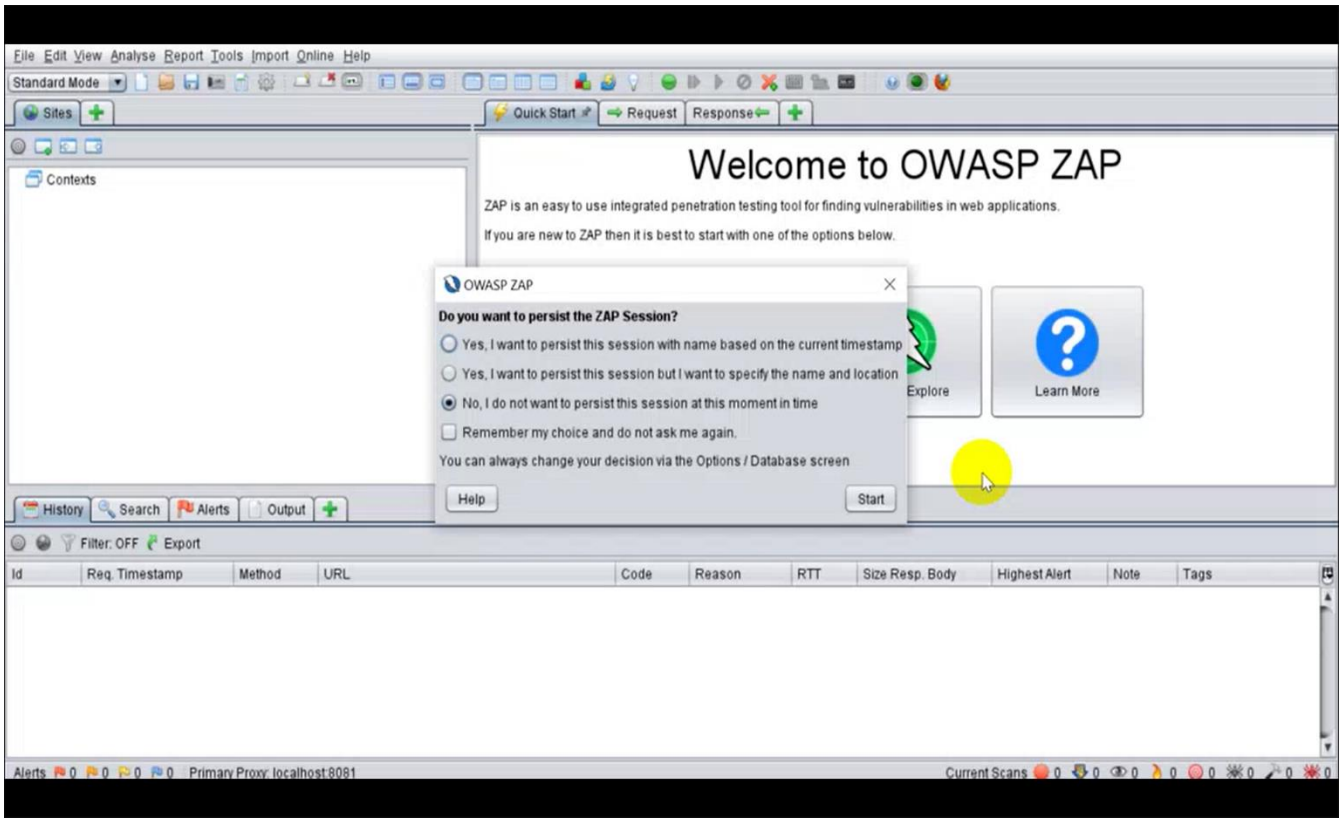
8. *Compare the reports*

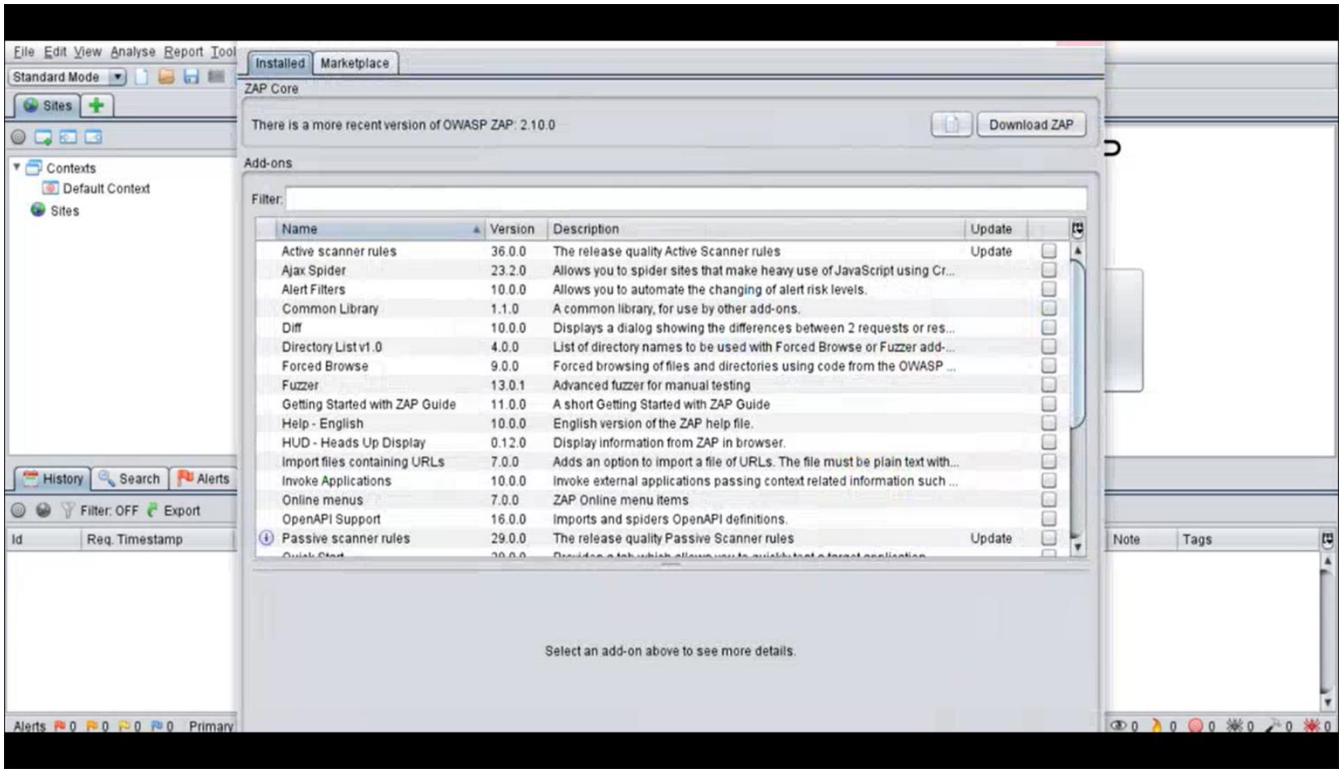
Compare both the reports generated from the sites burpsite and OWASP ZAP after manual scanning and automated scanning respectively.

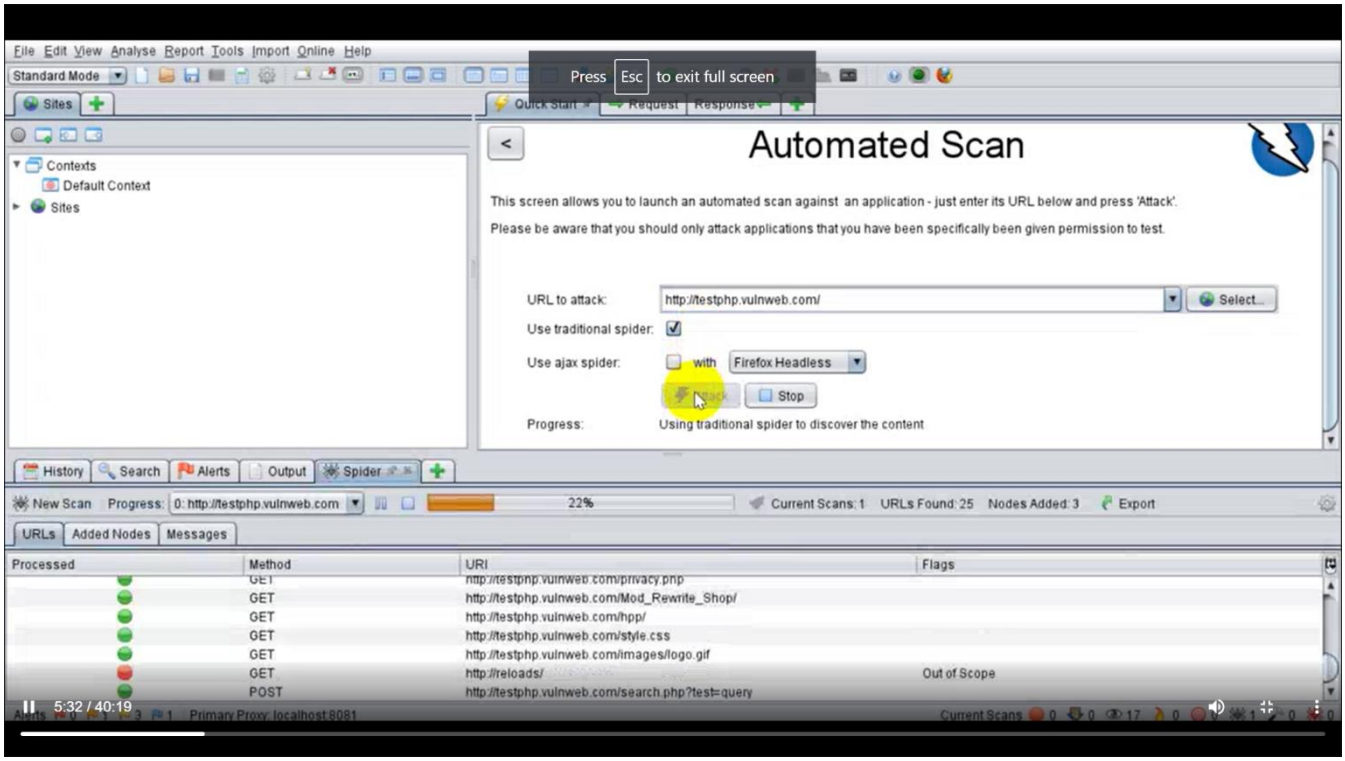
Screenshots:

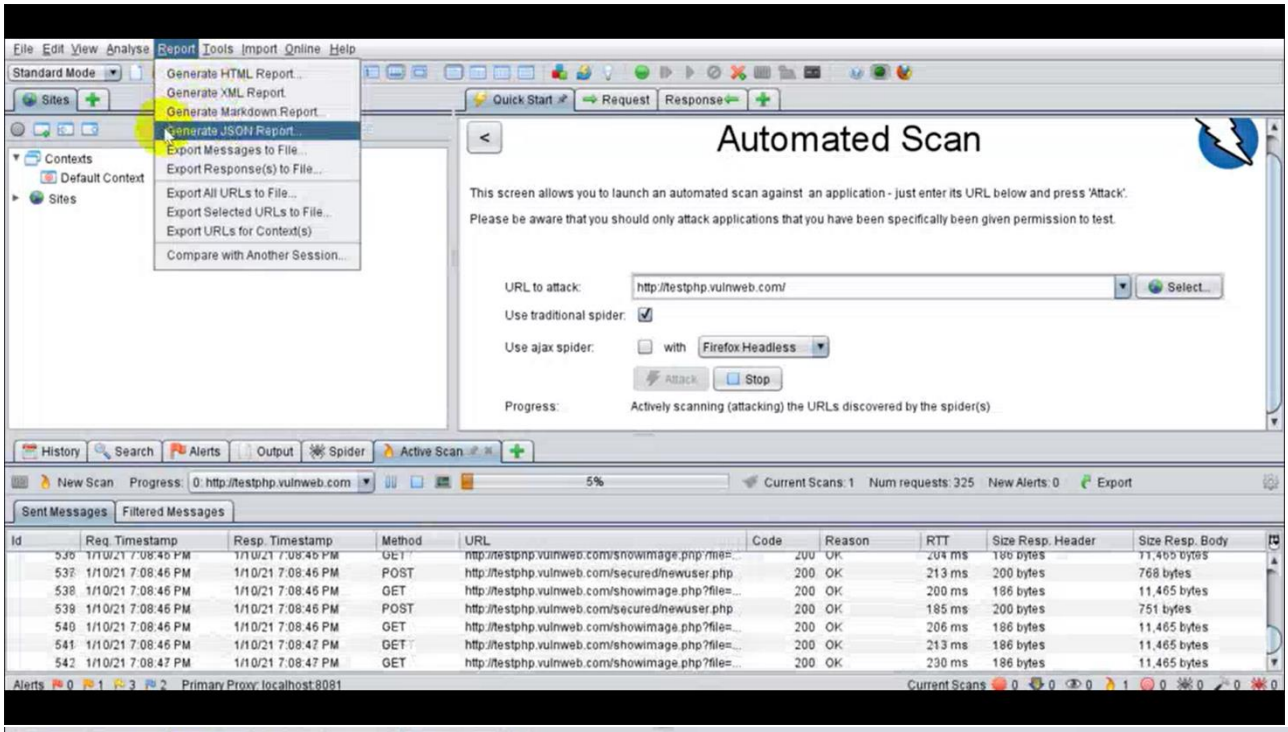


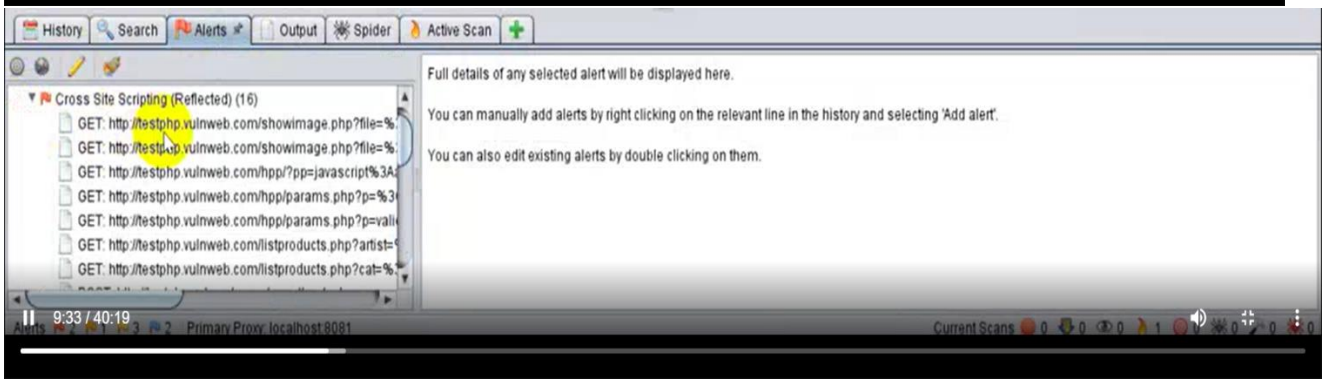

testphp.vulnweb.com
acunetix acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo
search art
go
Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo
Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer
welcome to our page
Test site for Acunetix WVS.
About Us | Privacy Policy | Contact Us | Shop | HTTP Parameter Pollution | ©2019 Acunetix Ltd
Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

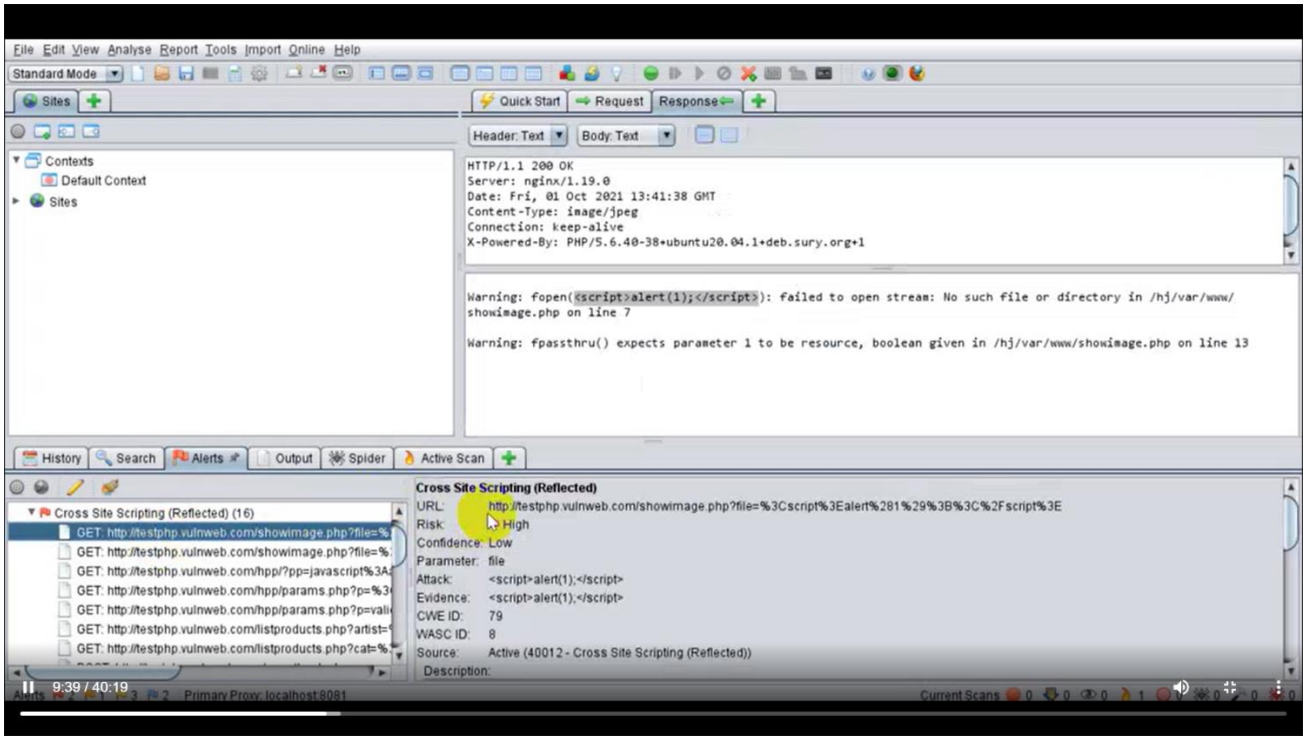

File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Contexts
Welcome to OWASP ZAP
ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.
If you are new to ZAP then it is best to start with one of the options below.
Explore
Learn More
OWASP ZAP
Do you want to persist the ZAP Session?
Yes, I want to persist this session with name based on the current timestamp
Yes, I want to persist this session but I want to specify the name and location
No, I do not want to persist this session at this moment in time
Remember my choice and do not ask me again.
You can always change your decision via the Options / Database screen
Help
Start
History
Search
Alerts
Output
Filter: OFF
Export
Id
Req. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Body
Highest Alert
Note
Tags
Alerts
Primary Proxy: localhost:8081
Current Scans

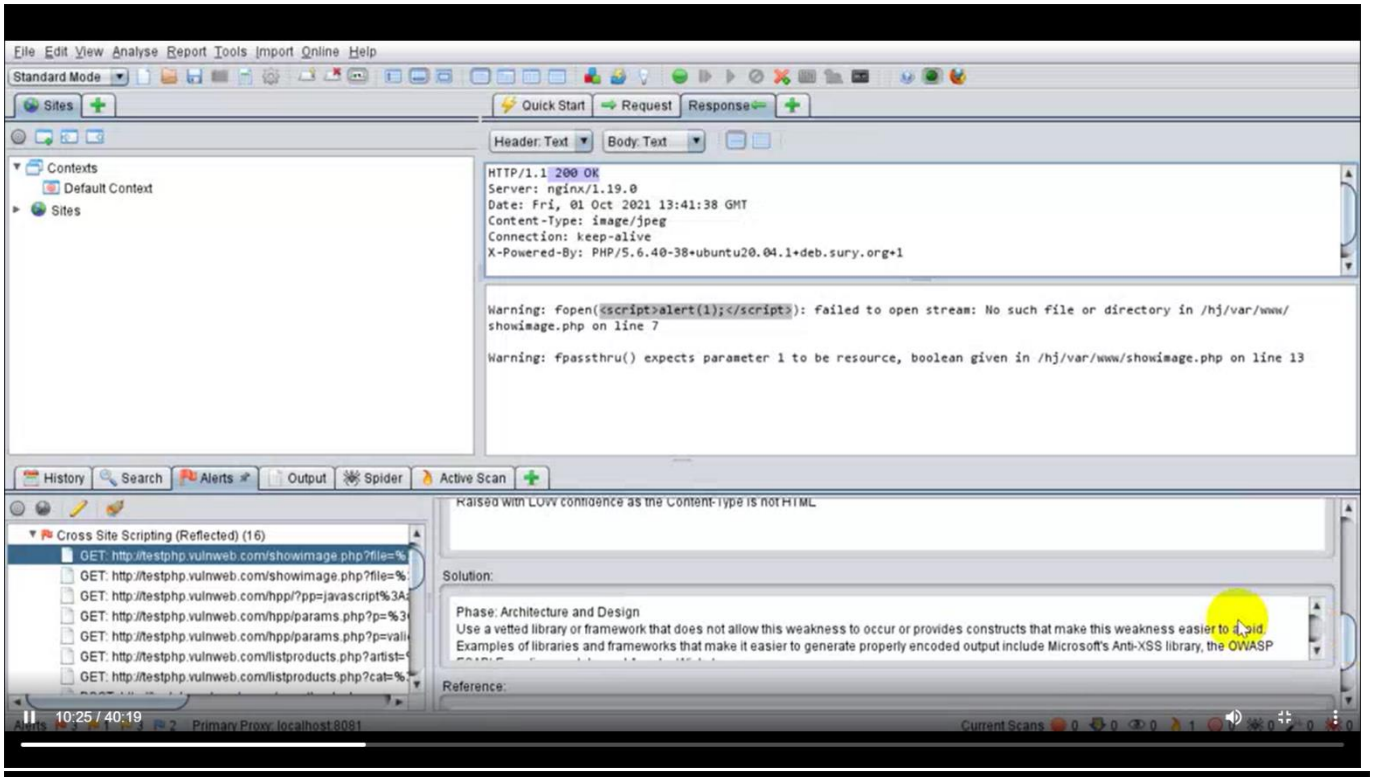

File Edit View Analyse Report Tool
Standard Mode
Sites
Contexts
Default Context
Sites
Installed
Marketplace
ZAP Core
There is a more recent version of OWASP ZAP: 2.10.0
Download ZAP
Add-ons
Filter:
Name
Version
Description
Update
Active scanner rules 36.0.0 The release quality Active Scanner rules Update
Ajax Spider 23.2.0 Allows you to spider sites that make heavy use of JavaScript using Cr...
Alert Filters 10.0.0 Allows you to automate the changing of alert risk levels.
Common Library 1.1.0 A common library, for use by other add-ons.
Diff 10.0.0 Displays a dialog showing the differences between 2 requests or res...
Directory List v1.0 4.0.0 List of directory names to be used with Forced Browse or Fuzzer add-...
Forced Browse 9.0.0 Forced browsing of files and directories using code from the OWASP ...
Fuzzer 13.0.1 Advanced fuzzer for manual testing
Getting Started with ZAP Guide 11.0.0 A short Getting Started with ZAP Guide
Help - English 10.0.0 English version of the ZAP help file.
HUD - Heads Up Display 0.12.0 Display information from ZAP in browser.
Import files containing URLs 7.0.0 Adds an option to import a file of URLs. The file must be plain text with...
Invoke Applications 10.0.0 Invoke external applications passing context related information such ...
Online menus 7.0.0 ZAP Online menu items
OpenAPI Support 16.0.0 Imports and spiders OpenAPI definitions.
Passive scanner rules 29.0.0 The release quality Passive Scanner rules Update
Select an add-on above to see more details.
History
Search
Alerts
Filter: OFF
Export
Id
Req. Timestamp
Note
Tags
Alerts 0 0 0 0 Primary

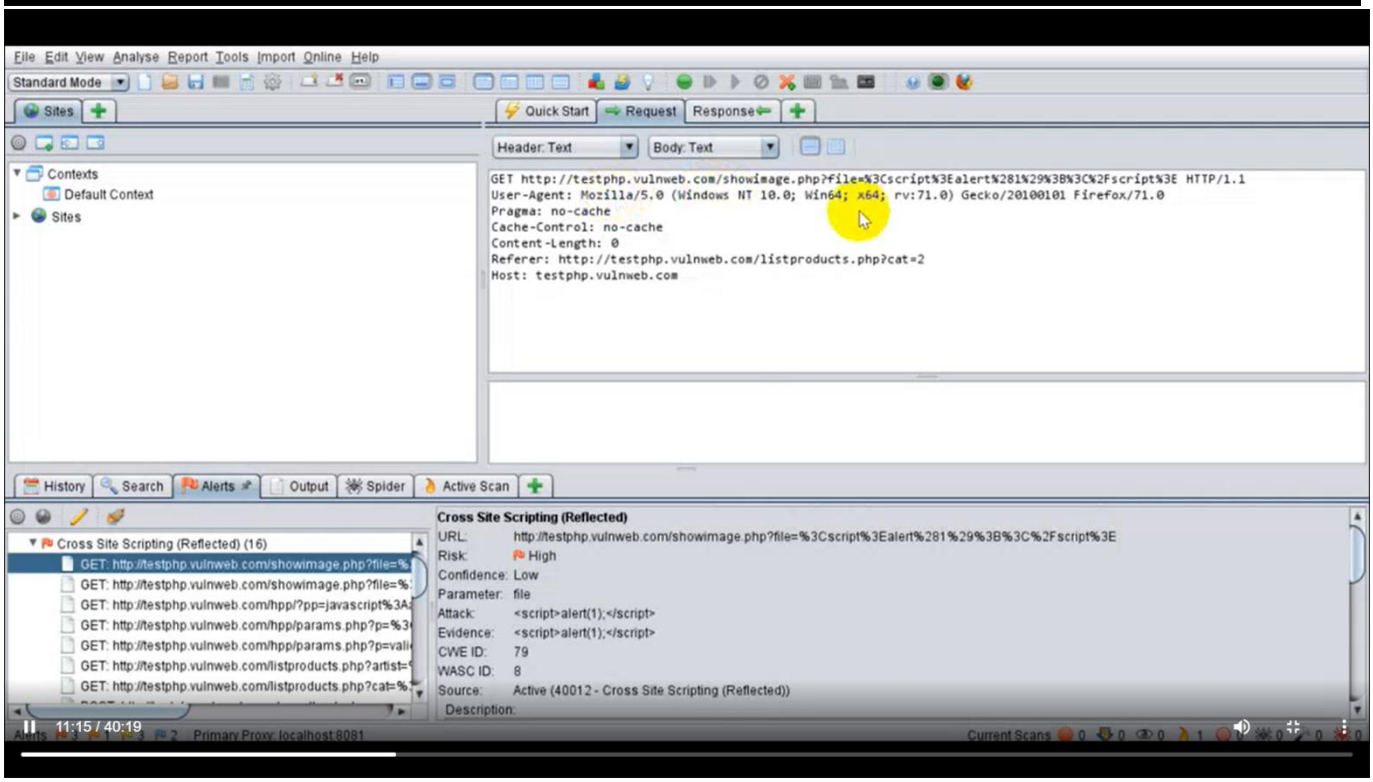

File Edit View Analyse Report Tools Import Online Help
Press Esc to exit full screen
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
Automated Scan
This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.
URL to attack: http://testphp.vulnweb.com/
Select...
Use traditional spider:
Use ajax spider: with Firefox Headless
Attack
Stop
Progress: Using traditional spider to discover the content
History
Search
Alerts
Output
Spider
New Scan
Progress: 0: http://testphp.vulnweb.com
22%
Current Scans: 1
URLs Found: 25
Nodes Added: 3
Export
URLs
Added Nodes
Messages
Processed
Method
URI
Flags
GET http://testphp.vulnweb.com/privacy.php
GET http://testphp.vulnweb.com/Mod_Rewrite_Shop/
GET http://testphp.vulnweb.com/hpp/
GET http://testphp.vulnweb.com/style.css
GET http://testphp.vulnweb.com/images/logo.gif
GET http://reloads/ Out of Scope
POST http://testphp.vulnweb.com/search.php?test=query
5:32 / 40:19
Primary Proxy: localhost:8081
Current Scans

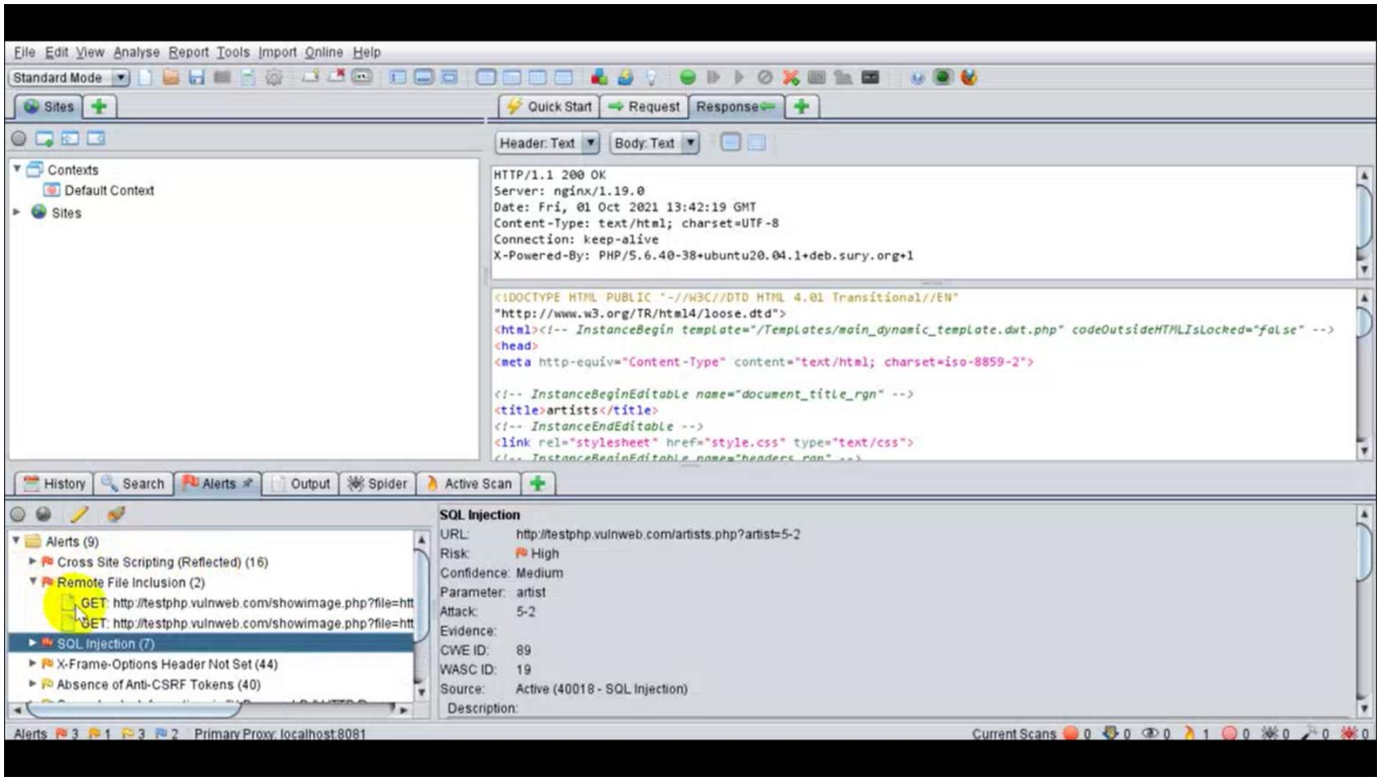

File Edit View Analyse Report Tools Import Online Help
Generate HTML Report...
Generate XML Report...
Generate Markdown Report...
Generate JSON Report...
Export Messages to File...
Export Response(s) to File...
Export All URLs to File...
Export Selected URLs to File...
Export URLs for Context(s)...
Compare with Another Session...
Automated Scan
This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.
URL to attack: http://testphp.vulnweb.com/
Use traditional spider:
Use ajax spider: with Firefox Headless
Attack
Stop
Progress: Actively scanning (attacking) the URLs discovered by the spider(s)
New Scan Progress: 0: http://testphp.vulnweb.com 5%
Current Scans: 1 Num requests: 325 New Alerts: 0 Export
Sent Messages Filtered Messages
Id Req. Timestamp Resp. Timestamp Method URL Code Reason RTT Size Resp. Header Size Resp. Body
537 1/10/21 7:08:46 PM 1/10/21 7:08:46 PM POST http://testphp.vulnweb.com/secured/newuser.php 200 OK 213 ms 200 bytes 768 bytes
538 1/10/21 7:08:46 PM 1/10/21 7:08:46 PM GET http://testphp.vulnweb.com/showimage.php?file=... 200 OK 200 ms 186 bytes 11,465 bytes
539 1/10/21 7:08:46 PM 1/10/21 7:08:46 PM POST http://testphp.vulnweb.com/secured/newuser.php 200 OK 185 ms 200 bytes 751 bytes
540 1/10/21 7:08:46 PM 1/10/21 7:08:46 PM GET http://testphp.vulnweb.com/showimage.php?file=... 200 OK 206 ms 186 bytes 11,465 bytes
541 1/10/21 7:08:46 PM 1/10/21 7:08:47 PM GET http://testphp.vulnweb.com/showimage.php?file=... 200 OK 213 ms 186 bytes 11,465 bytes
542 1/10/21 7:08:47 PM 1/10/21 7:08:47 PM GET http://testphp.vulnweb.com/showimage.php?file=... 200 OK 230 ms 186 bytes 11,465 bytes
Alerts Primary Proxy: localhost:8081 Current Scans

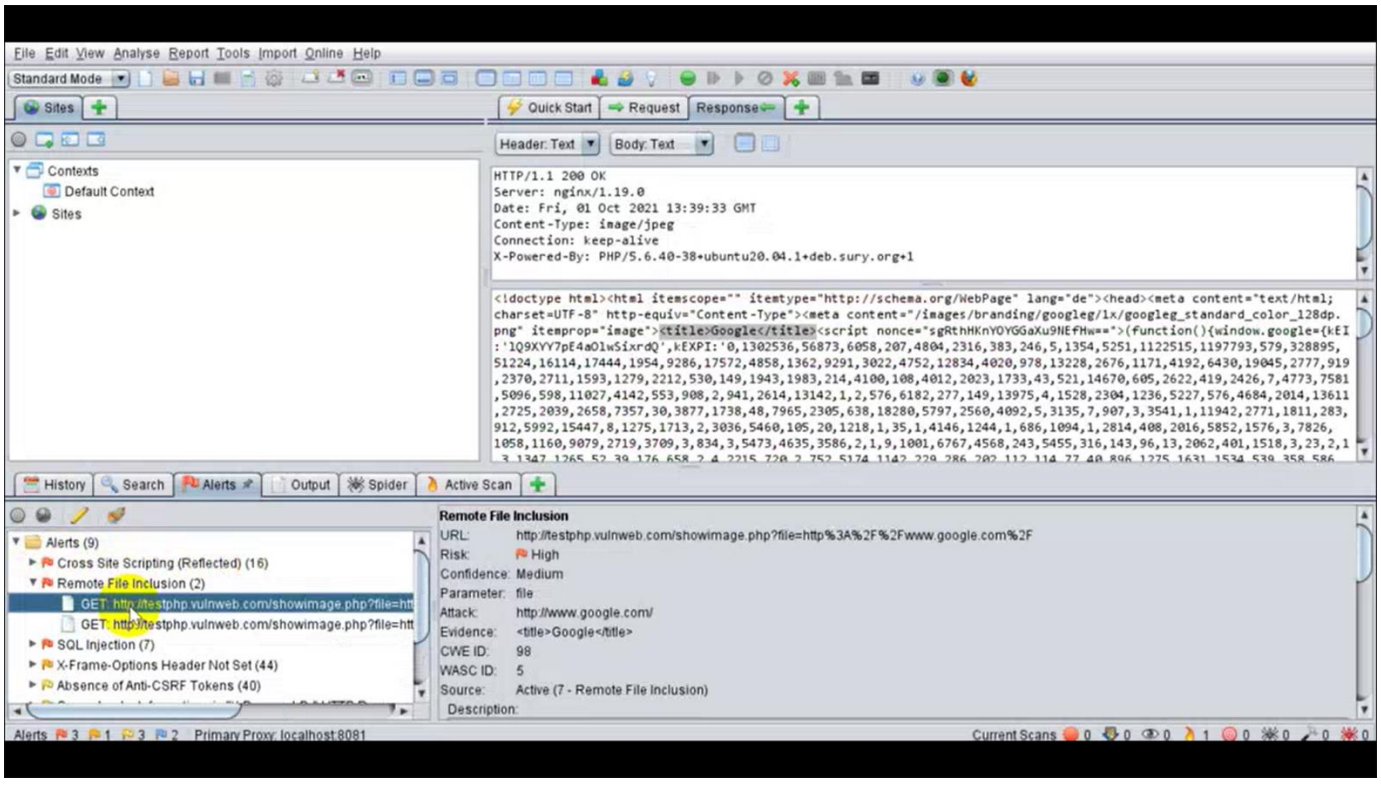

History Search Alerts Output Spider Active Scan
Cross Site Scripting (Reflected) (16)
GET: http://testphp.vulnweb.com/showimage.php?file=%
GET: http://testphp.vulnweb.com/showimage.php?file=%
GET: http://testphp.vulnweb.com/hpp/?pp=javascript%3A
GET: http://testphp.vulnweb.com/hpp/params.php?p=%3
GET: http://testphp.vulnweb.com/hpp/params.php?p=vali
GET: http://testphp.vulnweb.com/listproducts.php?artist=
GET: http://testphp.vulnweb.com/listproducts.php?cat=%
Full details of any selected alert will be displayed here.
You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.
You can also edit existing alerts by double clicking on them.
9:33 / 40:19
Alerts Primary Proxy: localhost:8081 Current Scans

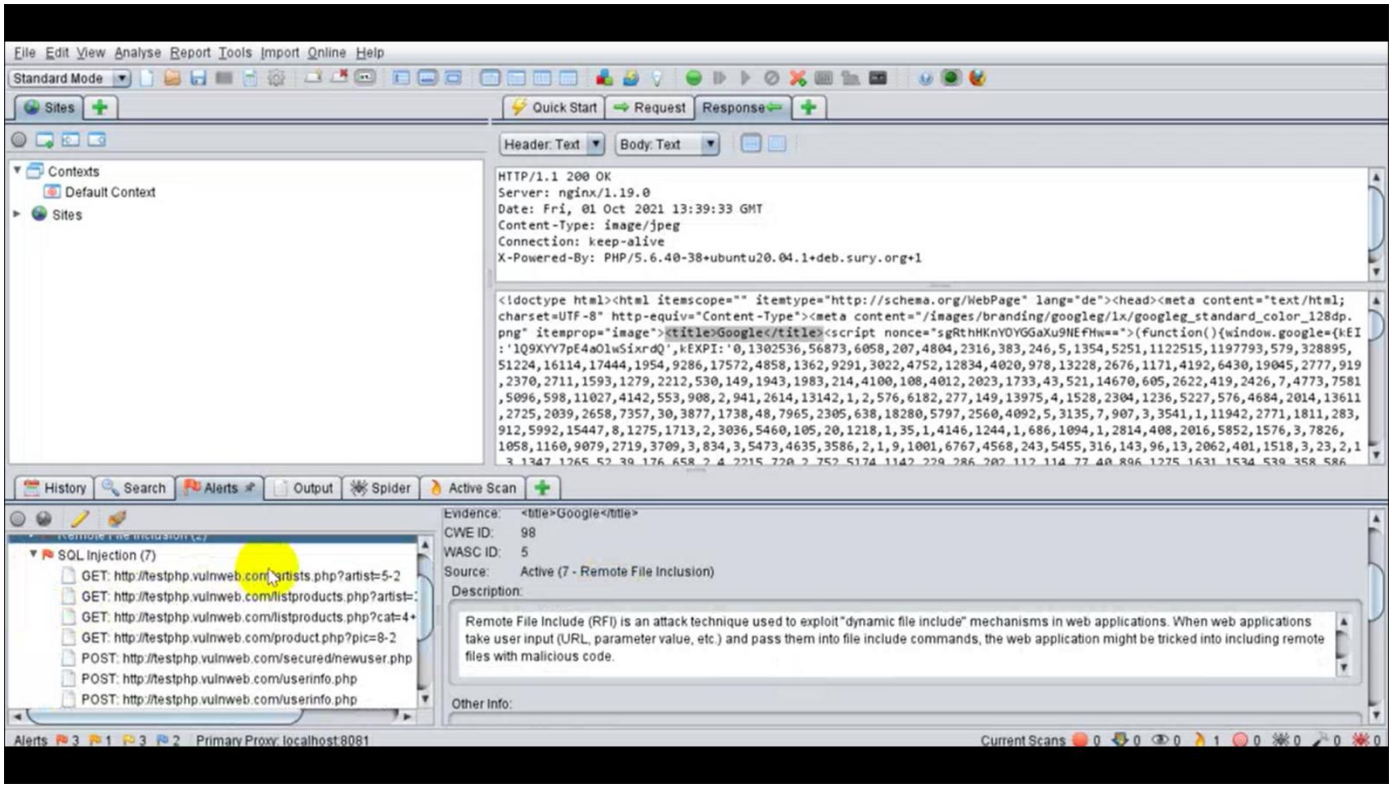

File Edit View Analyse Report Tools Import Online Help
Quick Start Request Response
Header: Text Body: Text
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Fri, 01 Oct 2021 13:41:38 GMT
Content-Type: image/jpeg
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Warning: fopen(<script>alert(1);</script>): failed to open stream: No such file or directory in /hj/var/www/showimage.php on line 7
Warning: fpassthru() expects parameter 1 to be resource, boolean given in /hj/var/www/showimage.php on line 13
Cross Site Scripting (Reflected)
URL: http://testphp.vulnweb.com/showimage.php?file=%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Risk: High
Confidence: Low
Parameter: file
Attack: <script>alert(1);</script>
Evidence: <script>alert(1);</script>
CWE ID: 79
WASC ID: 8
Source: Active (40012 - Cross Site Scripting (Reflected))
Description:
9:39 / 40:19

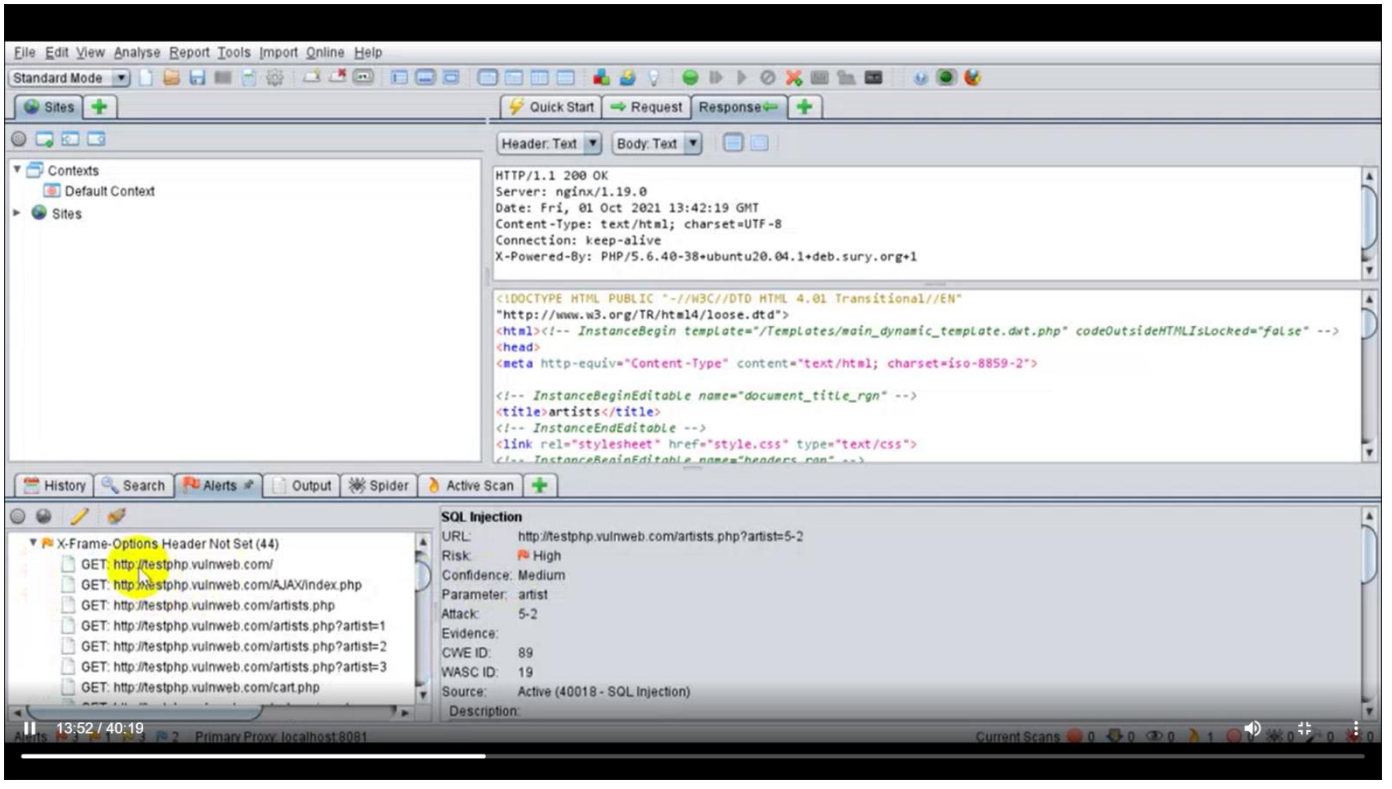

File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Fri, 01 Oct 2021 13:41:38 GMT
Content-Type: image/jpeg
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Warning: fopen(<script>alert(1);</script>): failed to open stream: No such file or directory in /hj/var/www/showimage.php on line 7
Warning: fpassthru() expects parameter 1 to be resource, boolean given in /hj/var/www/showimage.php on line 13
History
Search
Alerts
Output
Spider
Active Scan
Cross Site Scripting (Reflected) (16)
GET: http://testphp.vulnweb.com/showimage.php?file=%
Solution:
Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP
Reference:
10:25 / 40:19
Primary Proxy: localhost:8081
Current Scans

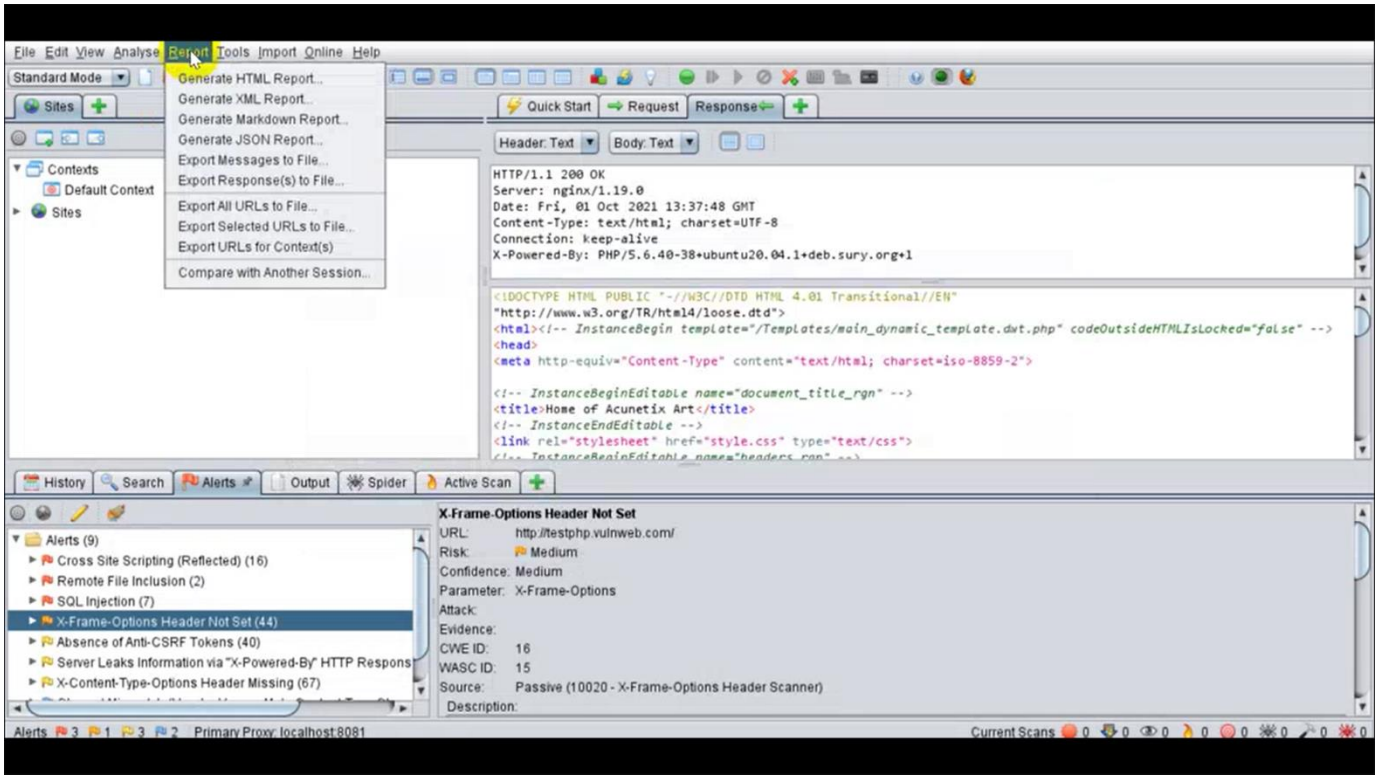

File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
GET http://testphp.vulnweb.com/showimage.php?file=%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0
Pragma: no-cache
Cache-Control: no-cache
Content-Length: 0
Referer: http://testphp.vulnweb.com/listproducts.php?cat=2
Host: testphp.vulnweb.com
History
Search
Alerts
Output
Spider
Active Scan
Cross Site Scripting (Reflected) (16)
GET: http://testphp.vulnweb.com/showimage.php?file=%
Cross Site Scripting (Reflected)
URL: http://testphp.vulnweb.com/showimage.php?file=%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
Risk: High
Confidence: Low
Parameter: file
Attack: <script>alert(1);</script>
Evidence: <script>alert(1);</script>
CWE ID: 79
WASC ID: 8
Source: Active (40012 - Cross Site Scripting (Reflected))
Description:
11:15 / 40:19
Primary Proxy: localhost:8081
Current Scans

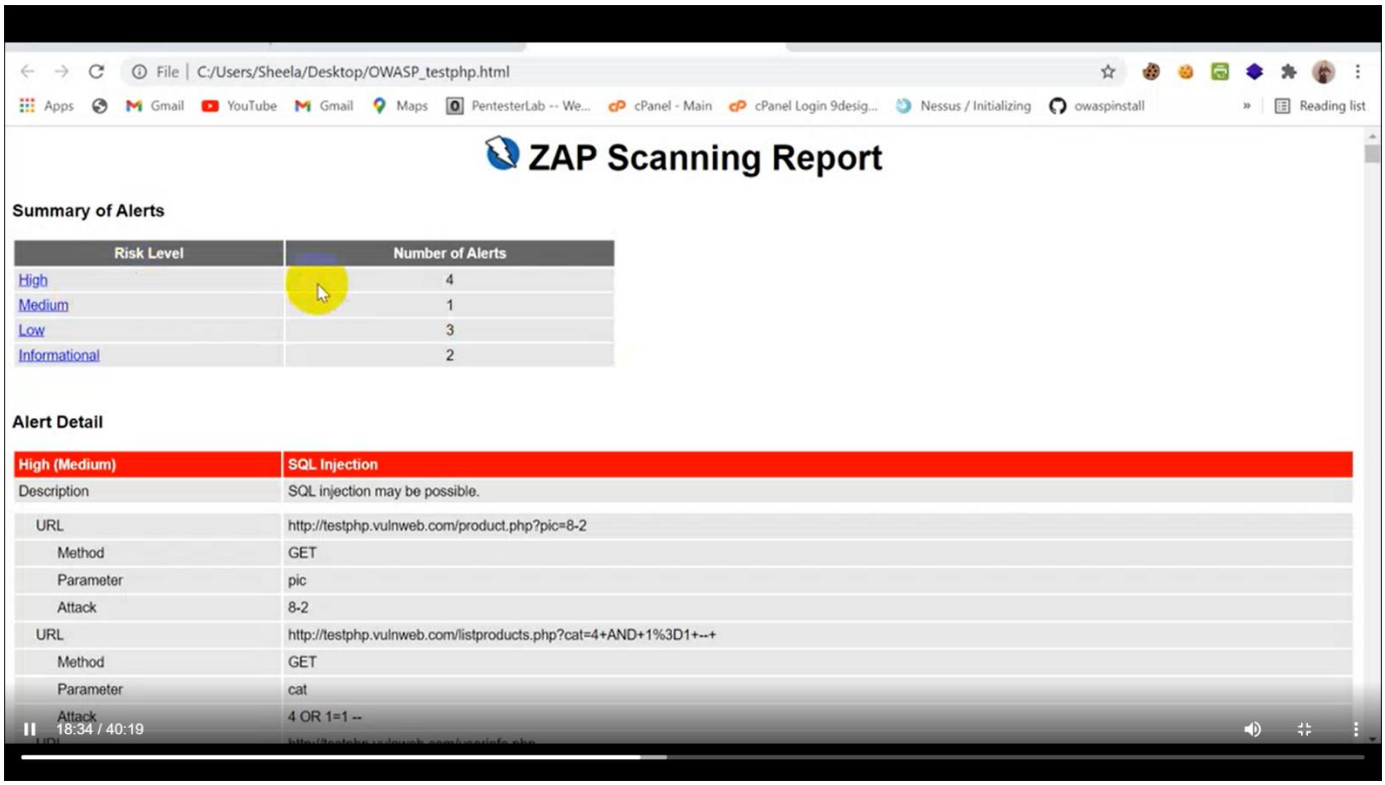

File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Fri, 01 Oct 2021 13:42:19 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" codeOutsideHTMLIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<!-- InstanceBeginEditable name="document_title_rgn" -->
<title>artists</title>
<!-- InstanceEndEditable -->
<link rel="stylesheet" href="style.css" type="text/css">
History
Search
Alerts
Output
Spider
Active Scan
Alerts (9)
Cross Site Scripting (Reflected) (16)
Remote File Inclusion (2)
GET: http://testphp.vulnweb.com/showimage.php?file=htt
GET: http://testphp.vulnweb.com/showimage.php?file=htt
SQL Injection (7)
X-Frame-Options Header Not Set (44)
Absence of Anti-CSRF Tokens (40)
SQL Injection
URL: http://testphp.vulnweb.com/artists.php?artist=5-2
Risk: High
Confidence: Medium
Parameter: artist
Attack: 5-2
Evidence:
CWE ID: 89
WASC ID: 19
Source: Active (40018 - SQL Injection)
Description:
Alerts 3 1 3 2 Primary Proxy: localhost:8081
Current Scans 0 0 0 1 0 0 0 0

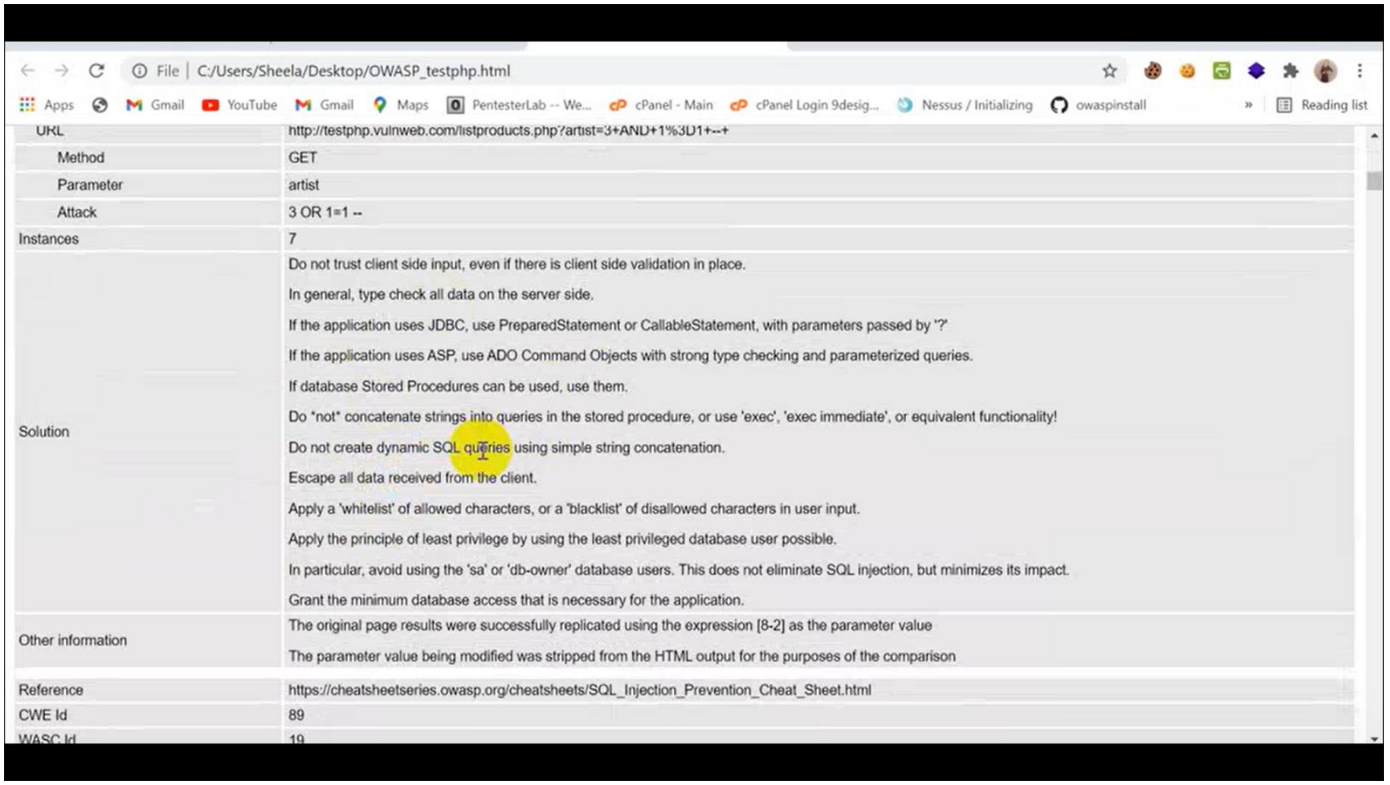

File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Fri, 01 Oct 2021 13:39:33 GMT
Content-Type: image/jpeg
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
<idoctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="de"><head><meta content="text/html;
charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.
png" itemprop="image"><title>Google</title><script nonce="sgRthHKnYOYGGaXu9NEfHw==">(function(){window.google={kEI
:'lQ9XYY7pE4aOlwSixrdQ',kEXPI:'0,1302536,56873,6058,207,4804,2316,383,246,5,1354,5251,1122515,1197793,579,328895,
51224,16114,17444,1954,9286,17572,4858,1362,9291,3022,4752,12834,4020,978,13228,2676,1171,4192,6430,19045,2777,919
,2370,2711,1593,1279,2212,530,149,1943,1983,214,4100,108,4012,2023,1733,43,521,14670,605,2622,419,2426,7,4773,7581
,5096,598,11027,4142,553,908,2,941,2614,13142,1,2,576,6182,277,149,13975,4,1528,2304,1236,5227,576,4684,2014,13611
,2725,2039,2658,7357,30,3877,1738,48,7965,2305,638,18280,5797,2560,4092,5,3135,7,907,3,3541,1,11942,2771,1811,283,
912,5992,15447,8,1275,1713,2,3036,5460,105,20,1218,1,35,1,4146,1244,1,686,1094,1,2814,408,2016,5852,1576,3,7826,
1058,1160,9079,2719,3709,3,834,3,5473,4635,3586,2,1,9,1001,6767,4568,243,5455,316,143,96,13,2062,401,1518,3,23,2,1
History
Search
Alerts
Output
Spider
Active Scan
Alerts (9)
Cross Site Scripting (Reflected) (16)
Remote File Inclusion (2)
GET: http://testphp.vulnweb.com/showimage.php?file=htt
GET: http://testphp.vulnweb.com/showimage.php?file=htt
SQL Injection (7)
X-Frame-Options Header Not Set (44)
Absence of Anti-CSRF Tokens (40)
Remote File Inclusion
URL: http://testphp.vulnweb.com/showimage.php?file=http%3A%2F%2Fwww.google.com%2F
Risk: High
Confidence: Medium
Parameter: file
Attack: http://www.google.com/
Evidence: <title>Google</title>
CWE ID: 98
WASC ID: 5
Source: Active (7 - Remote File Inclusion)
Description:
Alerts 3 1 3 2 Primary Proxy: localhost:8081
Current Scans 0 0 0 1 0 0 0 0


File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Fri, 01 Oct 2021 13:39:33 GMT
Content-Type: image/jpeg
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
History
Search
Alerts
Output
Spider
Active Scan
SQL Injection (7)
GET: http://testphp.vulnweb.com/artists.php?artist=5-2
GET: http://testphp.vulnweb.com/listproducts.php?artist=
GET: http://testphp.vulnweb.com/listproducts.php?cat=4
GET: http://testphp.vulnweb.com/product.php?pic=8-2
POST: http://testphp.vulnweb.com/secured/newuser.php
POST: http://testphp.vulnweb.com/userinfo.php
POST: http://testphp.vulnweb.com/userinfo.php
Evidence: <title>Google</title>
CWE ID: 98
WASC ID: 5
Source: Active (7 - Remote File Inclusion)
Description:
Remote File Include (RFI) is an attack technique used to exploit "dynamic file include" mechanisms in web applications. When web applications take user input (URL, parameter value, etc.) and pass them into file include commands, the web application might be tricked into including remote files with malicious code.
Other Info:
Alerts
Primary Proxy: localhost:8081
Current Scans


File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Fri, 01 Oct 2021 13:42:19 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" codeOutsideHTMLIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<!-- InstanceBeginEditable name="document_title_rgn" -->
<title>artists</title>
<!-- InstanceEndEditable -->
<link rel="stylesheet" href="style.css" type="text/css">
History
Search
Alerts
Output
Spider
Active Scan
X-Frame-Options Header Not Set (44)
GET: http://testphp.vulnweb.com/
GET: http://testphp.vulnweb.com/AJAX/index.php
GET: http://testphp.vulnweb.com/artists.php
GET: http://testphp.vulnweb.com/artists.php?artist=1
GET: http://testphp.vulnweb.com/artists.php?artist=2
GET: http://testphp.vulnweb.com/artists.php?artist=3
GET: http://testphp.vulnweb.com/cart.php
SQL Injection
URL: http://testphp.vulnweb.com/artists.php?artist=5-2
Risk: High
Confidence: Medium
Parameter: artist
Attack: 5-2
Evidence:
CWE ID: 89
WASC ID: 19
Source: Active (40018 - SQL Injection)
Description:
13:52 / 40:19
Primary Proxy: localhost:8081
Current Scans


File Edit View Analyse Report Tools Import Online Help
Generate HTML Report...
Generate XML Report...
Generate Markdown Report...
Generate JSON Report...
Export Messages to File...
Export Response(s) to File...
Export All URLs to File...
Export Selected URLs to File...
Export URLs for Context(s)
Compare with Another Session...
Quick Start
Request
Response
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Fri, 01 Oct 2021 13:37:48 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
History
Search
Alerts
Output
Spider
Active Scan
Alerts (9)
Cross Site Scripting (Reflected) (16)
Remote File Inclusion (2)
SQL Injection (7)
X-Frame-Options Header Not Set (44)
Absence of Anti-CSRF Tokens (40)
Server Leaks Information via "X-Powered-By" HTTP Respons
X-Content-Type-Options Header Missing (67)
X-Frame-Options Header Not Set
URL: http://testphp.vulnweb.com/
Risk: Medium
Confidence: Medium
Parameter: X-Frame-Options
Attack:
Evidence:
CWE ID: 16
WASC ID: 15
Source: Passive (10020 - X-Frame-Options Header Scanner)
Description:

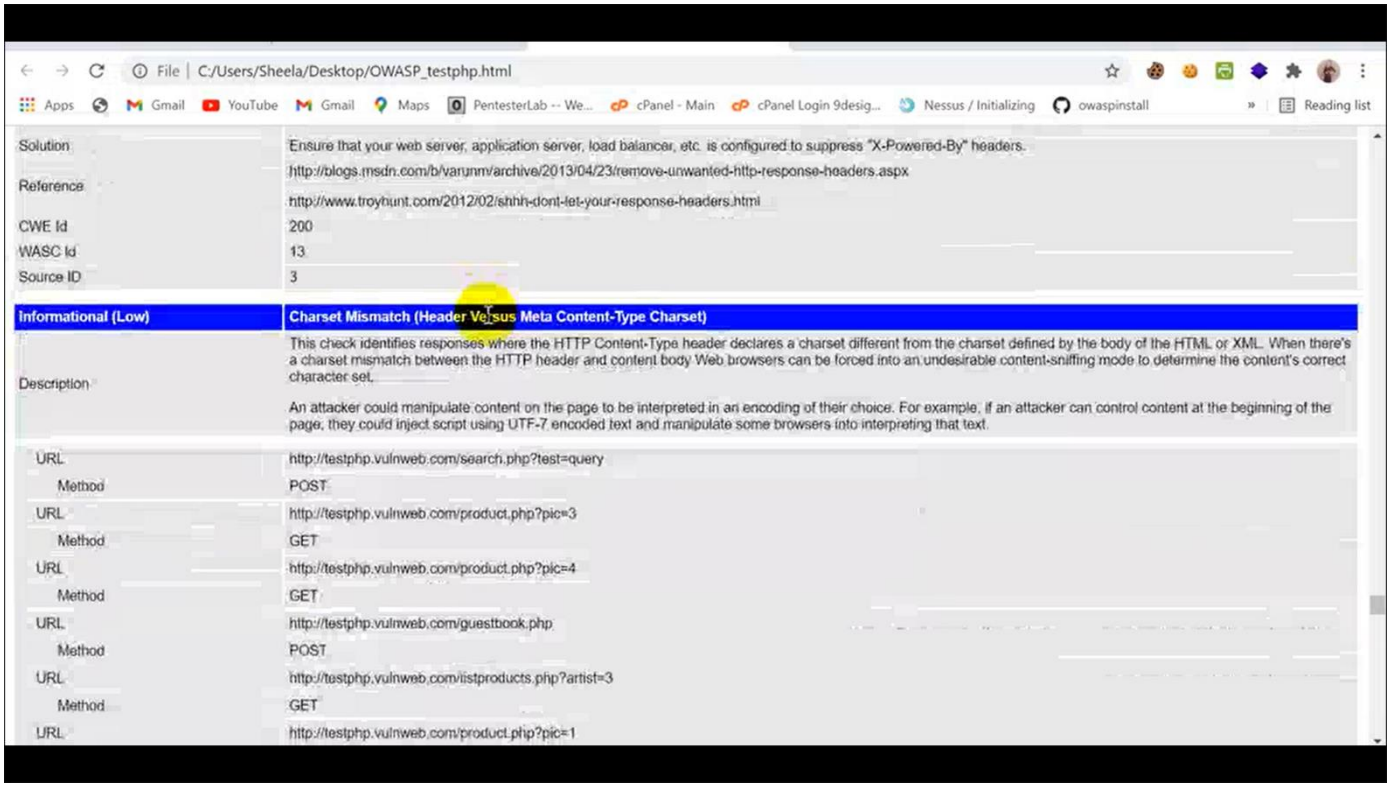

File | C:/Users/Sheela/Desktop/OWASP_testphp.html
ZAP Scanning Report
Summary of Alerts
Risk Level
Number of Alerts
High
4
Medium
1
Low
3
Informational
2
Alert Detail
High (Medium)
SQL Injection
Description
SQL injection may be possible.
URL
http://testphp.vulnweb.com/product.php?pic=8-2
Method
GET
Parameter
pic
Attack
8-2
URL
http://testphp.vulnweb.com/listproducts.php?cat=4+AND+1%3D1+--+
Method
GET
Parameter
cat
Attack
4 OR 1=1 --
18:34 / 40:19



File | C:/Users/Sheela/Desktop/OWASP_testphp.html

Apps Gmail YouTube Gmail Maps PentesterLab -- We... cPanel - Main cPanel Login 9desig... Nessus / Initializing owaspinstall Reading list

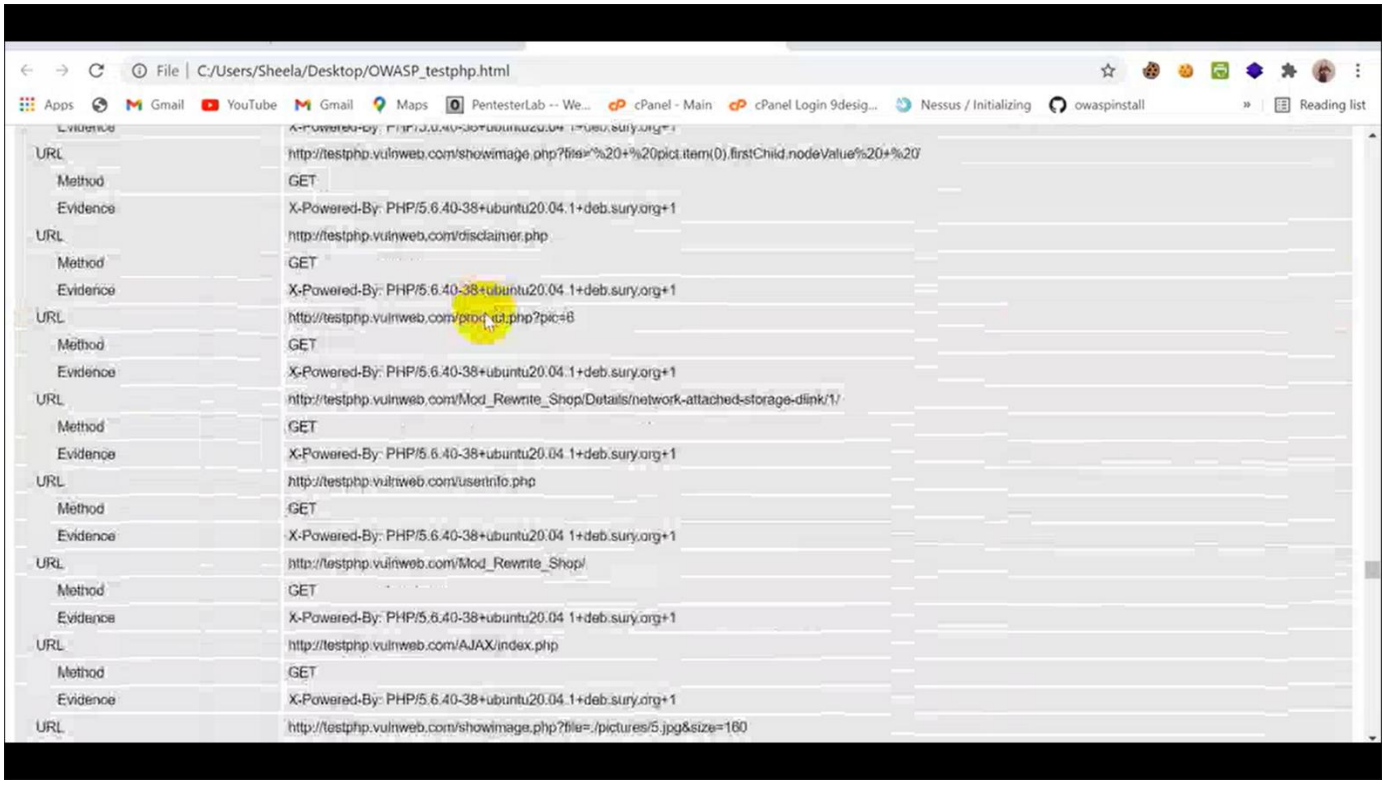
URL	http://testphp.vulnweb.com/listproducts.php?artist=3+AND+1%3D1+--+
Method	GET
Parameter	artist
Attack	3 OR 1=1 --
Instances	7
Solution	Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side. If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?' If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries. If database Stored Procedures can be used, use them. Do *not* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality! Do not create dynamic SQL queries using simple string concatenation. Escape all data received from the client. Apply a 'whitelist' of allowed characters, or a 'blacklist' of disallowed characters in user input. Apply the principle of least privilege by using the least privileged database user possible. In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact. Grant the minimum database access that is necessary for the application.
Other information	The original page results were successfully replicated using the expression [8-2] as the parameter value The parameter value being modified was stripped from the HTML output for the purposes of the comparison
Reference	https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html
CWE Id	89
WASC Id	19



File | C:/Users/Sheela/Desktop/OWASP_testphp.html

Apps Gmail YouTube Gmail Maps PentesterLab -- We... cPanel - Main cPanel Login 9desig... Nessus / Initializing owaspinstall Reading list

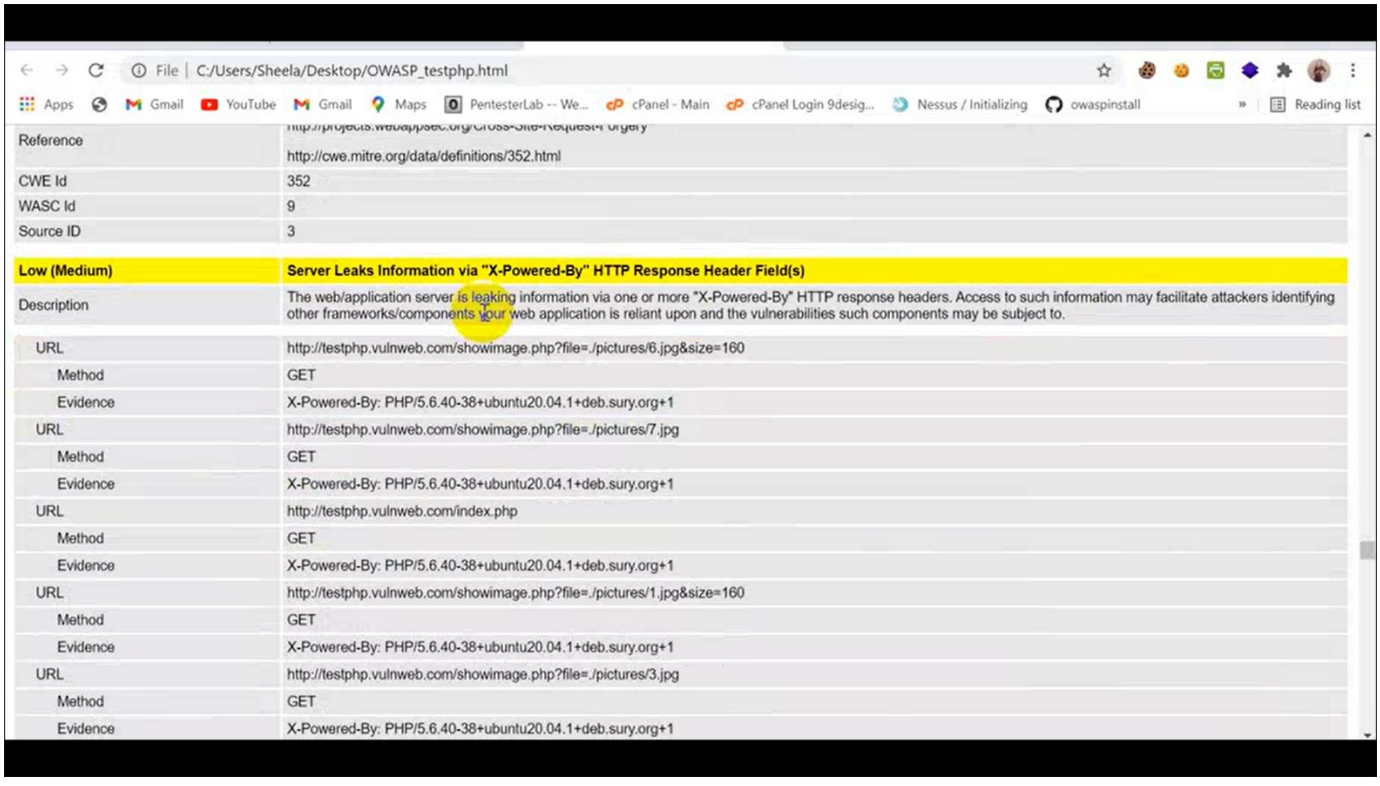
Solution	Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.
Reference	http://blogs.msdn.com/b/varunm/archive/2013/04/23/remove-unwanted-http-response-headers.aspx http://www.troyhunt.com/2012/02/shhh-dont-let-your-response-headers.html
CWE Id	200
WASC Id	13
Source ID	3
Informational (Low)	**Charset Mismatch (Header Versus Meta Content-Type Charset)**
Description	This check identifies responses where the HTTP Content-Type header declares a charset different from the charset defined by the body of the HTML or XML. When there's a charset mismatch between the HTTP header and content body Web browsers can be forced into an undesirable content-sniffing mode to determine the content's correct character set. An attacker could manipulate content on the page to be interpreted in an encoding of their choice. For example, if an attacker can control content at the beginning of the page, they could inject script using UTF-7 encoded text and manipulate some browsers into interpreting that text.
URL	http://testphp.vulnweb.com/search.php?test=query
Method	POST
URL	http://testphp.vulnweb.com/product.php?pic=3
Method	GET
URL	http://testphp.vulnweb.com/product.php?pic=4
Method	GET
URL	http://testphp.vulnweb.com/guestbook.php
Method	POST
URL	http://testphp.vulnweb.com/listproducts.php?artist=3
Method	GET
URL	http://testphp.vulnweb.com/product.php?pic=1



File | C:/Users/Sheela/Desktop/OWASP_testphp.html

Apps Gmail YouTube Gmail Maps PentesterLab -- We... cPanel - Main cPanel Login 9desig... Nessus / Initializing owaspinstall Reading list

URL	http://testphp.vulnweb.com/showimage.php?file=%20+%20pict.item(0).firstChild.nodeValue%20+%20
Method	GET
Evidence	X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
URL	http://testphp.vulnweb.com/disclaimer.php
Method	GET
Evidence	X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
URL	http://testphp.vulnweb.com/product.php?pic=6
Method	GET
Evidence	X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
URL	http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/
Method	GET
Evidence	X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
URL	http://testphp.vulnweb.com/userinfo.php
Method	GET
Evidence	X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
URL	http://testphp.vulnweb.com/Mod_Rewrite_Shop/
Method	GET
Evidence	X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
URL	http://testphp.vulnweb.com/AJAX/index.php
Method	GET
Evidence	X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
URL	http://testphp.vulnweb.com/showimage.php?file=./pictures/5.jpg&size=160

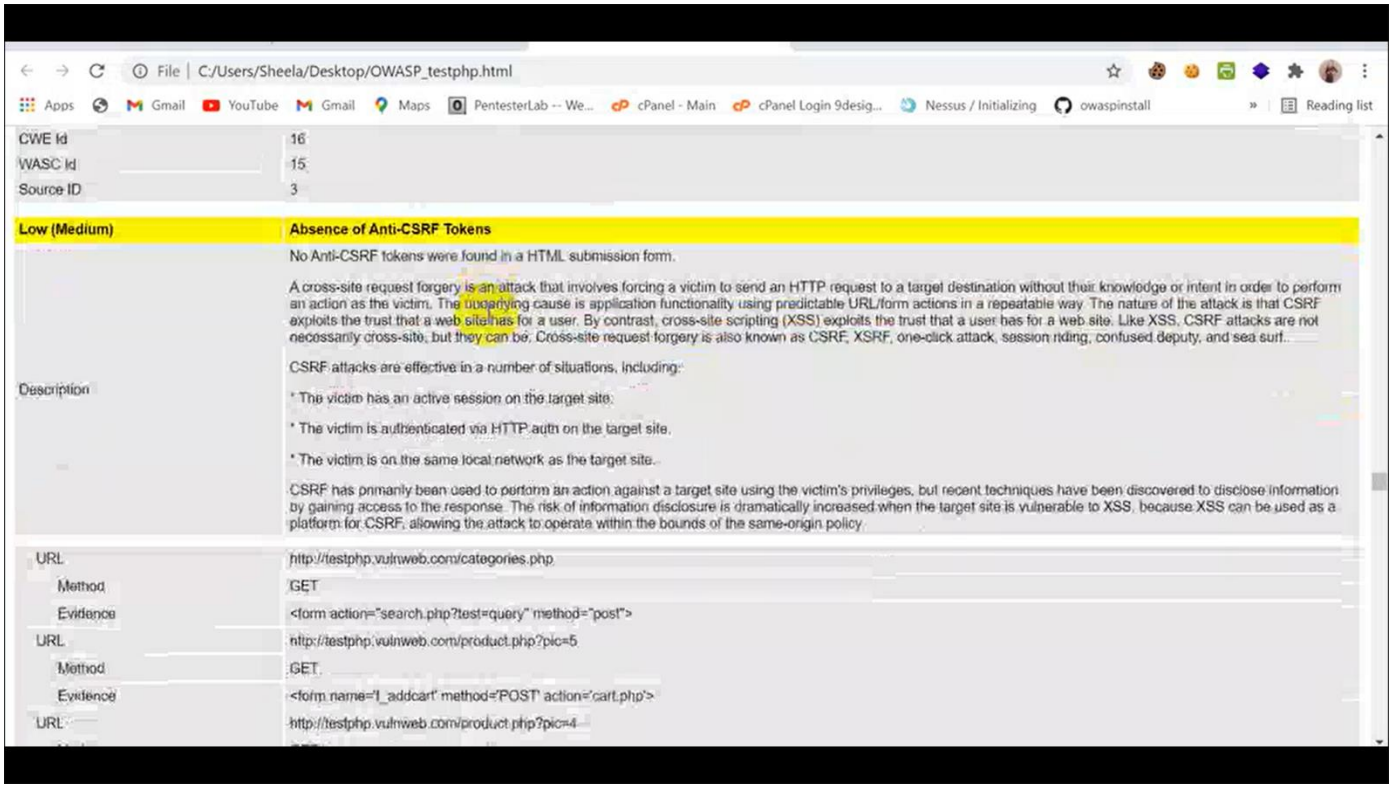


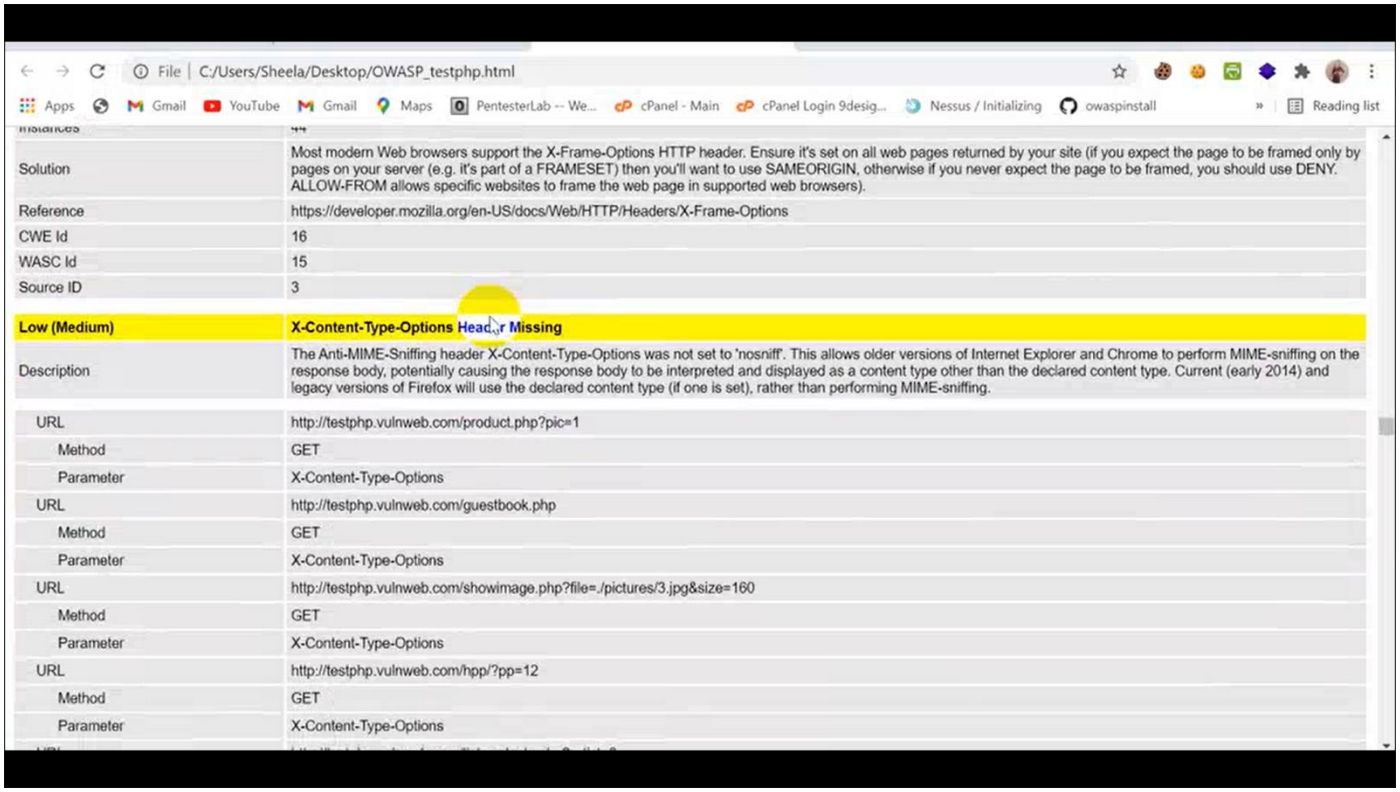
File | C:/Users/Sheela/Desktop/OWASP_testphp.html

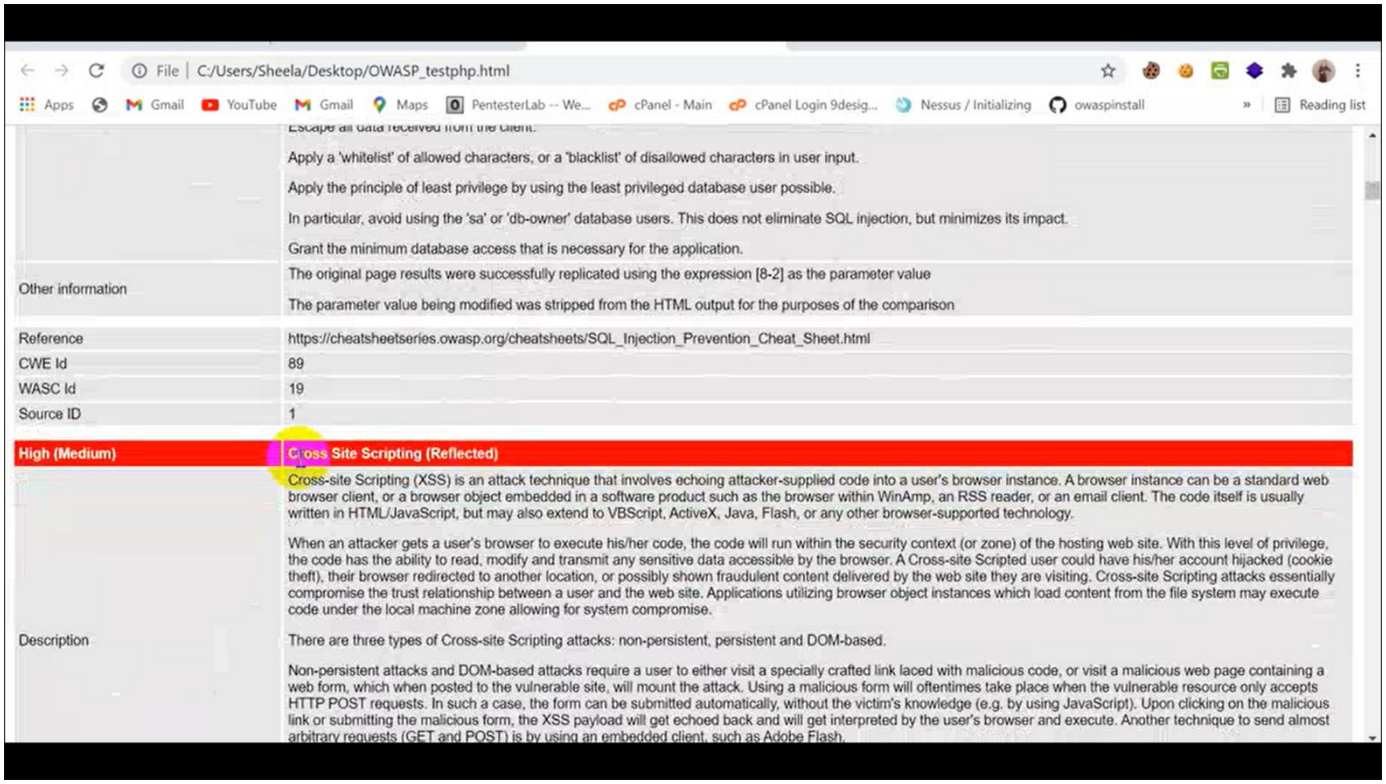
Apps Gmail YouTube Gmail Maps PentesterLab -- We... cPanel - Main cPanel Login 9desig... Nessus / Initializing owaspinstall Reading list

Reference	http://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Source ID	3

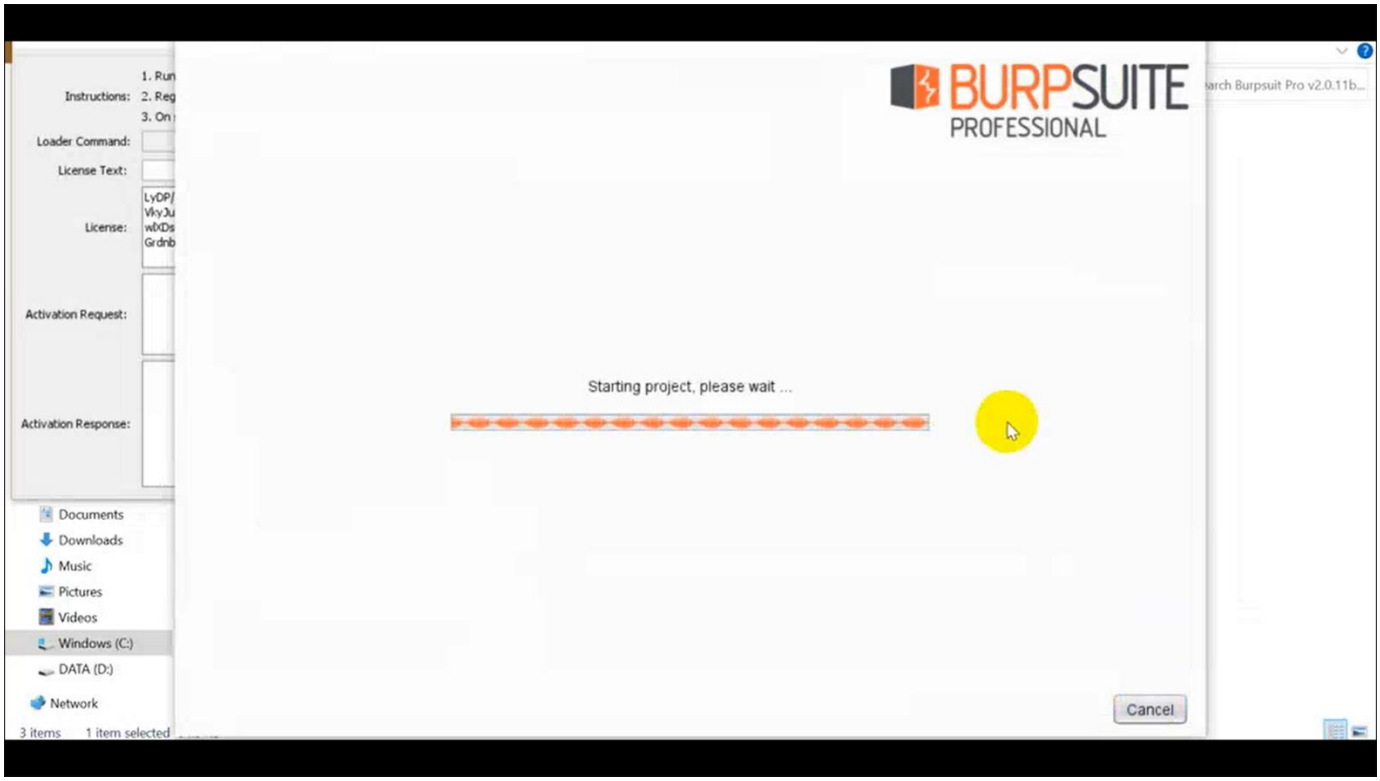
Low (Medium)	**Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)**
Description	The web/application server is leaking information via one or more "X-Powered-By" HTTP response headers. Access to such information may facilitate attackers identifying other frameworks/components your web application is reliant upon and the vulnerabilities such components may be subject to.
URL	http://testphp.vulnweb.com/showimage.php?file=./pictures/6.jpg&size=160
Method	GET
Evidence	X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
URL	http://testphp.vulnweb.com/showimage.php?file=./pictures/7.jpg
Method	GET
Evidence	X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
URL	http://testphp.vulnweb.com/index.php
Method	GET
Evidence	X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
URL	http://testphp.vulnweb.com/showimage.php?file=./pictures/1.jpg&size=160
Method	GET
Evidence	X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
URL	http://testphp.vulnweb.com/showimage.php?file=./pictures/3.jpg
Method	GET
Evidence	X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1

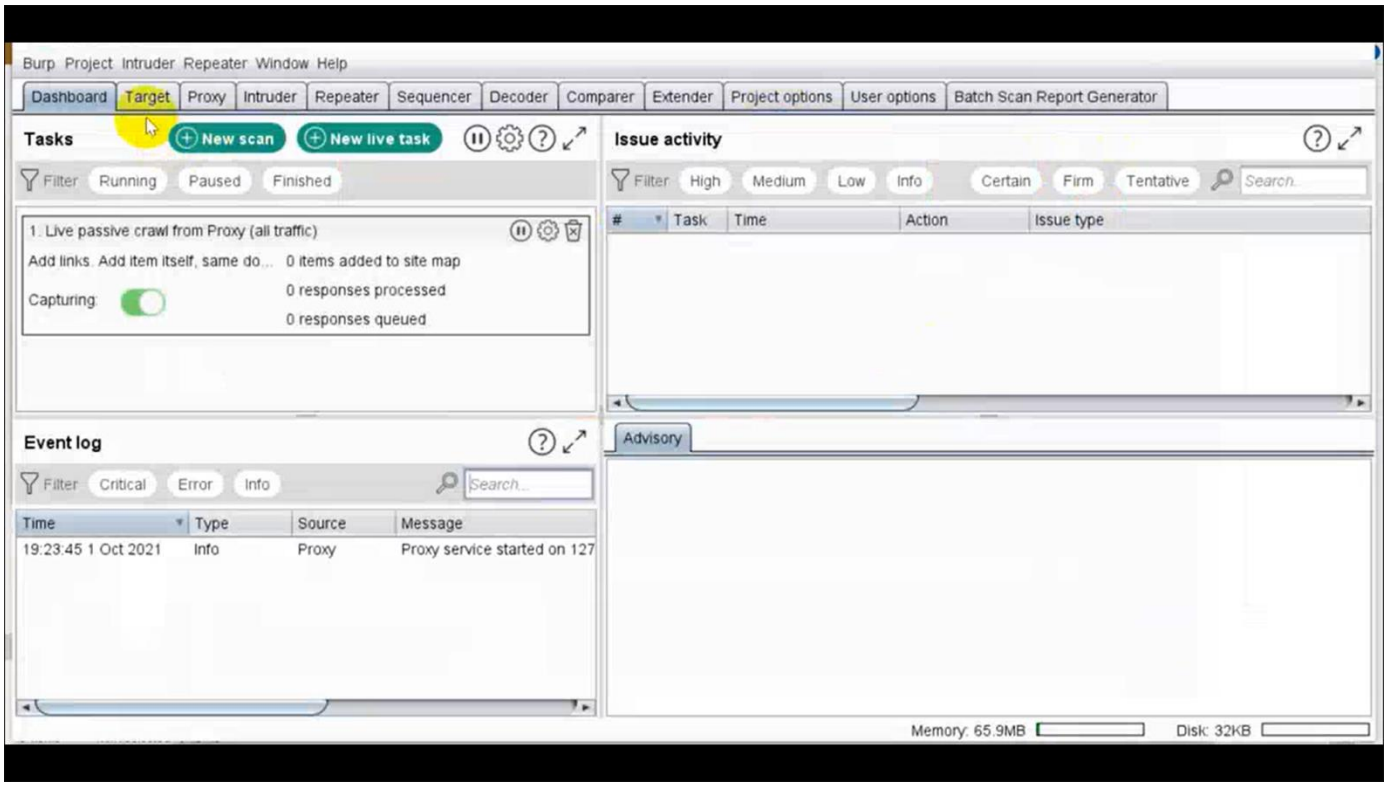

File | C:/Users/Sheela/Desktop/OWASP_testphp.html
Apps Gmail YouTube Gmail Maps PentesterLab -- We... cPanel - Main cPanel Login 9desig... Nessus / Initializing owaspinstall Reading list
CWE Id 16
WASC Id 15
Source ID 3
Low (Medium) Absence of Anti-CSRF Tokens
No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.
Description
CSRF attacks are effective in a number of situations, including:
* The victim has an active session on the target site.
* The victim is authenticated via HTTP auth on the target site.
* The victim is on the same local network as the target site.
CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.
URL http://testphp.vulnweb.com/categories.php
Method GET
Evidence <form action="search.php?test=query" method="post">
URL http://testphp.vulnweb.com/product.php?pic=5
Method GET
Evidence <form name='f_addcart' method='POST' action='cart.php'>
URL http://testphp.vulnweb.com/product.php?pic=4

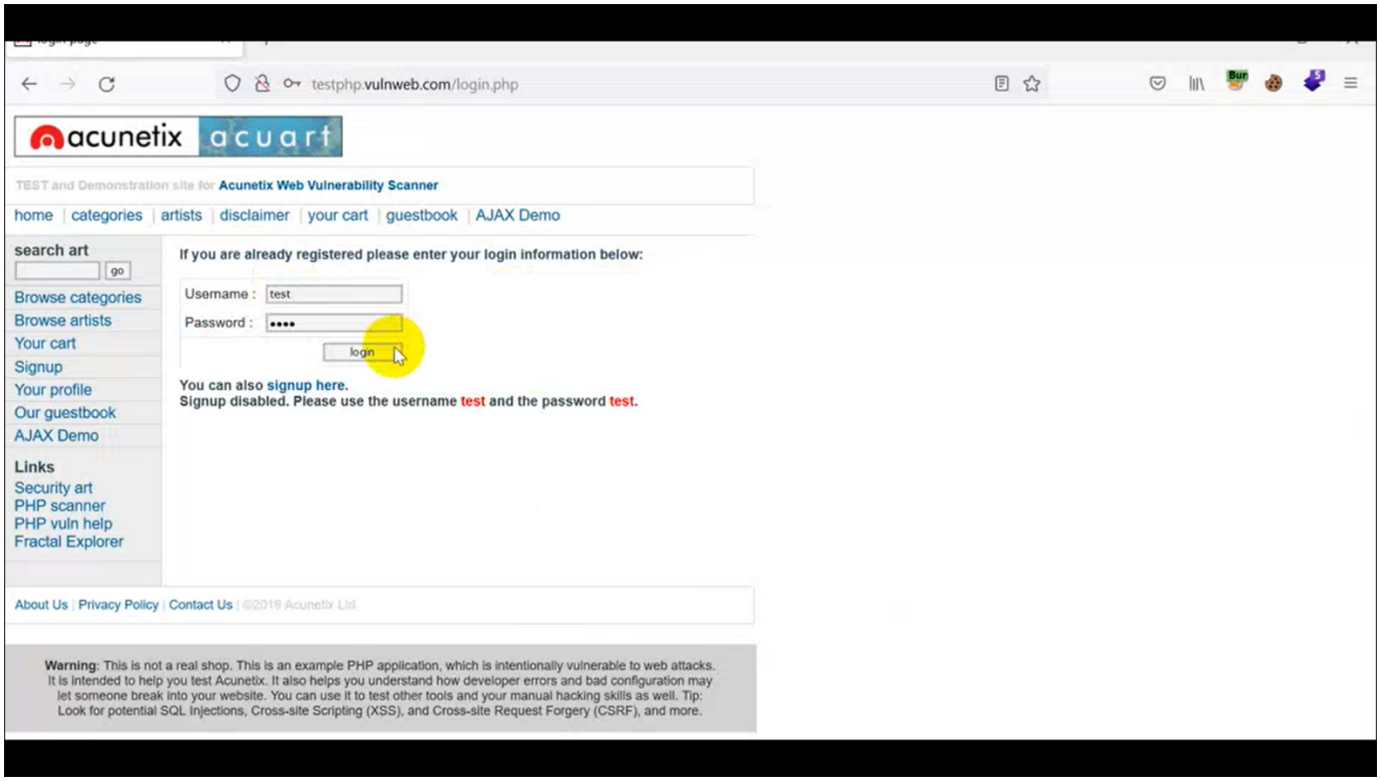

File | C:/Users/Sheela/Desktop/OWASP_testphp.html
Apps Gmail YouTube Gmail Maps PentesterLab -- We... cPanel - Main cPanel Login 9desig... Nessus / Initializing owaspinstall Reading list
Solution Most modern Web browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. ALLOW-FROM allows specific websites to frame the web page in supported web browsers).
Reference https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
CWE Id 16
WASC Id 15
Source ID 3
Low (Medium) X-Content-Type-Options Header Missing
Description The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
URL http://testphp.vulnweb.com/product.php?pic=1
Method GET
Parameter X-Content-Type-Options
URL http://testphp.vulnweb.com/guestbook.php
Method GET
Parameter X-Content-Type-Options
URL http://testphp.vulnweb.com/showimage.php?file=./pictures/3.jpg&size=160
Method GET
Parameter X-Content-Type-Options
URL http://testphp.vulnweb.com/hpp/?pp=12
Method GET
Parameter X-Content-Type-Options

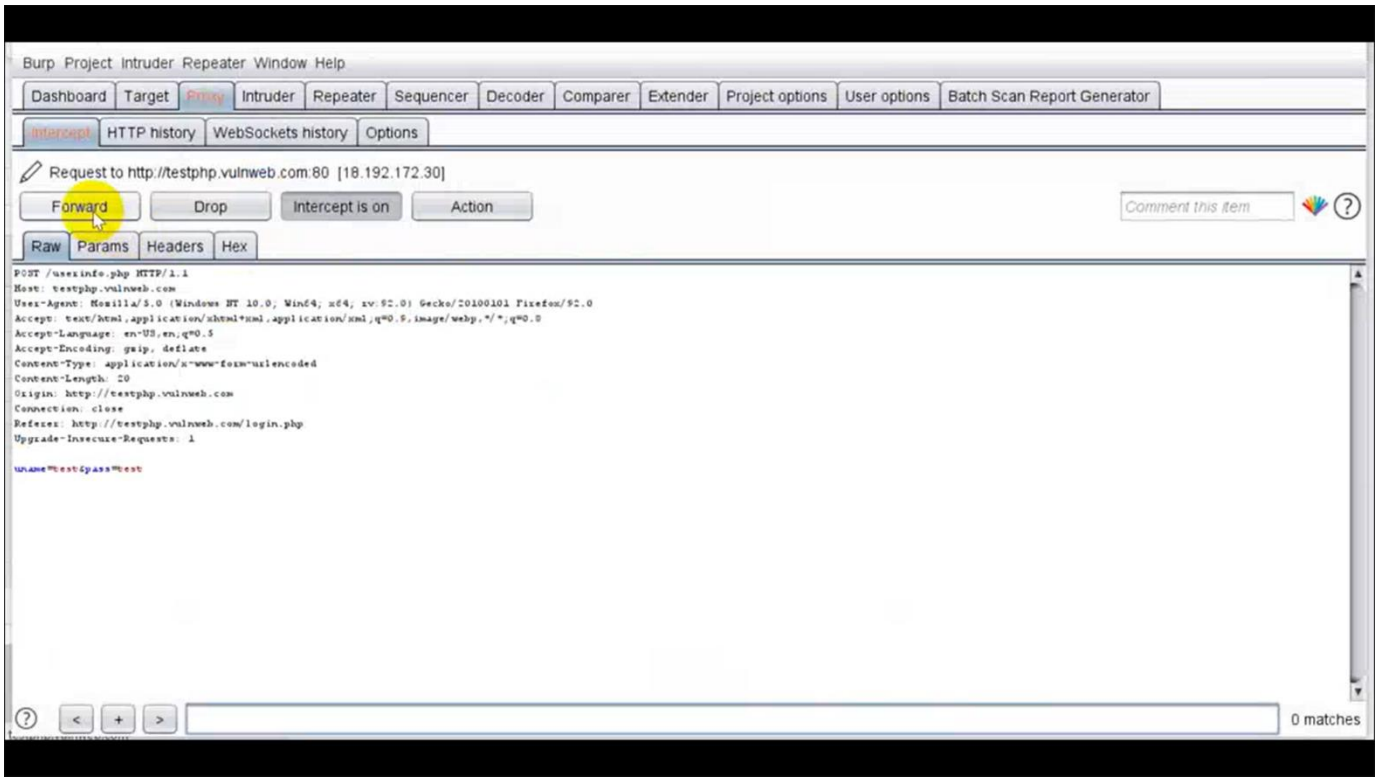

File | C:/Users/Sheela/Desktop/OWASP_testphp.html
Apps
Gmail
YouTube
Gmail
Maps
PentesterLab -- We...
cPanel - Main
cPanel Login 9desig...
Nessus / Initializing
owaspinstall
Reading list
Escape all data received from the client.
Apply a 'whitelist' of allowed characters, or a 'blacklist' of disallowed characters in user input.
Apply the principle of least privilege by using the least privileged database user possible.
In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact.
Grant the minimum database access that is necessary for the application.
Other information
The original page results were successfully replicated using the expression [8-2] as the parameter value
The parameter value being modified was stripped from the HTML output for the purposes of the comparison
Reference
https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html
CWE Id
89
WASC Id
19
Source ID
1
High (Medium)
Cross Site Scripting (Reflected)
Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.
When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Applications utilizing browser object instances which load content from the file system may execute code under the local machine zone allowing for system compromise.
Description
There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based.
Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, or visit a malicious web page containing a web form, which when posted to the vulnerable site, will mount the attack. Using a malicious form will oftentimes take place when the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by using an embedded client, such as Adobe Flash.

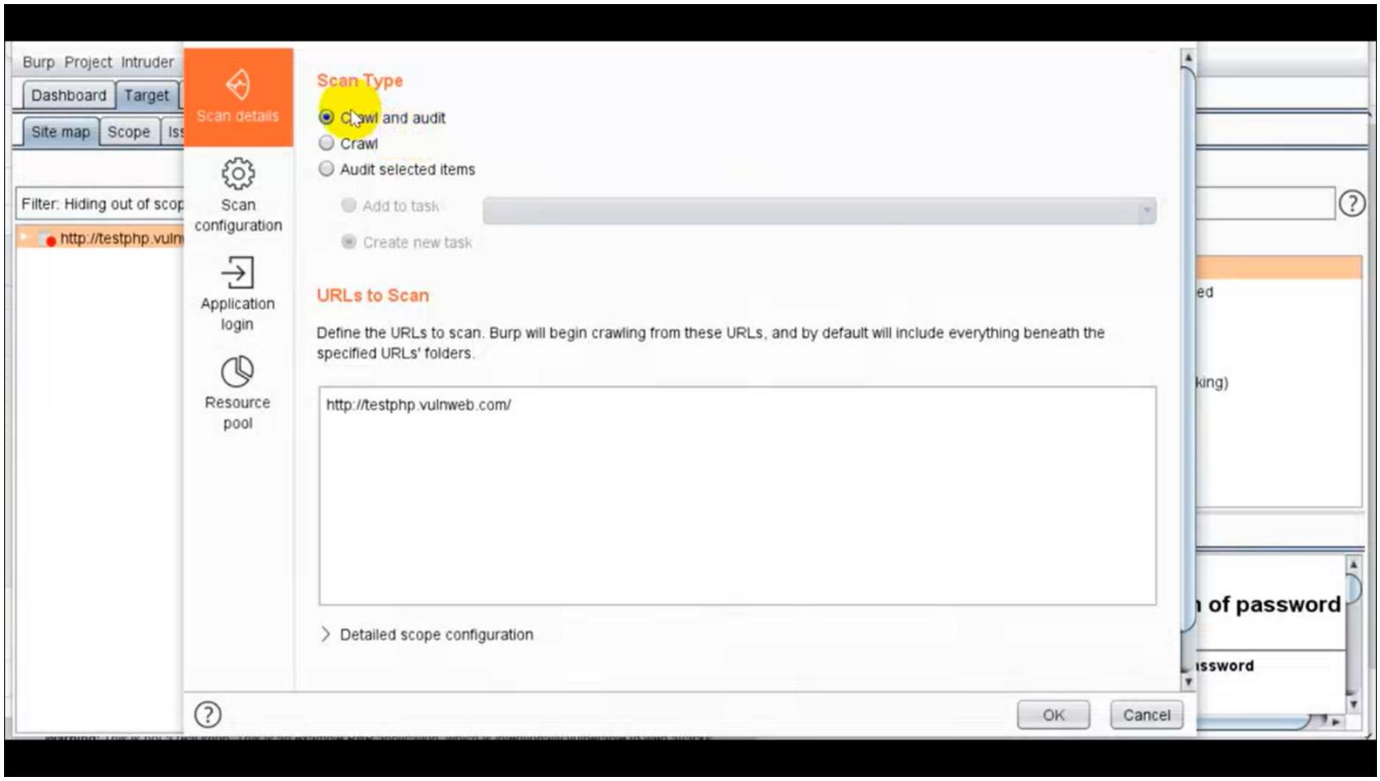

Burp Suite Pro 2.0.11 Beta Loader & Keygen - By MRun
Instructions:
1. Run Burp Suite Pro with the loader specified in the bootclasspath
2. Register using manual activation
3. On subsequent runs you must execute burpsuite with the loader otherwise it will become unregistered
Loader Command:
java -Xbootclasspath/p:BurpKeygen.jar -jar burpsuite_pro_v2.0.11beta.jar
Run
License Text:
licensed to MRun
License:
Activation Request:
Activation Response:
BURPSUITE
PROFESSIONAL
Search Burpsuit Pro v2.0.11b...
65 KB
159 KB
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
DATA (D:)
Network
3 items
1 item selected 64.5 KB

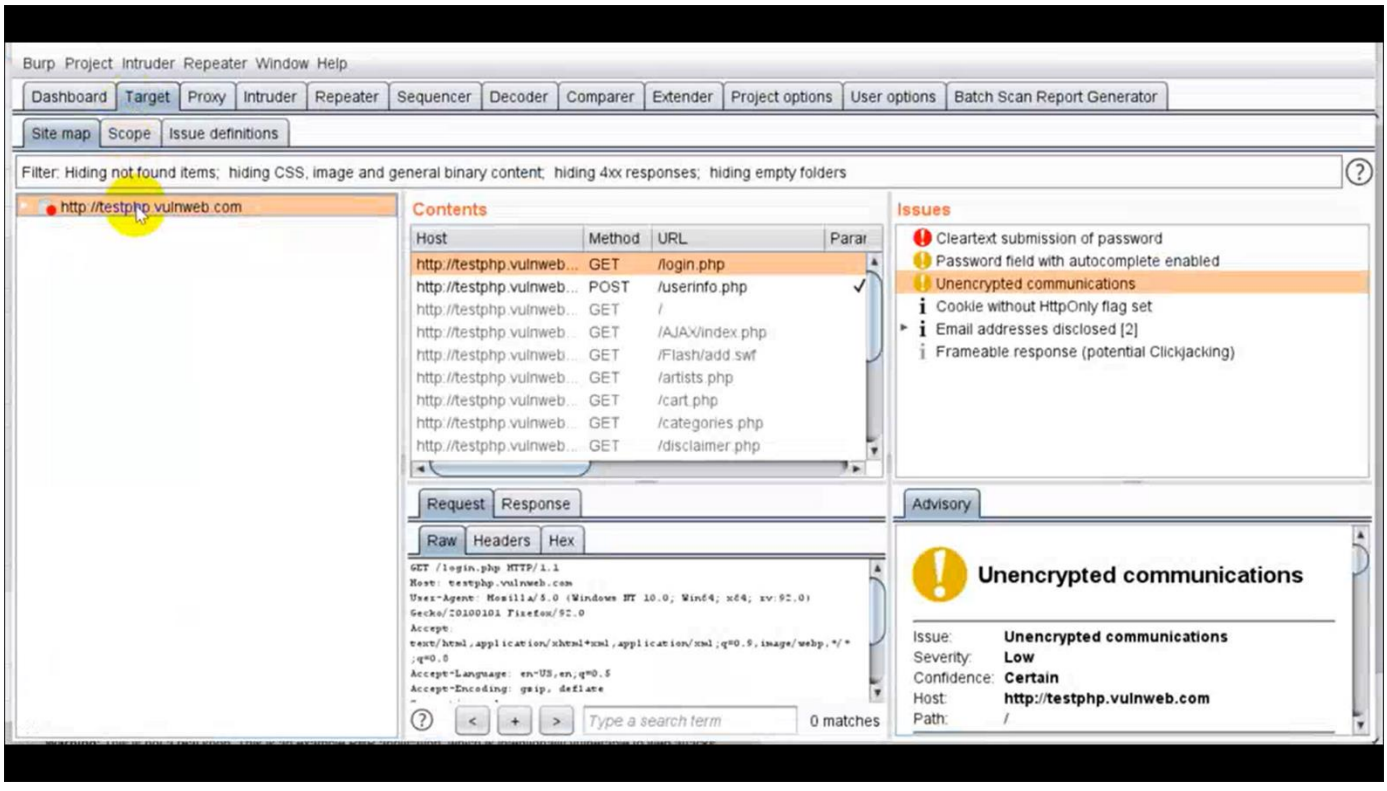

Instructions:
Loader Command:
License Text:
License:
Activation Request:
Activation Response:
BURPSUITE
PROFESSIONAL
Starting project, please wait ...
Cancel
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
DATA (D:)
Network
3 items
1 item selected

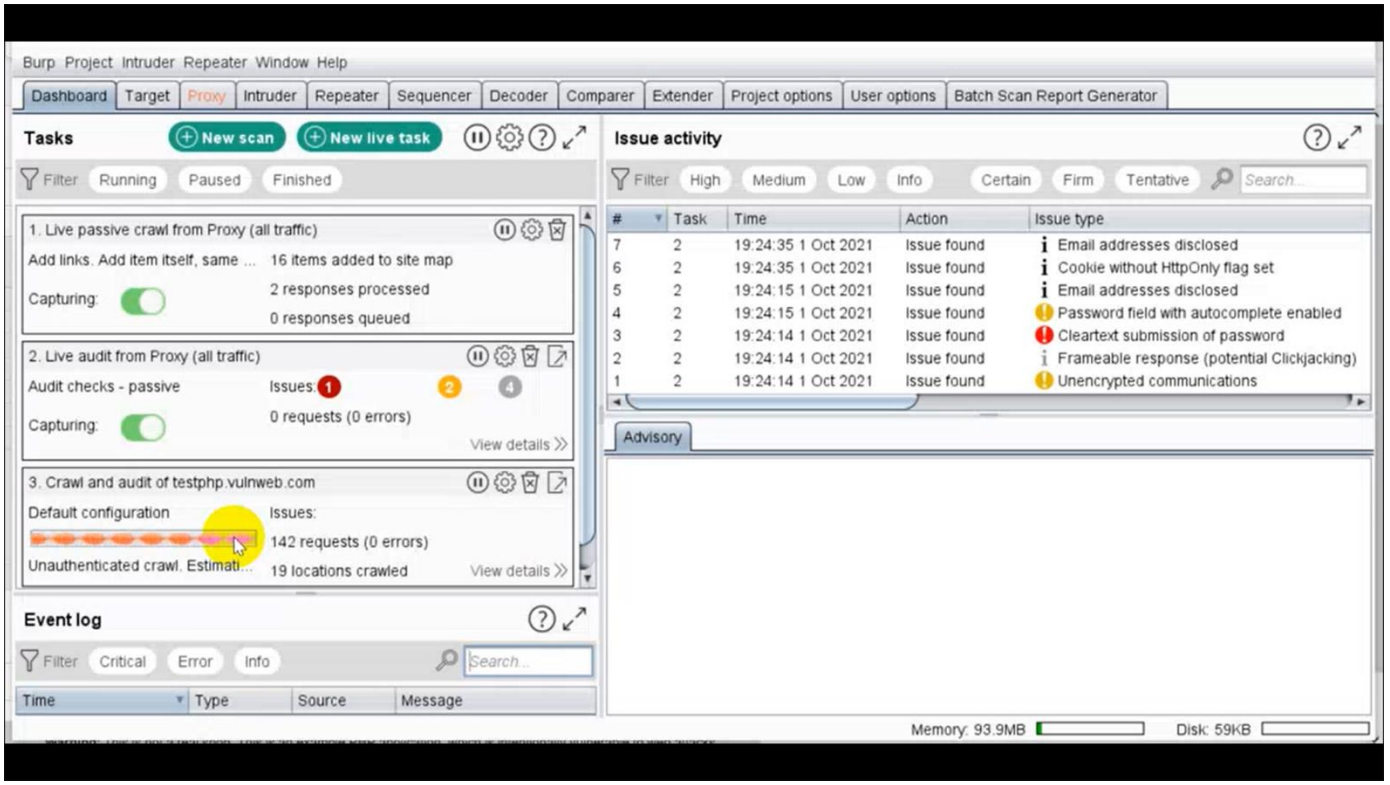

Burp Project Intruder Repeater Window Help
Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Extender
Project options
User options
Batch Scan Report Generator
Tasks
New scan
New live task
Filter
Running
Paused
Finished
1. Live passive crawl from Proxy (all traffic)
Add links. Add item itself, same do...
0 items added to site map
Capturing:
0 responses processed
0 responses queued
Issue activity
Filter
High
Medium
Low
Info
Certain
Firm
Tentative
Search...
#
Task
Time
Action
Issue type
Event log
Filter
Critical
Error
Info
Search...
Time
Type
Source
Message
19:23:45 1 Oct 2021
Info
Proxy
Proxy service started on 127
Advisory
Memory: 65.9MB
Disk: 32KB

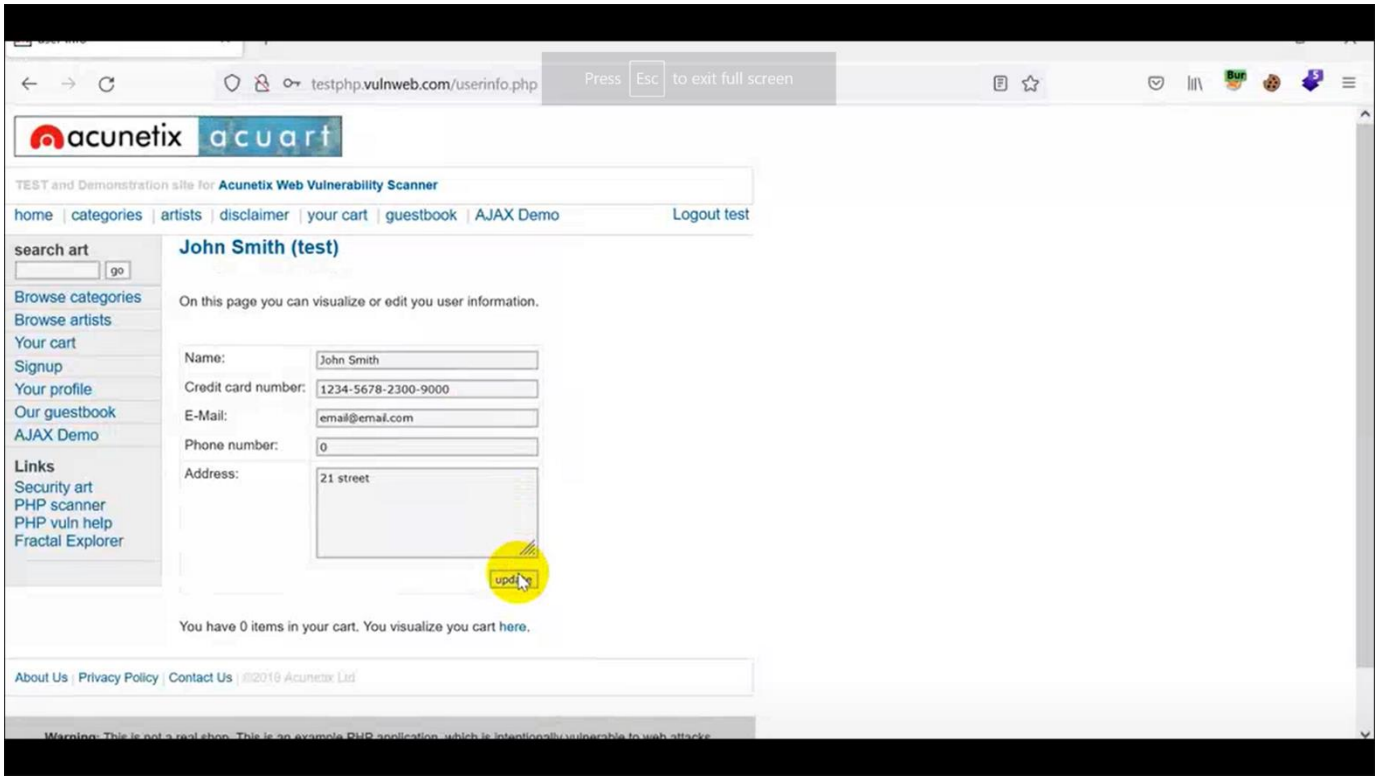

testphp.vulnweb.com/login.php
acunetix acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo
search art
go
Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo
Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer
If you are already registered please enter your login information below:
Username : test
Password : ••••
login
You can also signup here.
Signup disabled. Please use the username test and the password test.
About Us | Privacy Policy | Contact Us | ©2019 Acunetix Ltd
Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

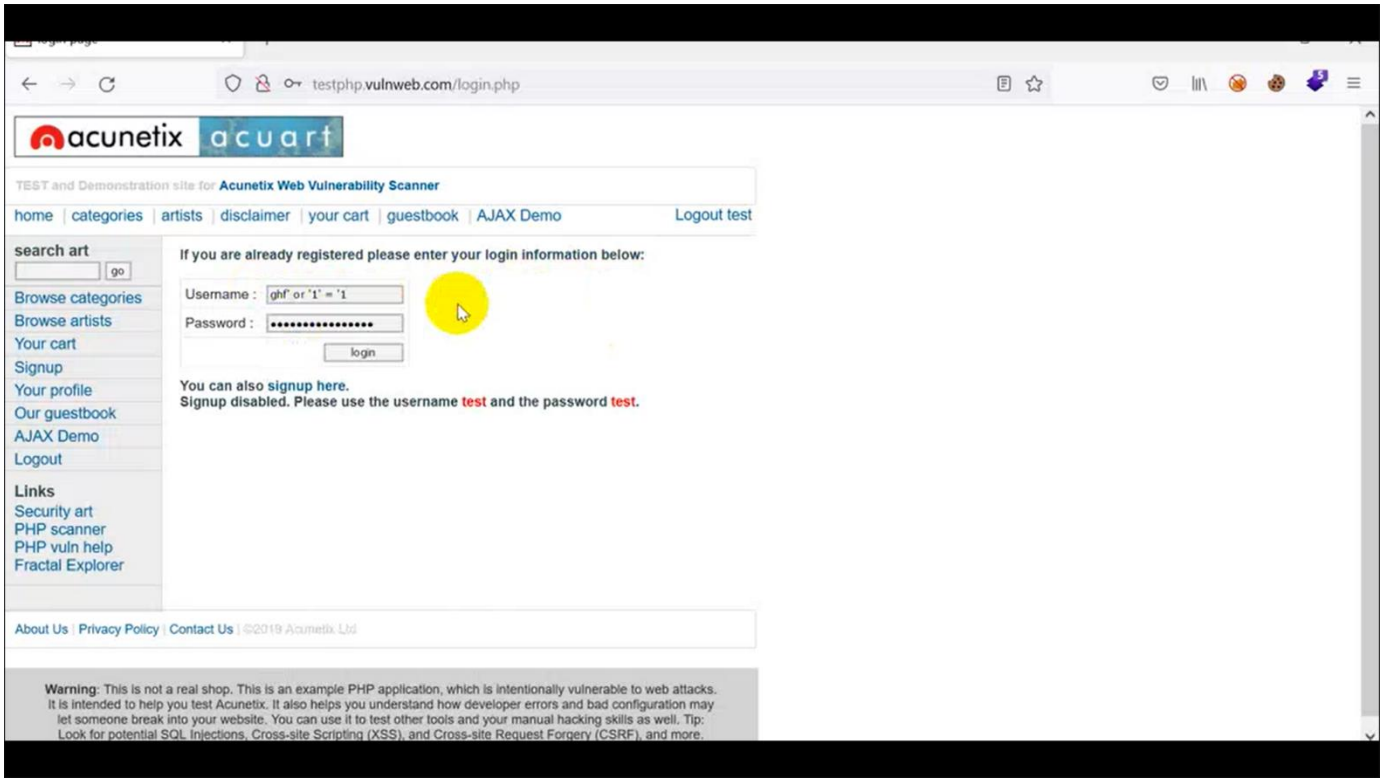

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Batch Scan Report Generator
Intercept HTTP history WebSockets history Options
Request to http://testphp.vulnweb.com:80 [18.192.172.30]
Forward Drop Intercept is on Action
Comment this item
Raw Params Headers Hex
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 20
Origin: http://testphp.vulnweb.com
Connection: close
Referer: http://testphp.vulnweb.com/login.php
Upgrade-Insecure-Requests: 1
uname=test&pass=test
0 matches

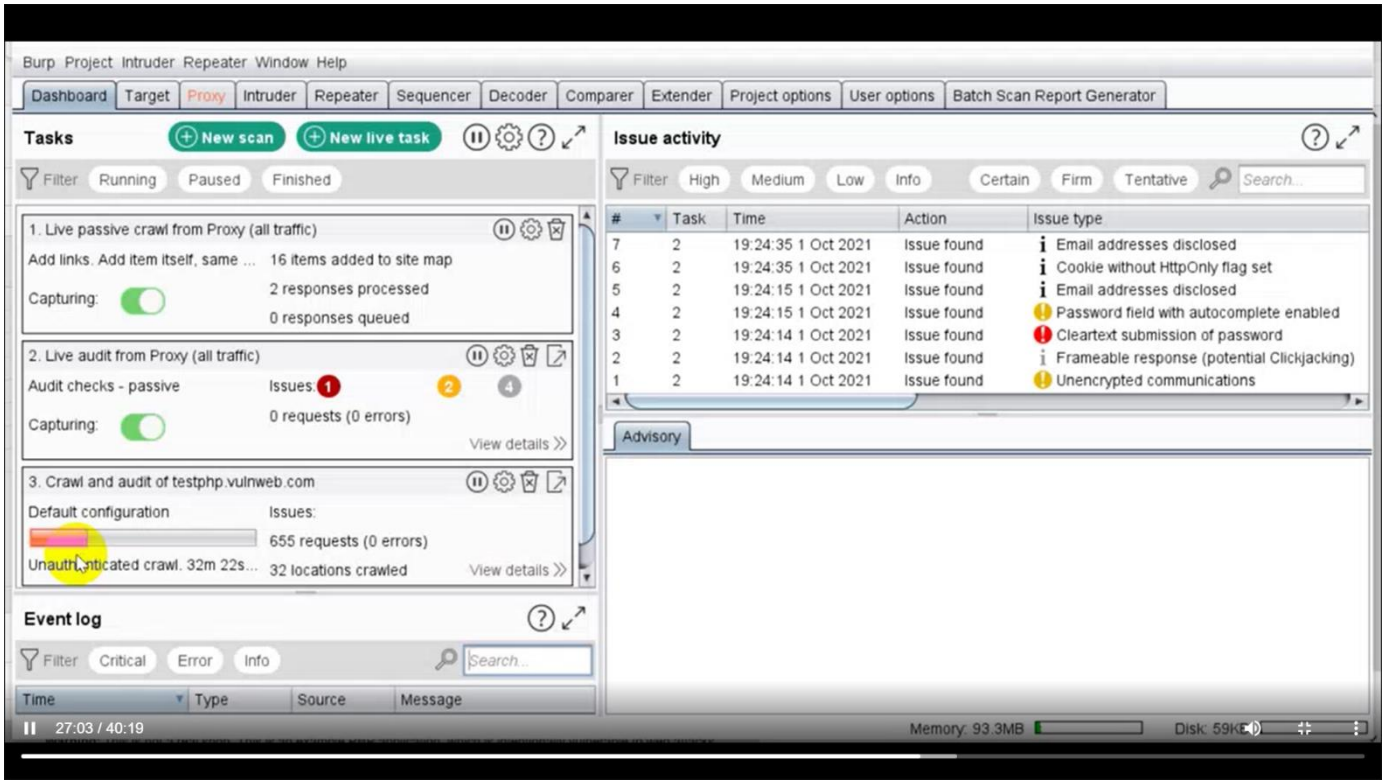

Burp Project Intruder
Dashboard Target
Site map Scope
Filter: Hiding out of scop
http://testphp.vuln
Scan details
Scan configuration
Application login
Resource pool
Scan Type
Crawl and audit
Crawl
Audit selected items
Add to task
Create new task
URLs to Scan
Define the URLs to scan. Burp will begin crawling from these URLs, and by default will include everything beneath the specified URLs' folders.
http://testphp.vulnweb.com/
Detailed scope configuration
OK
Cancel

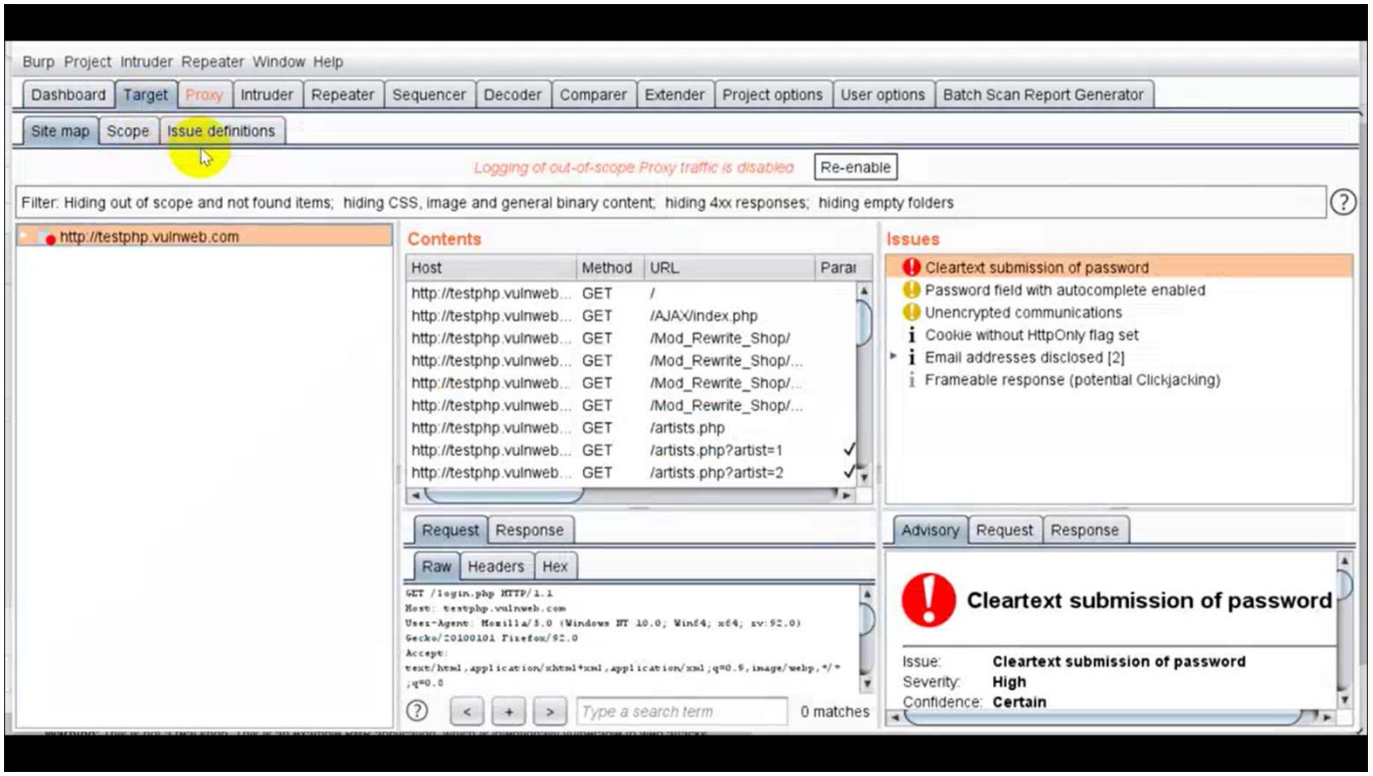

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Batch Scan Report Generator
Site map Scope Issue definitions
Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders
http://testphp.vulnweb.com
Contents
Host Method URL Parar
http://testphp.vulnweb... GET /login.php
http://testphp.vulnweb... POST /userinfo.php
http://testphp.vulnweb... GET /
http://testphp.vulnweb... GET /AJAX/index.php
http://testphp.vulnweb... GET /Flash/add.swf
http://testphp.vulnweb... GET /artists.php
http://testphp.vulnweb... GET /cart.php
http://testphp.vulnweb... GET /categories.php
http://testphp.vulnweb... GET /disclaimer.php
Issues
Cleartext submission of password
Password field with autocomplete enabled
Unencrypted communications
Cookie without HttpOnly flag set
Email addresses disclosed [2]
Frameable response (potential Clickjacking)
Request Response
Raw Headers Hex
GET /login.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0)
Gecko/20100101 Firefox/92.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*
;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Type a search term
0 matches
Advisory
Unencrypted communications
Issue: Unencrypted communications
Severity: Low
Confidence: Certain
Host: http://testphp.vulnweb.com
Path: /

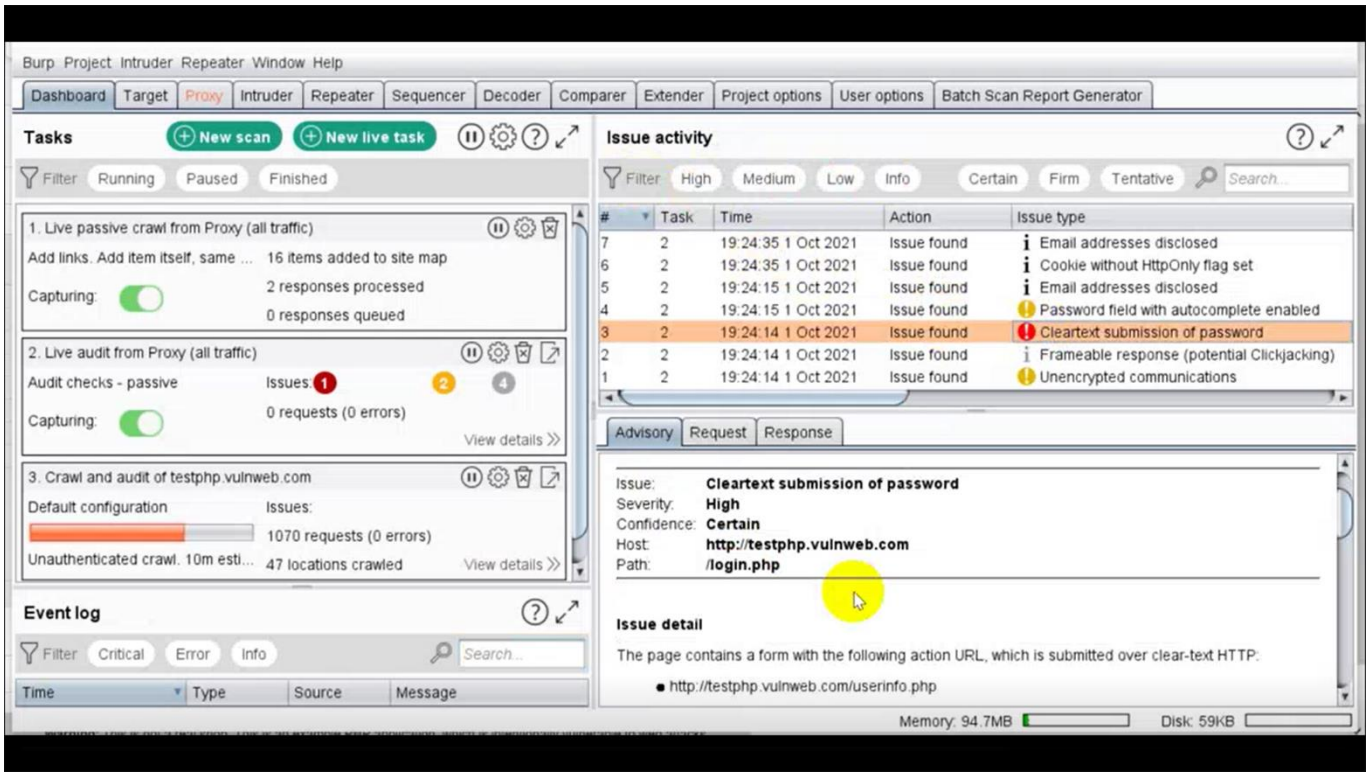

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Batch Scan Report Generator
Tasks
New scan
New live task
Filter Running Paused Finished
1. Live passive crawl from Proxy (all traffic)
Add links. Add item itself, same ...
16 items added to site map
Capturing:
2 responses processed
0 responses queued
2. Live audit from Proxy (all traffic)
Audit checks - passive
Issues:
Capturing:
0 requests (0 errors)
View details
3. Crawl and audit of testphp.vulnweb.com
Default configuration
Issues:
142 requests (0 errors)
Unauthenticated crawl. Estimati...
19 locations crawled
View details
Event log
Filter Critical Error Info
Search...
Time Type Source Message
Issue activity
Filter High Medium Low Info Certain Firm Tentative
Search...
Task Time Action Issue type
7 2 19:24:35 1 Oct 2021 Issue found Email addresses disclosed
6 2 19:24:35 1 Oct 2021 Issue found Cookie without HttpOnly flag set
5 2 19:24:15 1 Oct 2021 Issue found Email addresses disclosed
4 2 19:24:15 1 Oct 2021 Issue found Password field with autocomplete enabled
3 2 19:24:14 1 Oct 2021 Issue found Cleartext submission of password
2 2 19:24:14 1 Oct 2021 Issue found Frameable response (potential Clickjacking)
1 2 19:24:14 1 Oct 2021 Issue found Unencrypted communications
Advisory
Memory: 93.9MB
Disk: 59KB

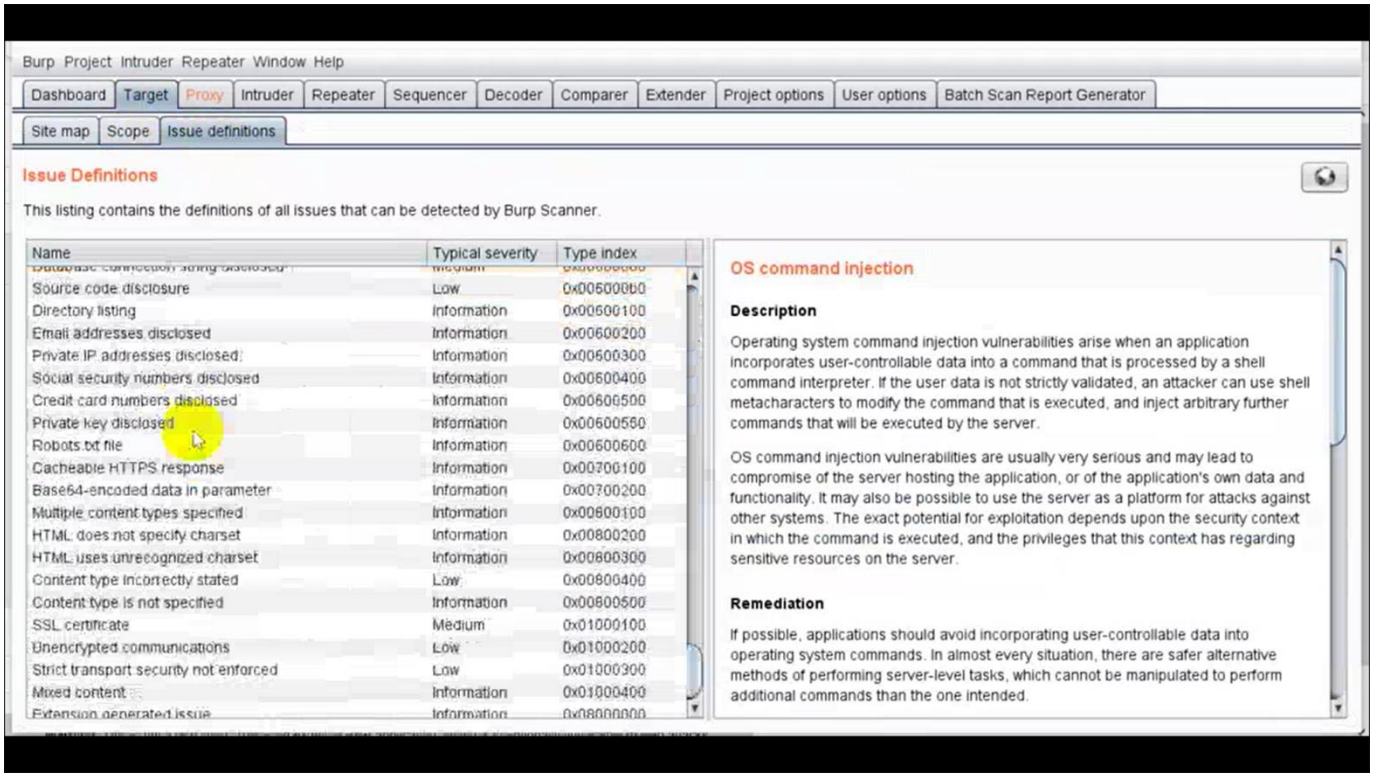

testphp.vulnweb.com/userinfo.php
Press Esc to exit full screen
acunetix acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo
Logout test
search art
go
Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo
Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer
John Smith (test)
On this page you can visualize or edit you user information.
Name: John Smith
Credit card number: 1234-5678-2300-9000
E-Mail: email@email.com
Phone number: 0
Address: 21 street
update
You have 0 items in your cart. You visualize you cart here.
About Us | Privacy Policy | Contact Us

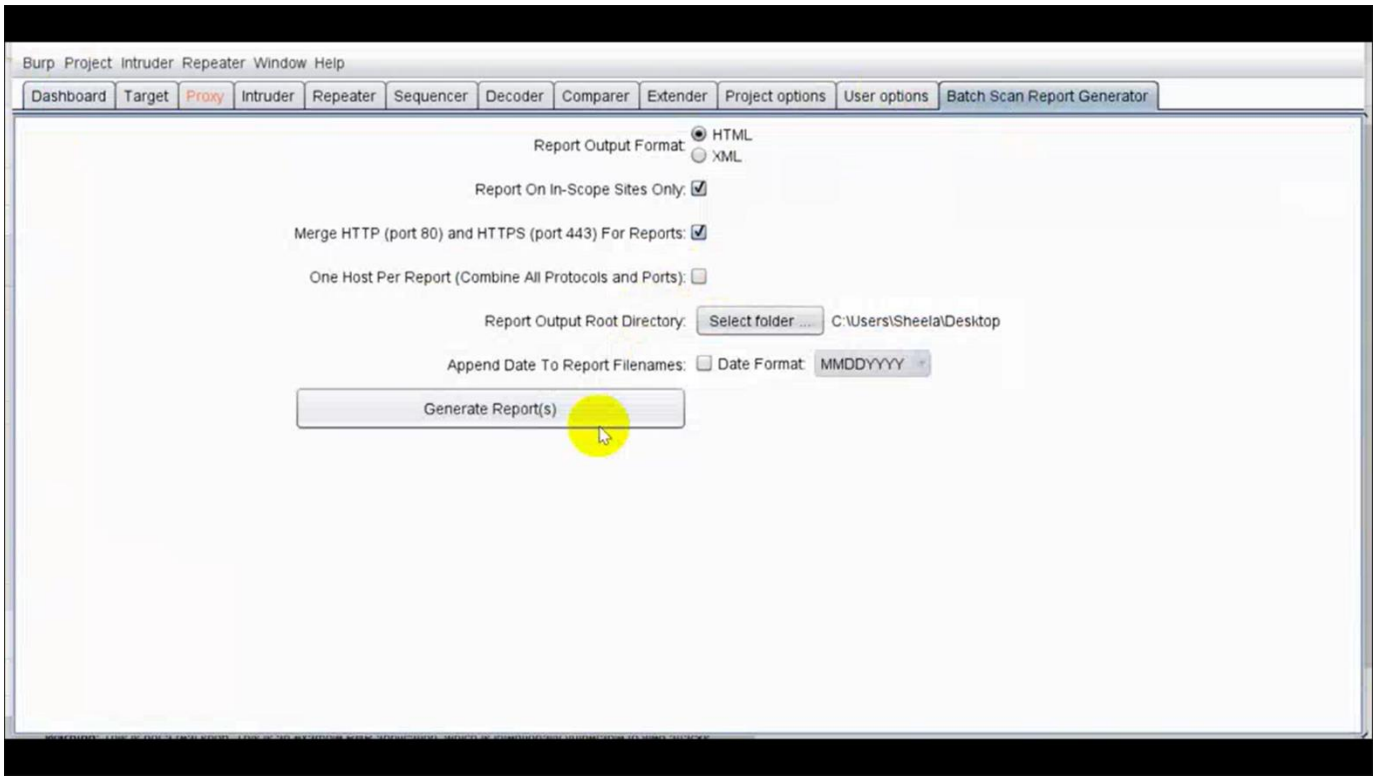

testphp.vulnweb.com/login.php
acunetix acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo
Logout test
search art
go
Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo
Logout
Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer
If you are already registered please enter your login information below:
Username : ghf' or '1' = '1
Password :
login
You can also signup here.
Signup disabled. Please use the username test and the password test.
About Us | Privacy Policy | Contact Us
Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks.
It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may
let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip:
Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

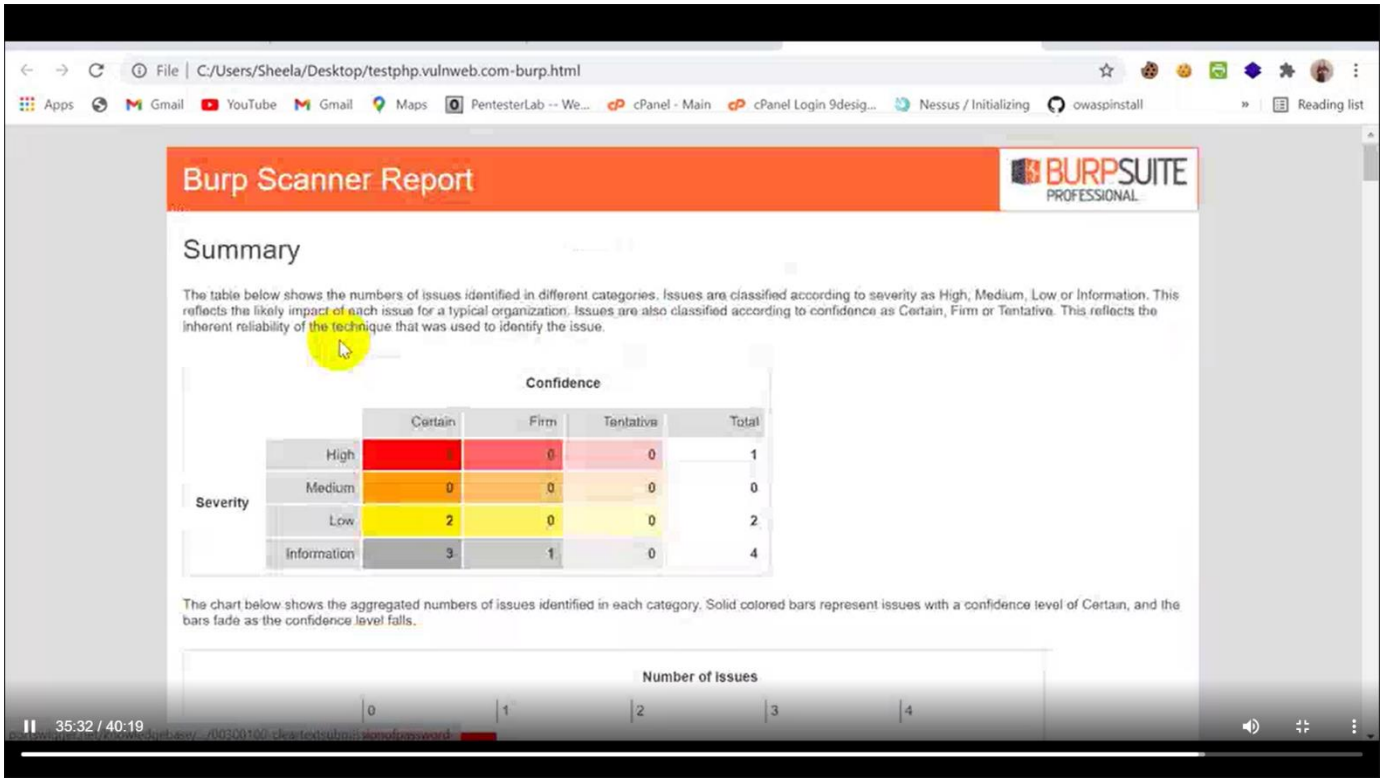

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Batch Scan Report Generator
Tasks
New scan
New live task
Filter Running Paused Finished
1. Live passive crawl from Proxy (all traffic)
Add links. Add item itself, same ...
16 items added to site map
Capturing:
2 responses processed
0 responses queued
2. Live audit from Proxy (all traffic)
Audit checks - passive
Issues:
Capturing:
0 requests (0 errors)
View details
3. Crawl and audit of testphp.vulnweb.com
Default configuration
Issues:
655 requests (0 errors)
Unauthenticated crawl. 32m 22s...
32 locations crawled
View details
Event log
Filter Critical Error Info
Search...
Time Type Source Message
Issue activity
Filter High Medium Low Info Certain Firm Tentative
Search...
Task Time Action Issue type
7 2 19:24:35 1 Oct 2021 Issue found Email addresses disclosed
6 2 19:24:35 1 Oct 2021 Issue found Cookie without HttpOnly flag set
5 2 19:24:15 1 Oct 2021 Issue found Email addresses disclosed
4 2 19:24:15 1 Oct 2021 Issue found Password field with autocomplete enabled
3 2 19:24:14 1 Oct 2021 Issue found Cleartext submission of password
2 2 19:24:14 1 Oct 2021 Issue found Frameable response (potential Clickjacking)
1 2 19:24:14 1 Oct 2021 Issue found Unencrypted communications
Advisory
27:03 / 40:19
Memory: 93.3MB
Disk: 59K

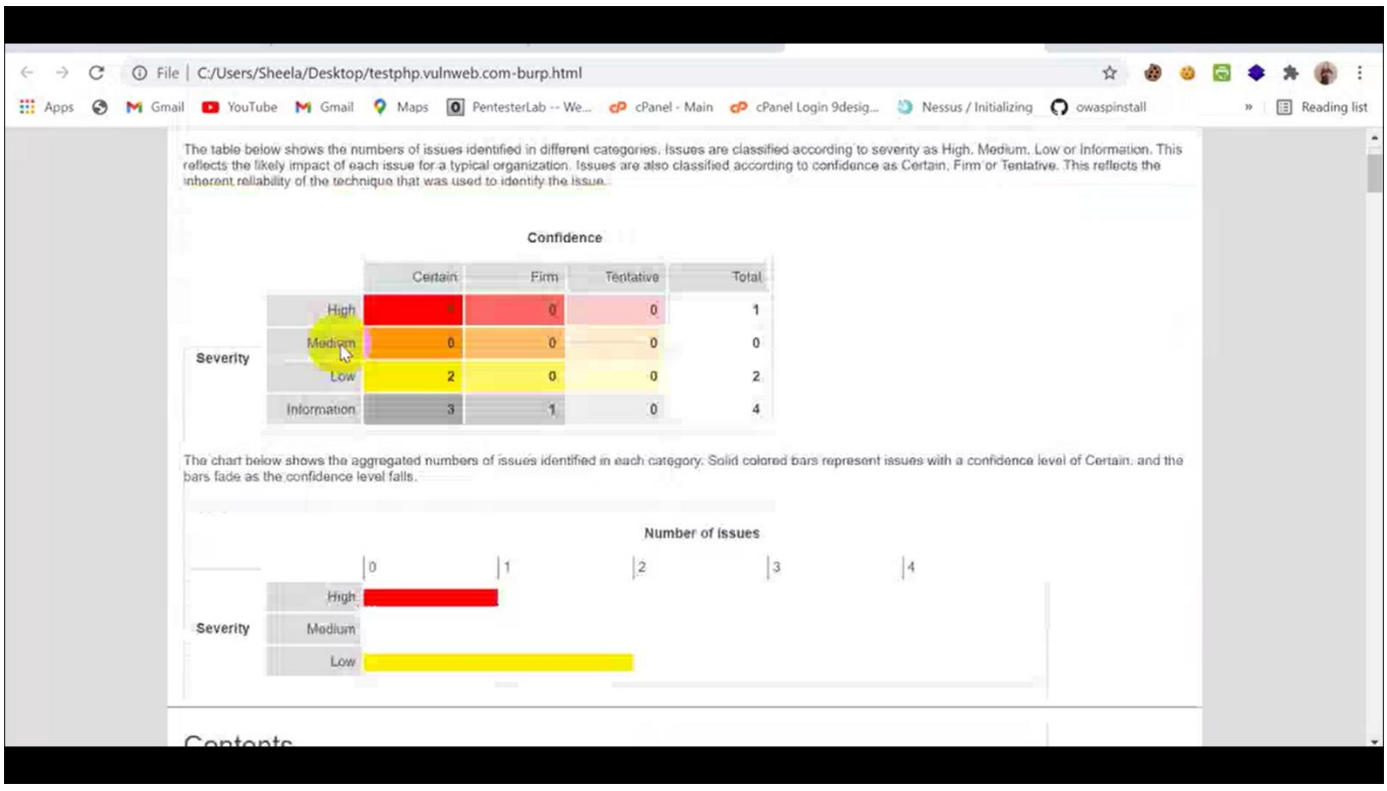

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Batch Scan Report Generator
Site map Scope Issue definitions
Logging of out-of-scope Proxy traffic is disabled
Re-enable
Filter: Hiding out of scope and not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders
http://testphp.vulnweb.com
Contents
Host Method URL Parai
http://testphp.vulnweb... GET /
http://testphp.vulnweb... GET /AJAX/index.php
http://testphp.vulnweb... GET /Mod_Rewrite_Shop/
http://testphp.vulnweb... GET /Mod_Rewrite_Shop/...
http://testphp.vulnweb... GET /Mod_Rewrite_Shop/...
http://testphp.vulnweb... GET /Mod_Rewrite_Shop/...
http://testphp.vulnweb... GET /artists.php
http://testphp.vulnweb... GET /artists.php?artist=1
http://testphp.vulnweb... GET /artists.php?artist=2
Issues
Cleartext submission of password
Password field with autocomplete enabled
Unencrypted communications
Cookie without HttpOnly flag set
Email addresses disclosed [2]
Frameable response (potential Clickjacking)
Request Response
Raw Headers Hex
GET /login.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0)
Gecko/20100101 Firefox/92.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*
;q=0.8
Type a search term
0 matches
Advisory Request Response
Cleartext submission of password
Issue: Cleartext submission of password
Severity: High
Confidence: Certain

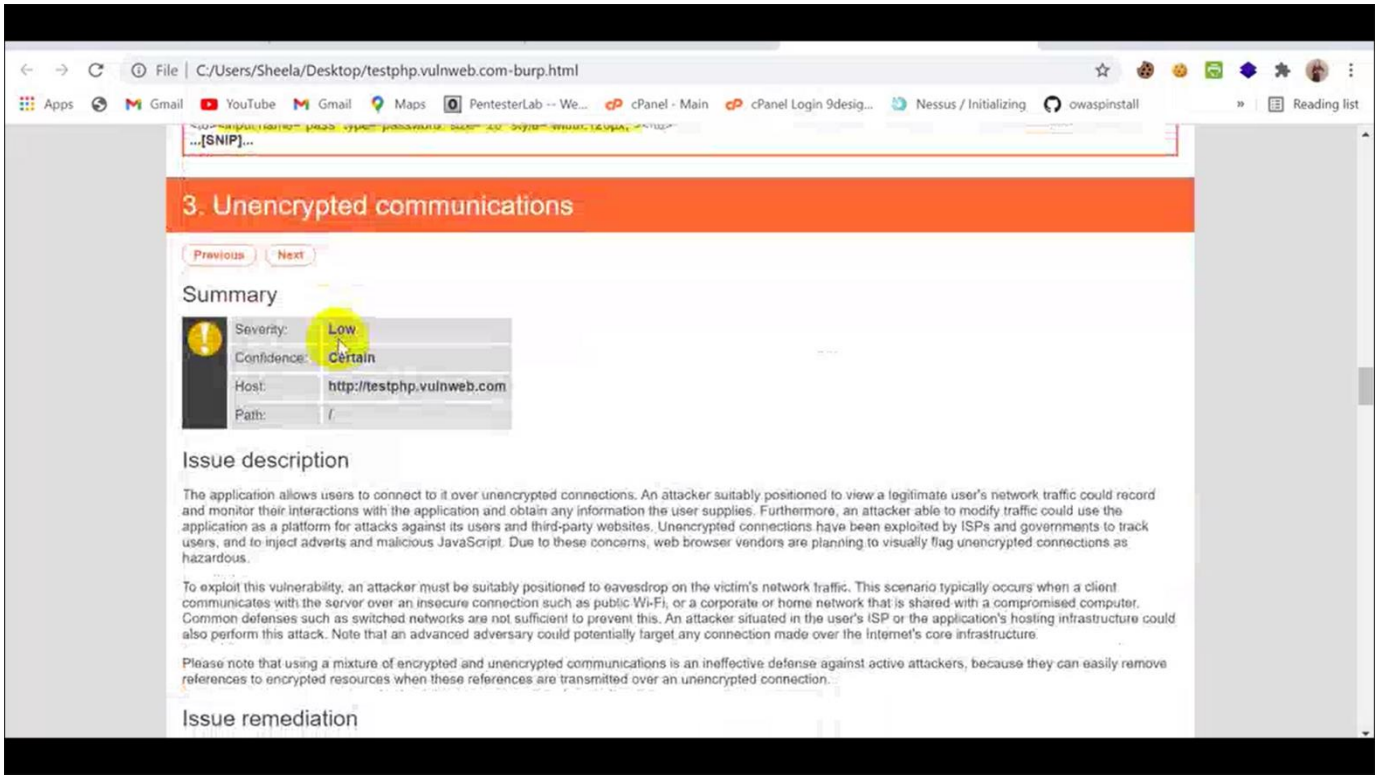

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Batch Scan Report Generator
Tasks
New scan
New live task
Filter Running Paused Finished
1. Live passive crawl from Proxy (all traffic)
Add links. Add item itself, same ...
16 items added to site map
Capturing:
2 responses processed
0 responses queued
2. Live audit from Proxy (all traffic)
Audit checks - passive
Issues:
Capturing:
0 requests (0 errors)
View details
3. Crawl and audit of testphp.vulnweb.com
Default configuration
Issues:
1070 requests (0 errors)
Unauthenticated crawl. 10m esti...
47 locations crawled
View details
Event log
Filter Critical Error Info
Search...
Time Type Source Message
Issue activity
Filter High Medium Low Info Certain Firm Tentative
Search...
Task Time Action Issue type
7 2 19:24:35 1 Oct 2021 Issue found Email addresses disclosed
6 2 19:24:35 1 Oct 2021 Issue found Cookie without HttpOnly flag set
5 2 19:24:15 1 Oct 2021 Issue found Email addresses disclosed
4 2 19:24:15 1 Oct 2021 Issue found Password field with autocomplete enabled
3 2 19:24:14 1 Oct 2021 Issue found Cleartext submission of password
2 2 19:24:14 1 Oct 2021 Issue found Frameable response (potential Clickjacking)
1 2 19:24:14 1 Oct 2021 Issue found Unencrypted communications
Advisory Request Response
Issue: Cleartext submission of password
Severity: High
Confidence: Certain
Host: http://testphp.vulnweb.com
Path: /login.php
Issue detail
The page contains a form with the following action URL, which is submitted over clear-text HTTP:
http://testphp.vulnweb.com/userinfo.php
Memory: 94.7MB
Disk: 59KB

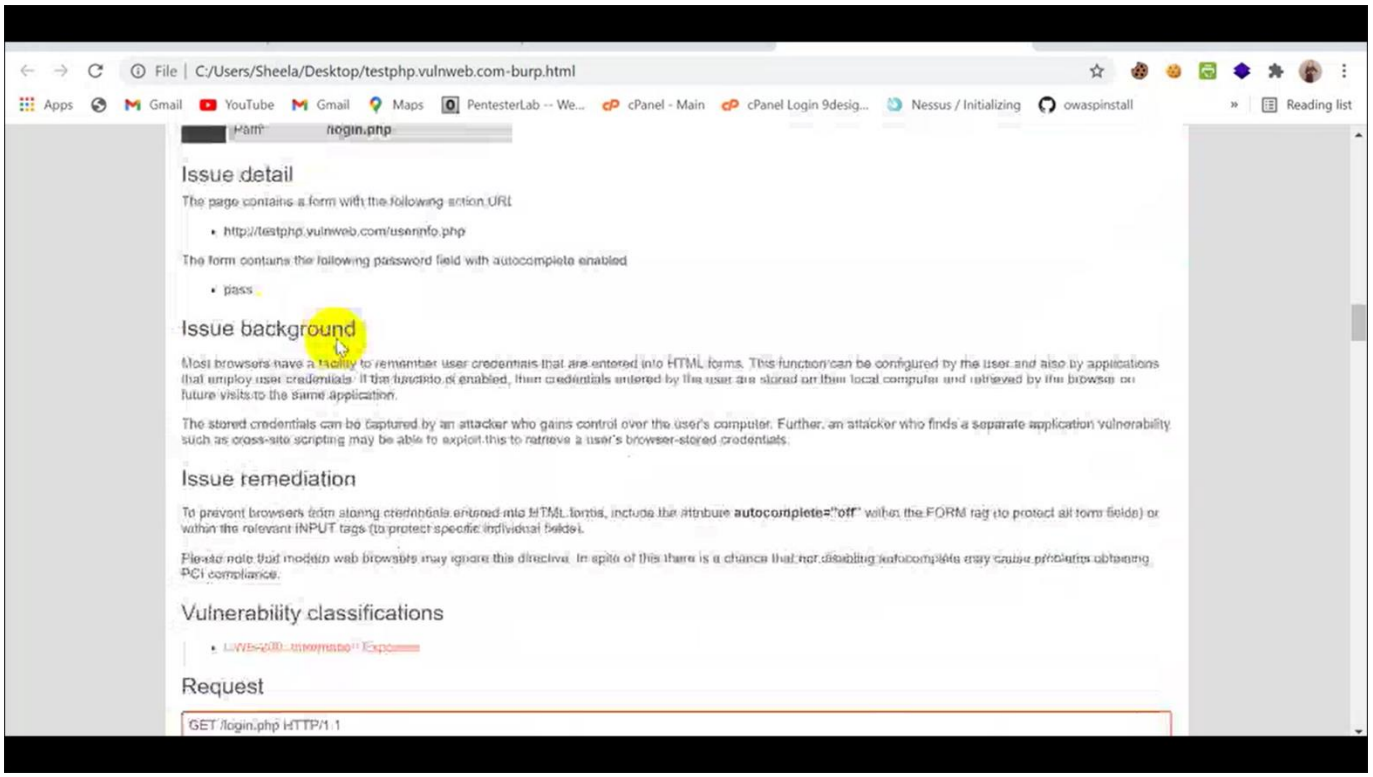

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Batch Scan Report Generator
Site map Scope Issue definitions
Issue Definitions
This listing contains the definitions of all issues that can be detected by Burp Scanner.
Name Typical severity Type index
Source code disclosure Low 0x00500000
Directory listing Information 0x00500100
Email addresses disclosed Information 0x00500200
Private IP addresses disclosed Information 0x00500300
Social security numbers disclosed Information 0x00500400
Credit card numbers disclosed Information 0x00500500
Private key disclosed Information 0x00500580
Robots.txt file Information 0x00500600
Cacheable HTTPS response Information 0x00700100
Base64-encoded data in parameter Information 0x00700200
Multiple content types specified Information 0x00800100
HTML does not specify charset Information 0x00800200
HTML uses unrecognized charset Information 0x00800300
Content type incorrectly stated Low 0x00800400
Content type is not specified Information 0x00800500
SSL certificate Medium 0x01000100
Unencrypted communications Low 0x01000200
Strict transport security not enforced Low 0x01000300
Mixed content Information 0x01000400
OS command injection
Description
Operating system command injection vulnerabilities arise when an application incorporates user-controllable data into a command that is processed by a shell command interpreter. If the user data is not strictly validated, an attacker can use shell metacharacters to modify the command that is executed, and inject arbitrary further commands that will be executed by the server.
OS command injection vulnerabilities are usually very serious and may lead to compromise of the server hosting the application, or of the application's own data and functionality. It may also be possible to use the server as a platform for attacks against other systems. The exact potential for exploitation depends upon the security context in which the command is executed, and the privileges that this context has regarding sensitive resources on the server.
Remediation
If possible, applications should avoid incorporating user-controllable data into operating system commands. In almost every situation, there are safer alternative methods of performing server-level tasks, which cannot be manipulated to perform additional commands than the one intended.

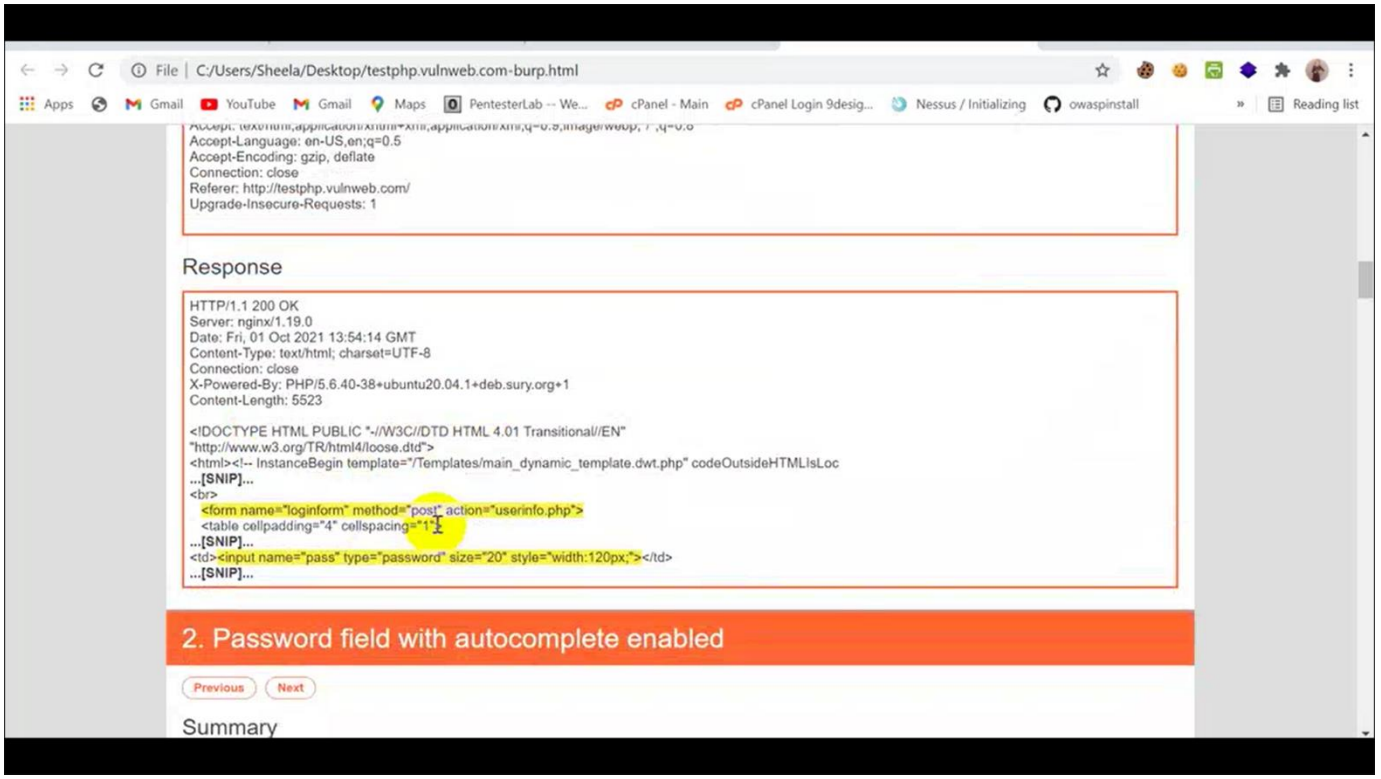

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Batch Scan Report Generator
Report Output Format: HTML XML
Report On In-Scope Sites Only:
Merge HTTP (port 80) and HTTPS (port 443) For Reports:
One Host Per Report (Combine All Protocols and Ports):
Report Output Root Directory: Select folder ... C:\Users\Sheela\Desktop
Append Date To Report Filenames: Date Format: MMDDYYYY
Generate Report(s)


File | C:/Users/Sheela/Desktop/testphp.vulnweb.com-burp.html
Apps Gmail YouTube Gmail Maps PentesterLab -- We... cPanel - Main cPanel Login 9desig... Nessus / Initializing owaspinstall Reading list
Burp Scanner Report
BURPSUITE PROFESSIONAL
Summary
The table below shows the numbers of issues identified in different categories. Issues are classified according to severity as High, Medium, Low or Information. This reflects the likely impact of each issue for a typical organization. Issues are also classified according to confidence as Certain, Firm or Tentative. This reflects the inherent reliability of the technique that was used to identify the issue.
Confidence
Certain Firm Tentative Total
Severity
High 1 0 0 1
Medium 0 0 0 0
Low 2 0 0 2
Information 3 1 0 4
The chart below shows the aggregated numbers of issues identified in each category. Solid colored bars represent issues with a confidence level of Certain, and the bars fade as the confidence level falls.
Number of issues
0 1 2 3 4
35:32 / 40:19

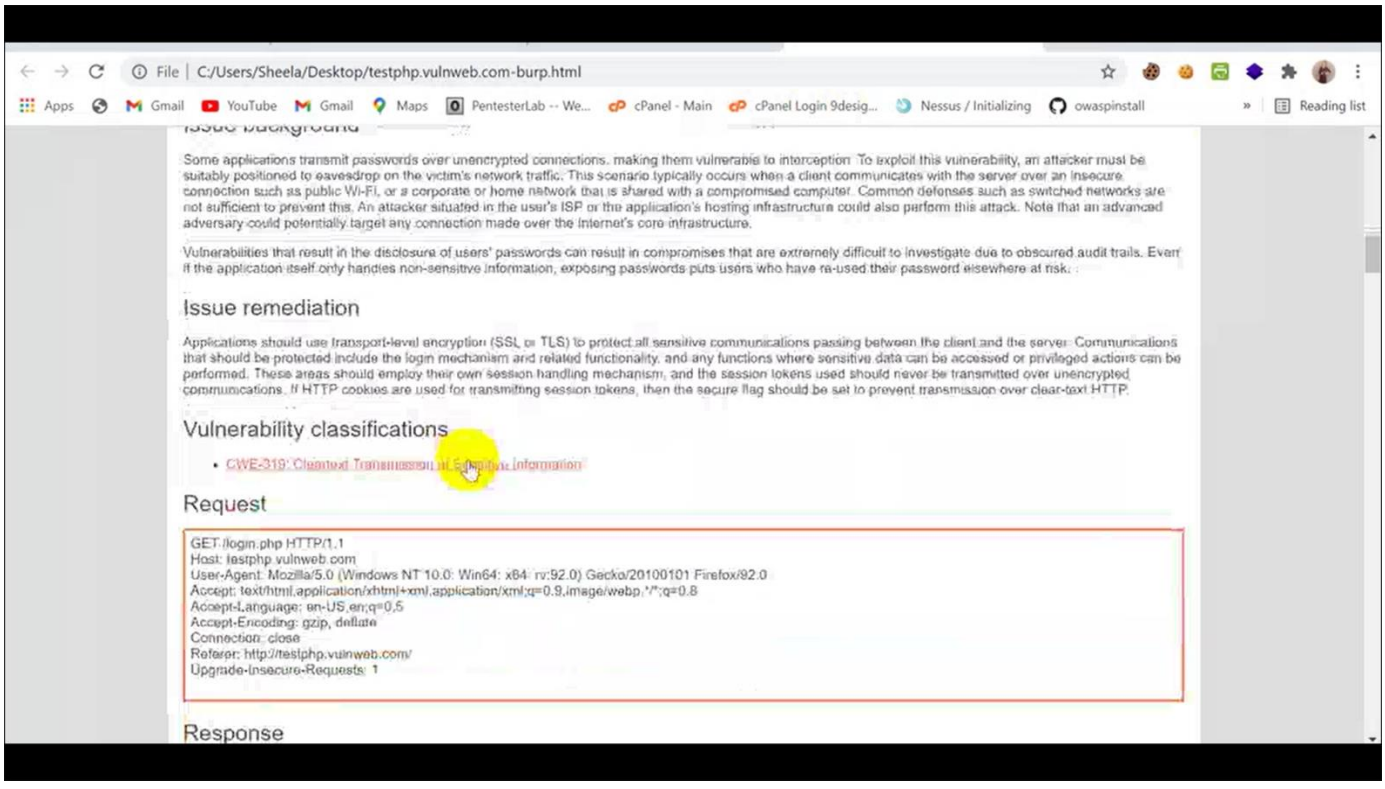

File | C:/Users/Sheela/Desktop/testphp.vulnweb.com-burp.html
Apps Gmail YouTube Gmail Maps PentesterLab -- We... cPanel - Main cPanel Login 9desig... Nessus / Initializing owaspinstall Reading list
The table below shows the numbers of issues identified in different categories. Issues are classified according to severity as High, Medium, Low or Information. This reflects the likely impact of each issue for a typical organization. Issues are also classified according to confidence as Certain, Firm or Tentative. This reflects the inherent reliability of the technique that was used to identify the issue.
Confidence
Certain Firm Tentative Total
Severity
High 1 0 0 1
Medium 0 0 0 0
Low 2 0 0 2
Information 3 1 0 4
The chart below shows the aggregated numbers of issues identified in each category. Solid colored bars represent issues with a confidence level of Certain, and the bars fade as the confidence level falls.
Number of issues
0 1 2 3 4
Severity
High
Medium
Low
Contents

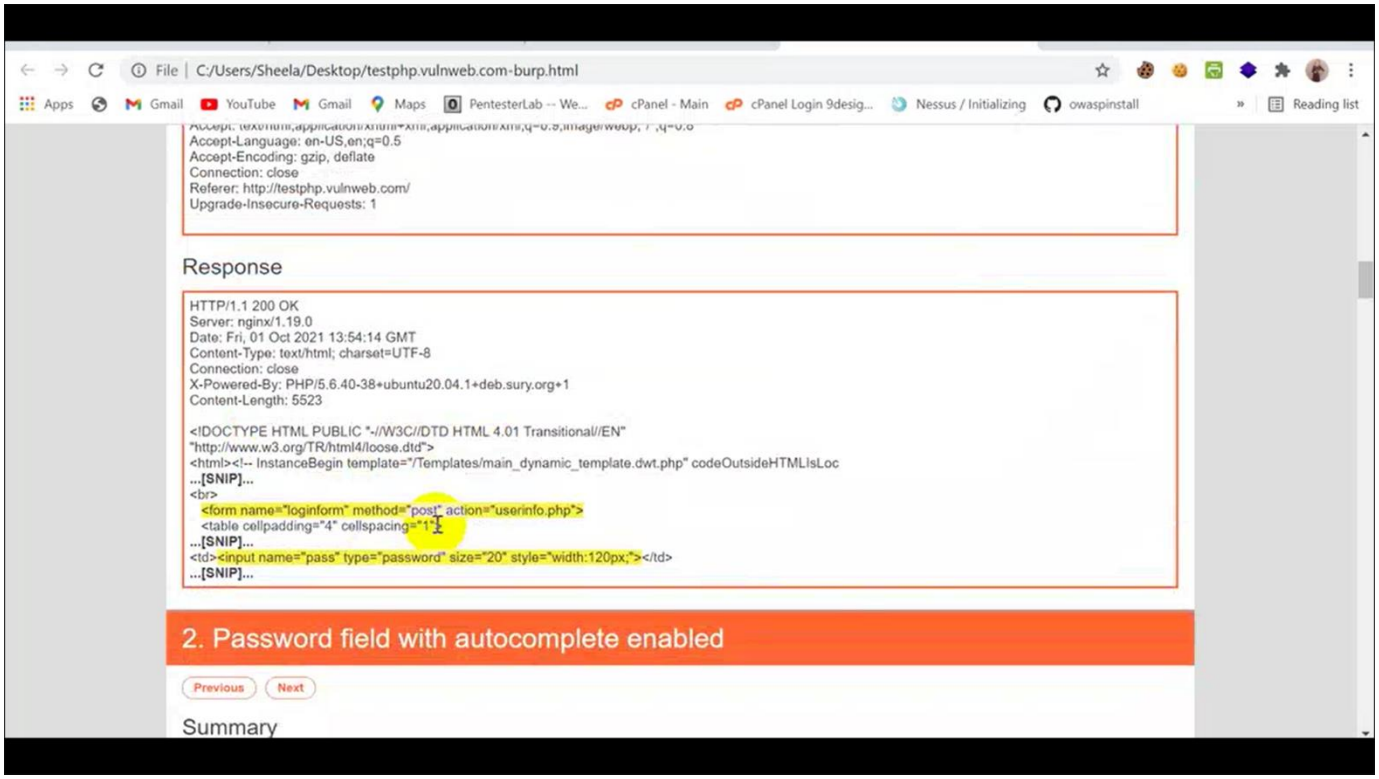

File | C:/Users/Sheela/Desktop/testphp.vulnweb.com-burp.html
Apps
Gmail
YouTube
Gmail
Maps
PentesterLab -- We...
cPanel - Main
cPanel Login 9desig...
Nessus / Initializing
owaspinstall
Reading list
...[SNIP]...
3. Unencrypted communications
Previous
Next
Summary
Severity: Low
Confidence: Certain
Host: http://testphp.vulnweb.com
Path: /
Issue description
The application allows users to connect to it over unencrypted connections. An attacker suitably positioned to view a legitimate user's network traffic could record and monitor their interactions with the application and obtain any information the user supplies. Furthermore, an attacker able to modify traffic could use the application as a platform for attacks against its users and third-party websites. Unencrypted connections have been exploited by ISPs and governments to track users, and to inject adverts and malicious JavaScript. Due to these concerns, web browser vendors are planning to visually flag unencrypted connections as hazardous.
To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.
Please note that using a mixture of encrypted and unencrypted communications is an ineffective defense against active attackers, because they can easily remove references to encrypted resources when these references are transmitted over an unencrypted connection.
Issue remediation


File | C:/Users/Sheela/Desktop/testphp.vulnweb.com-burp.html
Apps
Gmail
YouTube
Gmail
Maps
PentesterLab -- We...
cPanel - Main
cPanel Login 9desig...
Nessus / Initializing
owaspinstall
Reading list
Path: /login.php
Issue detail
The page contains a form with the following action URL
http://testphp.vulnweb.com/userinfo.php
The form contains the following password field with autocomplete enabled
pass
Issue background
Most browsers have a facility to remember user credentials that are entered into HTML forms. This function can be configured by the user and also by applications that employ user credentials. If the function is enabled, then credentials entered by the user are stored on their local computer and retrieved by the browser on future visits to the same application.
The stored credentials can be captured by an attacker who gains control over the user's computer. Further, an attacker who finds a separate application vulnerability such as cross-site scripting may be able to exploit this to retrieve a user's browser-stored credentials.
Issue remediation
To prevent browsers from storing credentials entered into HTML forms, include the attribute autocomplete="off" within the FORM tag (to protect all form fields) or within the relevant INPUT tags (to protect specific individual fields).
Please note that modern web browsers may ignore this directive. In spite of this there is a chance that not disabling autocomplete may cause problems obtaining PCI compliance.
Vulnerability classifications
CWE-200: Information Exposure
Request
GET /login.php HTTP/1.1


File | C:/Users/Sheela/Desktop/testphp.vulnweb.com-burp.html
Apps Gmail YouTube Gmail Maps PentesterLab -- We... cPanel - Main cPanel Login 9desig... Nessus / Initializing owaspinstall Reading list
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://testphp.vulnweb.com/
Upgrade-Insecure-Requests: 1
Response
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Fri, 01 Oct 2021 13:54:14 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Content-Length: 5523
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" codeOutsideHTMLIsLoc
...[SNIP]...

<form name="loginform" method="post" action="userinfo.php">
<table cellpadding="4" cellspacing="1">
...[SNIP]...
<td><input name="pass" type="password" size="20" style="width:120px;"></td>
...[SNIP]...
2. Password field with autocomplete enabled
Previous
Next
Summary


File | C:/Users/Sheela/Desktop/testphp.vulnweb.com-burp.html
Apps Gmail YouTube Gmail Maps PentesterLab -- We... cPanel - Main cPanel Login 9desig... Nessus / Initializing owaspinstall Reading list
Some applications transmit passwords over unencrypted connections, making them vulnerable to interception. To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.
Vulnerabilities that result in the disclosure of users' passwords can result in compromises that are extremely difficult to investigate due to obscured audit trails. Even if the application itself only handles non-sensitive information, exposing passwords puts users who have re-used their password elsewhere at risk.
Issue remediation
Applications should use transport-level encryption (SSL or TLS) to protect all sensitive communications passing between the client and the server. Communications that should be protected include the login mechanism and related functionality, and any functions where sensitive data can be accessed or privileged actions can be performed. These areas should employ their own session handling mechanism, and the session tokens used should never be transmitted over unencrypted communications. If HTTP cookies are used for transmitting session tokens, then the secure flag should be set to prevent transmission over clear-text HTTP.
Vulnerability classifications
CWE-319: Cleartext Transmission of Sensitive Information
Request
GET /login.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://testphp.vulnweb.com/
Upgrade-Insecure-Requests: 1
Response



Conclusion:

I learned about OWASP ZAP scanning app.

I also learned about various injections and the vulnerabilities and how to work with the OWASP ZAP.

I learned to work with the burpsite.

I learned to read the vulnerability report generated by the burpsite and OWASP ZAP.